

Linköping studies in science and technology. Dissertations.
No. 1476

Extended target tracking using PHD filters

Karl Granström



Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden

Linköping 2012

Cover illustration: Laser range data used in extended target tracking experiments. Two persons were measured at waist height as they walked around in front of the laser range sensor. The data is plotted in three dimensions, with time along the vertical axis. Tracking results for this data set are presented in Figure 11 in Paper B, and in Figure 9 in Paper D.

Nota bene: The illustration features no less than 400 different colors. No colors were harmed in the making of this cover illustration.

Linköping studies in science and technology. Dissertations.
No. 1476

Extended target tracking using PHD filters

Karl Granström

karl@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-7519-796-8

ISSN 0345-7524

Copyright © 2012 Karl Granström

Printed by LiU-Tryck, Linköping, Sweden 2012

Till mina föräldrar

Abstract

The world in which we live is becoming more and more automated, exemplified by the numerous robots, or autonomous vehicles, that operate in air, on land, or in water. These robots perform a wide array of different tasks, ranging from the dangerous, such as underground mining, to the boring, such as vacuum cleaning. In common for all different robots is that they must possess a certain degree of awareness, both of themselves and of the world in which they operate. This thesis considers aspects of two research problems associated with this, more specifically the Simultaneous Localization and Mapping (SLAM) problem and the Multiple Target Tracking (MTT) problem.

The SLAM problem consists of having the robot create a map of an environment and simultaneously localize itself in the same map. One way to reduce the effect of small errors that inevitably accumulate over time, and could significantly distort the SLAM result, is to detect loop closure. In this thesis loop closure detection is considered for robots equipped with laser range sensors. Machine learning is used to construct a loop closure detection classifier, and experiments show that the classifier compares well to related work.

The resulting SLAM map should only contain stationary objects, however the world also contains moving objects, and to function well a robot should be able to handle both types of objects. The MTT problem consists of having the robot keep track of where the moving objects, called targets, are located, and how these targets are moving. This function has a wide range of applications, including tracking of pedestrians, bicycles and cars in urban environments. Solving the MTT problem can be decomposed into two parts: one part is finding out the number of targets, the other part is finding out what the states of the individual targets are.

In this thesis the emphasis is on tracking of so called extended targets. An extended target is a target that can generate any number of measurements, as opposed to a point target that generates at most one measurement. More than one measurement per target raise interesting possibilities to estimate the size and the shape of the target. One way to model the number of targets and the target states is to use random finite sets, which leads to the Probability Hypothesis Density (PHD) filters. Two implementations of an extended target PHD filter are given, one using Gaussian mixtures and one using Gaussian inverse Wishart (GIW) mixtures. Two models for the size and shape of an extended target measured with laser range sensors are suggested. A framework for estimation of the number of measurements generated by the targets is presented, and reduction of GIW mixtures is addressed. Prediction, spawning and combination of extended targets modeled using GIW distributions is also presented. The extended target tracking functions are evaluated in simulations and in experiments with laser range data.

Populärvetenskaplig sammanfattning

Den värld i vilken vi lever har med tiden blivit allt mer automatiserad. Ett av många tecken på detta är det stora antal robotar, eller autonoma farkoster, som verkar bland annat i luften, på land, eller i vatten. De här robotarna kan utföra ett brett spektrum av olika uppgifter, allt ifrån direkt farliga, som underjordisk gruvdrift och sanering av havererade kärnreaktorer, till alldagliga och tråkiga, som dammsugning och gräsklippning. På samma sätt som en människa behöver använda sina sinnen och sitt medvetande för att hantera vardagen, måste alla typer av robotar ha en viss medvetenhet för att kunna utföra sina uppgifter. Det krävs bland annat att robotarna kan uppfatta och förstå sin arbetsmiljö.

I den här avhandlingen behandlas ett antal delar av två stycken övergripande forskningsproblem som är relaterade till detta. Det första forskningsproblemet kallas för samtidig positionering och kartering, vilket på engelska heter Simultaneous Localization and Mapping och förkortas SLAM. Det andra forskningsproblemet kallas för målföljning.

SLAM-problemet går ut på att låta roboten skapa en karta av ett område, och samtidigt som kartan skapas positionera sig i den. Exakt vad som menas med karta i det här sammanhanget varierar beroende på robotens specifika arbetsuppgift. Exempelvis kan det, för en inomhusrobot, röra sig om en virtuell modell av var golv, väggar och möbler finns i ett hus. En oundviklig del av SLAM-problemet är att roboten hela tiden gör små fel, vilket påverkar kartan som skapas, samt hur väl roboten kan positionera sig. Enskilda fel har inte särskilt stor inverkan, men om felen ackumuleras under en längre tid kan det leda till att kartan förvrängs, eller att roboten helt enkelt inte kan finna sin position i kartan.

Ett sätt att undvika att så sker är att utrusta roboten med en funktion vilken gör det möjligt för roboten att känna igen platser som den har besökt tidigare, vilket kallas platsigenkänning. När roboten känner igen en plats kan den jämföra med vad kartan och positionen säger. Om kartan och positionen inte säger att roboten är tillbaka på en plats som tidigare besökts kan denna diskrepans korrigeras. Resultatet är en karta och en position som bättre representerar verkligheten. I den här avhandlingen har platsigenkänning studerats för robotar som är utrustade med laserscannern, och en funktion för platsigenkänning har skapats. I en serie experiment har det visats att funktionen kan känna igen platser såväl inomhus i kontorsmiljö, som utomhus i stadsmiljö. Det har även visats att funktionens egenskaper jämför sig väl med tidigare arbete på området.

Den resulterande SLAM-kartan bör av naturliga skäl endast innehålla stationära föremål. Vår värld innehåller dock även rörliga föremål, och för att en robot ska kunna arbeta på ett säkert sätt måste den även hålla reda på alla rörliga föremål som finns i dess närhet. Det andra forskningsproblemet som behandlats i avhandlingen, målföljning, går ut på att utrusta roboten med funktioner som gör det möjligt för den att hålla reda på var de rörliga målen är, samt vart de är på väg att röra sig. Exempelvis kan den här typen av funktioner användas till att hålla reda på fotgängare, cyklister och bilar i en stadsmiljö.

Tidigare har forskningen inom målföljning varit fokuserad på så kallade punktmål. Vid följdning av punktmål kan följningsproblemet sägas ha två delar: den ena är att räkna ut hur många rörliga mål det finns, den andra är att räkna ut var varje enskilt mål befinner sig, samt vart det är på väg.

Här har fokus istället legat på följdning av vad som kallas för utsträckta mål, en typ av mål som rönt ökande uppmärksamhet i forskningsvärlden de senaste fem till tio åren. Med utsträckta mål får följningsproblemet en tredje del: att för varje enskilt mål räkna ut storleken och formen på målet, det vill säga den spatiala utsträckningen. Att känna till utsträckningen på de rörliga målen är viktigt exempelvis för en robot som ska ta sig genom ett rum där många personer befinner sig. För att göra det krävs att roboten rör sig nära personerna, utan att för den skull krocka med någon. Att lösa detta på ett bra sätt kräver att roboten har kunskap inte bara om var personerna befinner sig, utan även hur mycket plats de tar upp.

I avhandlingen har ett antal aspekter av följdning av utsträckta mål studerats. En viktig och komplicerande aspekt av följdning av såväl punktmål, som utsträckta mål, är att roboten på förhand inte vet hur många mål som finns i dess närhet. En funktion för att hantera osäkerheterna kring antalet mål som finns, samt osäkerheterna kring var varje mål befinner sig, har implementerats.

I många situationer är det nödvändigt att kunna prediktera, eller förutsäga, var de olika målen kommer att befinna sig i den närmaste framtiden. Det kan exempelvis röra sig om en robot som ska köra genom en vägkorsning, och då måste undvika att krocka med övrig trafik. För detta ändamål har en prediktionsfunktion tagits fram.

När ett större antal mål rör sig i robotens närhet kan det bli svårt att följa varje enskilt mål. Istället kan roboten följa grupper av mål. Det blir då nödvändigt att hålla reda på vad som sker när mål lämnar gruppen, eller nya mål ansluter till gruppen. Fritt översatt från engelska till svenska kan dessa två händelser kallas för målproduktion och målkombination. Funktioner för att hantera produktion och kombination av utsträckta mål har tagits fram.

För att roboten ska kunna beräkna ett måls spatiala utsträckning krävs modeller för formen på målen. När laserscanners används kan formen på en bil sägas vara approximativt rektangulär, och formen på en person kan sägas vara approximativt elliptisk. Beräkning av storleken på rektangulära och elliptiska mål har studerats för robotar utrustade med laserscanners.

Målföljningsfunktionerna som nämnts ovan har utvärderats med hjälp av såväl simulerade data, som experimentella data insamlade med laserscanners. Resultaten visar att det arbete som har utförts jämför sig väl med tidigare arbete på området.

Acknowledgments

Whenever I get my hands on a thesis, there are two things that I always read: the dedication and the acknowledgments. There, in the middle of the objective, and often dry, academic writing lies the only chance to get a glimpse of the authors personality, a glimpse of the living and breathing human being behind the theories and the hypotheses, behind the theorems and the proofs. I shall therefore try to seize this opportunity, and make an attempt to show you my personality – especially my sense of humor – as I give my thanks to the persons to whom my thanks are due. However, please do not make yourself the illusion that I do not take this seriously. What is written below, is written with the utmost respect for all those that are mentioned.

With that said, in the interest of letting first things be first, I will start at the beginning. This thesis, and the research that it contains, is a product of my time at the Automatic Control group at Linköping University. If you should ever find yourself in Linköping, you may wish to visit the central library. Built in 2000, after a fire ravaged the old library in 1996, it features large glass windows that offer great views of Linköping Cathedral, and it boast in excess of 359604 book-titles. On one of the many shelves you can find Jack Kerouac's *On the road*, a story in which the main character claims to have

nothing to offer anybody except my own confusion.

While I would like to hope that I had something more than just my confusion to offer during those first months as a member of Automatic Control, I will gladly admit that starting a voyage towards a PhD-thesis indeed can be confusing.

In ironing out the worst kinks in my sheet of confusion, I received tremendous help from my supervisor Dr Thomas Schön. Thomas made sure that I understood the importance of structure, be it for a paper, a presentation, undergraduate teaching, or just for planning which graduate courses to take, and when to take them. He always served as a great source of enthusiasm and encouragement, and during the last parts of my PhD studies he was gracious enough to let me venture off in my own research direction.

Fortunately, on my continued path I found myself not alone, but under the guidance and patient support of Dr Umut Orguner. Had he not been there to straighten my steps, I probably would have

looked like a crab scurrying across the sand looking for bacon.

Our collaboration has been most fruitful, and it has been very inspiring for me to be part of it. I am forever indebted to Umut for all the knowledge and wisdom he has bestowed upon me. I have enjoyed all our discussions, and I truly hope that we will continue to write papers together.

If Thomas and Umut are to thank for helping me survive these four years, Dr Fabio Ramos and Dr Juan Nieto are to thank for making me believe that I could pull it all together. I may have left Sweden thinking that I was just going to write a

Master's thesis, but thanks to them I came back from Australia with my mind set on writing a PhD thesis. It has been a while since we last saw each other, but I hope our paths will cross again soon, so that we once again can enjoy the finer pleasures of academic life: the exquisite wines, the delicious barbeques, and the late night karaoke.

My days with Automatic Control would not have been, had it not been for the job offer I received from Prof Lennart Ljung and Prof Fredrik Gustafsson. I may have hesitated for a moment before accepting it, but once I joined the group I never looked back. Fredrik is the epicenter of the vibrant and bustling Sensor Fusion group, which I am proud to be part of. Lennart stood at the helm of Automatic Control for some thirty years, and managed to create a very impressive research environment before handing over the ship to Prof Svante Gunnarsson. Svante may have had big shoes to fill, but he filled them with an ease and elegance that is second to none. The only thing I could never quite understand is why he, towards the end of Mjärdevistafetten 2011, took that

fateful, pear shaped, left turn.

But then again, I am hardly in the position to question the judgment of a winner of the teaching award *Gyllene Moroten*. However, I am in the position to extend my sincere gratitude to Automatic Control's secretaries Ulla Salaneck, Åsa Karmelind and Ninna Stensgård, who have made sure that the administrative machinery has run without interruption.

This thesis was written using the thesis template constructed by T_EX-gurus-at-large, Dr Gustaf Hendeby and Dr Henrik Tidefelt. Thanks to their meticulous attention to detail, the process of writing became smooth as silk. The thesis was proofread by Thomas, Umut, Fredrik, Dr Christian Lundquist and Lic Jonas Callmer, who all contributed with invaluable comments. Any and all remaining errors are – naturally – mine.

Jeffrey Bernard, a British *causeur* famous for the exhortation

aim low – and miss,

ended his days on earth by ingesting a toxic amount of bananas, an event he referred to as a *banana split*. In his obituary – written by none other than himself – he professed to having developed a fantasy that, starting tomorrow, things would finally take a turn for the better. Thinking that a geographical relocation would solve his problems, he longed for various dream cottages on the countryside. The experience was always ruined when he found himself living at the same location.

Personally, I have never longed for dream cottages – my grandmother already owns one on the Swedish west coast – but I have longed for dream colleagues. At Automatic Control in Linköping I found them, and I have enjoyed every day I have spent with them. I can only hope that their experience was not ruined when they found myself working at the same location.

Dr Christian Lundquist was kind enough to let me tag along as his PHD-train left

the station, a journey I have never regretted being part of. Our collaboration has always be inspiring to me, and I have always appreciated his constructive and invigorating critique of my work. It seems that lately the writing of two certain PhD theses has halted our forward motion. However, I am confident that we will pick up speed soon enough, provided that I do not stop completely to barrage him with another assortment of *sill*-related jokes.

One of my closest friends, Lic Jonas Callmer, has been along for the ride ever since we, in between spending time on Sydney's beaches and drinking free beer at The Gaff, managed to write a Master's thesis. I always enjoy our discussions, regardless of whether they are about serious politics, or about office related gossip. Alas, I never did understand his interest in getting fit by kicking other men in the balls, but then again, a great friendship is only strengthened by differences.

Lic Morgan Skoglund, *First Lord Protector of the Order of the Räv hjälm*, has always been there to remind me that it is time for coffee, or time for cross-country skiing, or time for our lunchtime 5K run, or simply time to head to the lab so that all the microphones can be hung – not like horses – but from the ceiling. No matter what we do, with Martin it always is, and always has been, a good time.

When I first started working at Automatic Control, there was this tall guy talking loudly in the *fika*-room. Often, as I walked back to my office after finishing my coffee, I could not help but wonder,

Wha' happen'!?!

To tell you the truth, I still do not know, but I do know this: Lic Zoran Sjanic is not only tall and loud, he is also super-hilarious and fun, and above all he is a great friend. I can always trust Zoran when I want to have a bit of fun, regardless of whether we are in Linköping, on the Balkans, or in some godforsaken corner of the globe.

The *travelling-banana-salesman-and-monkey-dompteur*, Lic André Carvalho Bittencourt, was an awesome travel companion for two weeks in Taiwan. I feel very fortunate that I had the chance to get to know him, and I hope he is not too sad about the fact that I turned out to be the most handsome one of us.

If there is anyone you can trust when it is time to round up the troops for a pub crawl, it is Automatic Control's resident *BBQ-connoisseur*, Lic Sina Khoshfetrat Pakazad. He has a most infectious laughter, he always serves copious amounts of delicious meat, and he has absolutely despicable taste in movies.

As one goes through life, one meets an innumerable number of persons. Sometimes these encounters are like two ships, steaming past each other on a stormy ocean, to the sound of thunder and lightning. Other times the encounters are more like

*två gistna ekor, guppandes förbi varandra på en försurad
sjö i Småland, till ljudet av storlommens lockrop.*

The siren call of a black-throated loon?! What is it? Well, it's a chirp signal with logarithmically increasing frequency, but that's not important right now.

What is important, are all the friends that I have met on my path through life. This path has stretched over my childhood in Enebyberg, via a year as a foreign exchange student in Cottage Grove, to what has become a decade in Linköping, with a brief *caesura* for a years worth of surfing in New Zealand. You are far too many to mention, and far too important to forget. I hold the memories of the fun times we have had very dear, and I smile at the thought of all the fun that is yet to come.

Of all the gifts I received as a child, the finest one must be to have the best possible siblings I could have ever wanted. Together, Emma, Erik and I have made *kajsor av godisnappar*, fished for crab at the *campingbryggan*, *snorklat* in the *poolen*, and helped each other avoid temptation to

the greatest of the cardinal sins: landkrabberi.

By all means, measures, metrics and standards, it has been very fun, and I'm certain it will continue to be so.

In a letter to Robert Hooke, Isaac Newton expressed that "If I have seen further it is by standing on ye sholders of Giants." Blessed with my grandfathers height, one might think that I do not need shoulders, or anything at all for that matter, to stand on. Nevertheless, my parents have provided me with all the shoulders I have ever needed to see far. With love and care they have instilled in me a feeling of being capable of anything that I set my mind to. They have supported me in wet and dry, they have celebrated my accomplishments, and they have accepted my mistakes. *För allt ni har gjort, och allt ni har betytt – tack!*

Last but not least, I wish to acknowledge the financial support that I have received from the Linnaeus research environment CADICS and the frame project grant Extended Target Tracking (621-2010-4301), both funded by the Swedish Research Council, and from the project Collaborative Unmanned Aircraft Systems (CUAS), funded by the Swedish Foundation for Strategic Research (SSF).

Linköping, October 2012
Karl Granström

Ladies and gentlemen!
The tall man with the glasses,
has now left the patio!

Contents

Notation	xxi
I Background	
1 Introduction	3
1.1 Motivation	3
1.2 Loop closure detection	4
1.3 Multiple target tracking	5
1.4 Publications	6
1.5 Main contributions	8
1.5.1 Loop closure detection	8
1.5.2 Target tracking	9
1.6 Thesis outline	11
2 The laser range sensor	13
2.1 Introduction	13
2.2 Laser range data in 2D	15
2.3 Laser range data in 3D	17
2.4 Occlusion	18
2.5 Registration	19
3 Classification	21
3.1 The classification problem	21
3.2 Boosting	22
3.2.1 Adaptive boosting	22
3.2.2 Examples	25
3.2.3 Properties	28
3.2.4 S-fold cross validation	30
3.3 Performance evaluation	31
3.3.1 Basic concepts	31
3.3.2 Detection and false alarm	32
3.3.3 Receiver operating characteristic	33

4	Estimation	37
4.1	The estimation problem	37
4.2	Dynamic models and measurement models	38
4.2.1	Example in continuous time	39
4.2.2	Examples in discrete time	39
4.3	Recursive single state Bayes filter	41
4.4	Some solutions to the estimation problem	43
4.4.1	Linear estimation with the Kalman filter	43
4.4.2	Non-linear estimation	45
4.5	Performance evaluation	45
4.5.1	The root mean square error	45
4.5.2	The normalized estimation error square	47
4.6	Simultaneous localization and mapping	48
5	Target tracking	53
5.1	The target tracking problem	53
5.1.1	Single target tracking	53
5.1.2	Multiple target tracking	54
5.2	Data association methods	55
5.2.1	Single target tracking	55
5.2.2	Multiple target tracking	56
5.3	Performance evaluation	57
6	Random finite sets and the probability hypothesis density	61
6.1	Introduction	61
6.1.1	Random finite sets	61
6.1.2	A brief overview of multi-target calculus	62
6.2	Recursive multi-state Bayes filter	64
6.2.1	The probability hypothesis density filter	65
6.2.2	The cardinalized probability hypothesis density filter	68
6.3	A brief revisit to the SLAM problem	69
7	Extended target tracking	71
7.1	Introduction	71
7.2	Extended target modeling	72
7.2.1	Extended target measurements	72
7.2.2	Extended target state	73
7.2.3	Extension shape models	78
7.3	Measurement set partitioning	79
7.4	Performance evaluation	81
8	Concluding remarks	85
8.1	Conclusions	85
8.2	Future work	86
	Bibliography	89

II Publications

A	Learning to close loops from range data	99
1	Introduction	101
2	Related work	103
3	Loop closure detection	106
	3.1 Algorithm overview	107
	3.2 Features	107
	3.3 Classification using AdaBoost	109
4	Simultaneous localization and mapping	110
	4.1 Exactly Sparse Delayed-state Filters	111
	4.2 Robot pose, process and measurement models	112
	4.3 Point cloud registration	113
5	Experimental results	114
	5.1 Data	115
	5.2 Classifier evaluation	118
	5.3 SLAM experiments	132
	5.4 Summary and comparison	134
6	Conclusions and future work	138
A	Appendix	139
	A.1 Feature definitions	139
	A.2 Compounding operations	142
	Bibliography	146
B	Extended target tracking using a Gaussian mixture PHD filter	151
1	Introduction	153
2	Target Tracking Problem Formulation	155
3	Gaussian-Mixture Implementation	157
4	Partitioning the Measurement Set	161
	4.1 Distance Partitioning	162
	4.2 Alternative Partitioning Methods	163
	4.3 Sub-Partitioning	165
5	Target Tracking Setup	168
6	Simulation Results	169
	6.1 True target tracks	169
	6.2 Comparison of Distance Partitioning and K -means++	170
	6.3 Benefits of Sub-Partition	171
	6.4 Comparison with GM-PHD	171
	6.5 Standard single measurement targets	173
	6.6 Unknown expected number of measurements γ	174
7	Experiment Results	179
	7.1 Experiment with close targets	179
	7.2 Experiment with occlusion	181
8	Conclusions	182
9	Future Work	183
A	Appendix	183

A.1	Proof of Theorem 1	183
A.2	Variable Probability of Detection for the Laser Sensor	184
	Bibliography	186
C	Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements	189
1	Introduction	191
2	State representation	194
3	Extended target tracking	195
3.1	GM-PHD target tracking	195
3.2	Problem definition	197
3.3	Multiple shapes and multiple targets	198
4	Computing predicted measurements and innovation covariances	199
4.1	Predicted measurements for rectangular targets	199
4.2	Predicted measurements for elliptical targets	200
4.3	Innovation covariances	202
5	Extended target tracking performance evaluation	203
6	Results	204
6.1	Simulations	204
6.2	Experiment	205
7	Conclusions and future work	207
	Bibliography	209
D	A PHD filter for tracking multiple extended targets using random matrices	211
1	Introduction	213
2	Modeling the target extension	216
3	Target Tracking Problem Formulation	217
4	The Gaussian inverse Wishart PHD filter	218
4.1	Assumptions	220
4.2	Prediction	222
4.3	Correction	223
4.4	Pruning and merging	225
4.5	Implementation of the GIWPHD filter	226
5	Partitioning the measurement set	226
5.1	Prediction Partition	228
5.2	EM Partition	228
5.3	Discussion	229
6	Simulation Results	230
6.1	Target tracking setup	230
6.2	Crossing tracks	231
6.3	Parallel tracks	231
6.4	Separating tracks	233
6.5	Closely spaced targets	234
7	Experiment results	237

7.1	Target tracking setup	237
7.2	Experiment with close targets	237
7.3	Experiment with occlusion	238
7.4	Discussion	240
8	Conclusions and future work	241
A	Appendix	242
A.1	Derivation of the correction	242
A.2	Variable probability of detection for the laser range sensor	244
A.3	Pseudo-code	245
A.4	Implementation issues	245
A.5	Computational complexity analysis	246
	Bibliography	253
E	Estimation and Maintenance of Measurement Rates for Multiple Ex-	
	tended Target Tracking	257
1	Introduction	259
2	Bayesian recursion for γ_k	261
2.1	Measurement update and prediction	261
2.2	Extended target predicted likelihood	263
3	Multi-target mixture reduction	264
3.1	Merging N Gamma components	265
3.2	Merging criterion for Gamma components	266
3.3	Merging of extended target components	267
4	Results	267
4.1	Merging criterion	268
4.2	Comparison of merging algorithms	268
4.3	Single target results	269
4.4	Multiple target results	269
5	Concluding Remarks	270
A	Appendix	271
A.1	Expected value of logarithm	272
A.2	Proof of Theorem 1	272
	Bibliography	274
F	On the Reduction of Gaussian inverse Wishart Mixtures	277
1	Introduction	279
2	Problem formulation	281
3	Approximating a weighted sum of GIW-components with one GIW- component	282
4	Merging criterion	284
4.1	Distance measure	284
4.2	A closer look at the Gaussian KL-DIFF	286
4.3	A closer look at the inverse Wishart KL-DIFF	286
4.4	Discussion	287
5	Merging algorithm	287
6	Simulation results	287

6.1	Merging criterion	288
6.2	Merging of one dimensional components	290
6.3	Merging of two dimensional components	290
6.4	Comparison of merging algorithms	290
7	Concluding remarks	291
A	Appendix	292
A.1	Expected value of inverse extension	292
A.2	Expected value of log determinant of extension	292
A.3	Proof of Theorem 1	294
	Bibliography	296
G	A New Prediction for Extended Targets with Random Matrices	299
1	Introduction	301
2	The random matrix framework	302
3	Problem formulation	304
4	Preliminaries	307
4.1	Known results	307
4.2	Approximating a \mathcal{GB}_d^{II} with an \mathcal{IW}_d	307
4.3	Approximating the density of \mathbb{V}_x with a \mathcal{W}_d	309
4.4	Marginalizing $\mathcal{IW}_d(X V)\mathcal{W}(V)$ over V	310
5	A new prediction update for the extension	310
5.1	Predicting the kinematical state	310
5.2	Predicting the extension state	311
5.3	Another look at the parameter n_{k+1}	313
6	Simulations	314
6.1	Determining τ	314
6.2	Difference measure for probability density functions	314
6.3	Results in one dimension	315
6.4	Results in two dimensions	317
7	Concluding remarks	324
A	Appendix	325
A.1	Preliminaries	325
A.2	Expected values of the \mathcal{GB}_d^{II} -distribution	326
A.3	Proof of Theorem 1	328
A.4	Proof of Theorem 2	330
A.5	Proof of Theorem 3	331
	Bibliography	333
H	On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices	337
1	Introduction	339
2	Extended target framework and problem formulation	341
3	Preliminary results on probability density approximations	343
3.1	Approximating the distribution of functions of gamma distributed random variables	344

3.2	Approximating matrix variate densities	345
3.3	Numerical root-finding	346
4	Target combination	346
4.1	Combination model	347
4.2	Combined distribution for two extended targets	348
4.3	Target combination criterion	350
5	Target spawning	351
5.1	Spawning model	351
5.2	Spawning hypotheses	352
6	On the use of other spatial distributions	354
7	Multiple target tracking framework	355
7.1	The GGIW-PHD filter	355
7.2	Combination in the GGIW-PHD filter	356
7.3	Spawning in the GGIW-PHD filter	357
7.4	Performance Evaluation	357
8	Simulation study	359
8.1	Multiple target tracking setup	359
8.2	True target tracks	359
8.3	Combination results	361
8.4	Spawning results	361
8.5	Cycle times	366
9	Concluding remarks	367
A	Appendix	368
A.1	Properties of Gamma distributed random variables	368
A.2	Matrix product of sum of Gaussians	368
A.3	Marginal distribution of kinematical state	369
A.4	Expected values	369
A.5	Proof of Theorem 1	372
A.6	Proof of Theorem 2	373
A.7	Proof of Theorem 3	373
A.8	Proof of Theorem 4	374
	Bibliography	375

Notation

ABBREVIATIONS

Abbreviation	Meaning
CPHD	Cardinalized Probability Hypothesis Density
CRF	Conditional Random Field
D	Detection
EKF	Extended Kalman Filter
ESDF	Exactly Sparse Delayed-state Filter
FA	False Alarm
FISST	Finite Set Statistics
GG	Gamma Gaussian
GIW	Gaussian inverse Wishart
GGIW	Gamma Gaussian inverse Wishart
GM-PHD	Gaussian Mixture Probability Hypothesis Density
GNN	Global Nearest Neighbor
GPS	Global Positioning System
ICP	Iterative Closest Point
IOU	Intersection Over Union
JPDA	Joint Probabilistic Data Association
MHT	Multi Hypothesis Tracking
NDT	Normal Distributions Transform
NEES	Normalized Estimation Error Square
NN	Nearest Neighbor
OSPA	Optimal Subpattern Assignment
PDA	Probabilistic Data Association
PHD	Probability Hypothesis Density
RFS	Random Finite Set
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
SLAM	Simultaneous Localization and Mapping

THE LASER RANGE SENSOR

Notation	Meaning
\mathbb{R}	Set of real numbers
\mathbf{p}_k	Point cloud acquired at time t_k
p_i^k	Point i from \mathbf{p}_k
\mathbf{s}	Sensor position
\mathbf{M}	Environment
T_{p2c}	Transformation from polar to Cartesian coordinates
T_{s2c}	Transformation from spherical to Cartesian coordinates
x	Cartesian x -coordinate
y	Cartesian y -coordinate
z	Cartesian z -coordinate
r	Range
φ	Horizontal angle
ψ	Vertical angle
r_{\max}	Maximum measurable range for laser range sensor
R	Rotation matrix
\mathbf{t}	Translation vector

CLASSIFICATION

Notation	Meaning
C_p	Positive class
C_n	Negative class
\mathbf{f}	Data vector
f_j	Component j of data vector
y	Class label
N_p	Number of training data in positive class
N_n	Number of training data in negative class
$c(\cdot)$	Weak classifier
$\mathbf{c}(\cdot)$	Strong classifier
T	Number of training iterations
w_t^j	Weight of data j at iteration t
t_p	True positive rate
t_n	True negative rate
f_p	False positive rate
f_n	False negative rate
N_p^t	Number of test data in positive class
N_n^t	Number of test data in negative class
D	Detection rate
FA	False alarm rate

ESTIMATION

Notation	Meaning
$\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{z}(t), \mathbf{w}(t), \mathbf{e}(t)$	State, parameter, measurement, process noise, and measurement noise, respectively, at time t
$\mathbf{x}_k, \boldsymbol{\theta}_k, \mathbf{z}_k, \mathbf{w}_k, \mathbf{e}_k$	State, parameter, measurement, process noise, and measurement noise, respectively, at time step k
\mathbf{z}^k	Set of all measurements up to, and including, time step k
T_s	Sampling time
$\hat{\mathbf{x}}_{k \ell}, \mathbf{P}_{k \ell}$	State estimate and covariance at time t_k given measurements up to and including time t_ℓ
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian pdf defined over the random vector \mathbf{x} with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
\mathbf{Q}, \mathbf{R}	Process noise covariance and measurement noise covariance
$p(\mathbf{x}_k \mathbf{z}^\ell)$	Probability density over \mathbf{x}_k given \mathbf{z}^ℓ
$\xrightarrow{c}, \xrightarrow{p}$	Correction and prediction, i.e. measurement update and time update
\mathcal{E}	Estimation error
η	Normalized estimation error square
ρ	Root mean square error
${}^r \mathbf{x}_k$	Robot pose k
${}^r \mathbf{x}_c$	Current robot pose
${}^t \mathbf{x}$	Robot trajectory, i.e. history of poses
\mathbf{M}	Map state, i.e. environment
\mathbf{m}_i	Landmark i

TARGET TRACKING

Notation	Meaning
$\mathbf{x}_k^{(i)}$	State of target i at time k
$N_{x,k}$	Number of targets at time k
\mathbf{X}_k	Set of present targets at time k
$\mathbf{z}_k^{(i)}$	Measurement i at time k
$N_{z,k}$	Number of measurements at time k
\mathbf{Z}_k	Set of measurements at time k
\mathbf{Z}^k	Set of all measurement sets up to time k
\mathcal{H}_i	Hypothesis i
$\mathcal{P}(\cdot)$	Probability
$\bar{d}_p^c(\cdot, \cdot)$	OSPA metric of order p with cut-off c

RANDOM FINITE SETS AND THE PROBABILITY HYPOTHESIS DENSITY

Notation	Meaning
Ξ	Random finite set variable
\mathbf{X}	Finite set realization of Ξ
$ \mathbf{X} $	Number of elements in set \mathbf{X}
\mathcal{X}	Hyperspace of underlying space \mathcal{X}_0
$P_{\mathbf{x}}S$	Probability mass function of random variable \mathbf{x}
$p_{\mathbf{x}}(x)$	Probability density function of random variable \mathbf{x}
$\beta_{\Xi}(S)$	Belief-mass function of random set Ξ
$p_{\Xi}(\mathbf{X})$	Probability density function of random set Ξ
$p_{\Xi}(n)$	Cardinality distribution of random set Ξ
$E[\cdot]$	Expected value
$D_{k \ell}(\mathbf{x})$	Probability hypothesis density
$D_{k+1 k}^b(\mathbf{x})$	Birth PHD
$p_S(\mathbf{x})$	Probability of survival
$p_{k+1 k}(\mathbf{x} \mathbf{x}')$	Transition density
$p_{k+1 k}^s(\mathbf{x} \mathbf{x}')$	Spawning likelihood
$p_D(\mathbf{x})$	Probability of detection
$p_{k+1}(\mathbf{z} \mathbf{x})$	Sensor likelihood function
$c(\mathbf{z})$	Distribution of false alarms
$P_{k \ell}(n)$	Estimated cardinality distribution

EXTENDED TARGET TRACKING

Notation	Meaning
ξ_k	Extended target state at time k
\mathbb{R}^n	Set of real vectors of length n
\mathbb{S}_{++}^d	Set of symmetric positive definite $d \times d$ matrices
\mathbf{x}_k	Random vector at time k
X_k	Random matrix at time k
γ_k	Random measurement rate at time k
$PS(N; \lambda)$	Poisson pmf defined over the random non-negative integer N with rate parameter λ
$IW(X; v, V)$	Inverse Wishart pdf defined over the random matrix X with degrees of freedom v and parameter matrix V
$GAM(\gamma; \alpha, \beta)$	Gamma pdf defined over scalar γ with scalar shape parameter α and scalar inverse scale parameter β

Part I

Background

1

Introduction

This chapter introduces the research topics that are considered in this thesis, and summarizes the research contributions. In Section 1.1 a motivation to the research topics is given, and in Section 1.2 and Section 1.3 the topics are described in more detail. A list of published work is given in Section 1.4, and the main contributions of the thesis are summarized in section 1.5. The chapter is ended with a thesis outline in Section 1.6.

1.1 Motivation

The research presented in this thesis was undertaken at the Division of Automatic Control, Department of Electrical Engineering, at Linköping university. Automatic control is a research area that can be given the following definition, deliberately intended to be as broad as possible:

Definition 1.1 (Automatic control). To automatically make a system behave as desired. _____

In this context *automatically* is to be understood as *without human intervention*. A *system* may refer to anything whose behavior can be controlled, however this thesis will be limited to mobile robots, also called autonomous vehicles.

In the January 2007 issue of the magazine Scientific American, Bill Gates, co-founder and former CEO of Microsoft, predicted that the next hot research field would be robotics (Gates, 2007). About four years later, a free online course in artificial intelligence, given by Stanford University during the fall semester of 2011, attracted more than 58000 students globally (Markoff, 2011). Both these examples serve as a testament to the interest in, and relevance of, robotics research.

Indeed, the past decades have seen a large research effort in the field of robotics. In order for a mobile robot to *behave as desired* in the dynamic and complex world within which humans live, the robot must be aware of itself and its surroundings. In this thesis we refer to this as knowing the state of the robot and the state of its surroundings. The state of the robot includes its location, knowledge of which requires the ability to recognize places that the robot has visited earlier, also called loop detection. The state of the robots' surroundings includes the location of moving objects, called targets, knowledge of which requires the ability to track the targets as they move.

The main research topics considered in this thesis are

1. loop closure detection, i.e. recognizing places that have been visited before;
2. multiple target tracking, i.e. estimating how many targets there are and estimating each target's state.

To solve both these problem, the robot needs to sense the environment around it, similarly to how humans use their five senses¹ to be able to go about their days. In this thesis data from laser range sensors is used for both loop closure detection and target tracking. A reconstruction of 2317 2D laser range scans, acquired inside a shopping mall, is shown in Figure 1.1. Figure 1.2, a small portion of a full scene constructed by 34 3D laser range scans, shows an outdoor environment with some buildings and vegetation.

Laser range sensors are versatile sensors that provide data rich in information content, and the sensor data can be used for many different tasks. In the Defense Advanced Research Projects Agency's (DARPA) Urban Challenge, e.g., laser range sensors were used for task such as staying in lane; maintain vehicle separation; vehicles at an intersection; leaving lane to pass; U-turn; following a vehicle; queuing at an intersection; negotiate obstacle field; road blockages; merging to traffic circle; sparse waypoints (straight); road follow: GPS outages; merging at T intersection; merging at 4-way intersection; left turns at intersections; emergency vehicle avoid; and blocked intersection (Campbell et al., 2007).

1.2 Loop closure detection

Loop closure detection is an important part of the Simultaneous Localization and Mapping (SLAM) problem. The SLAM problem consists of finding out where the robot is (localization), while simultaneously finding out what the surrounding environment looks like (mapping), see e.g. the two part SLAM tutorial by Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006). To solve the SLAM problem the acquired sensor data must be organized such that, when the individual pieces of data are put together, they together constitute a coherent map. However, one of the fundamental properties of the SLAM problem is that small errors, due to sensor inaccuracies, are constantly inserted into the localiza-

¹Vision, hearing, smell, taste, and touch.

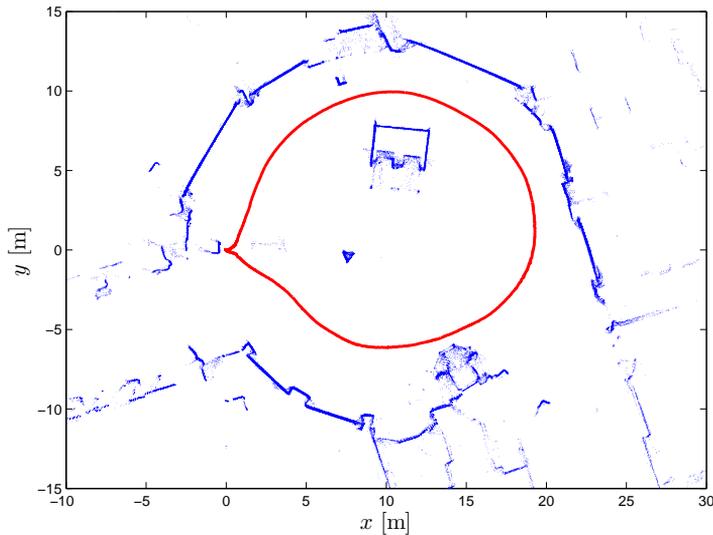


Figure 1.1: Laser range data in 2D, from the ground floor of the shopping mall “Gränden” in central Linköping, Sweden. The laser range data is shown in blue, the robot trajectory is shown in red. Data courtesy of Petter Torle, C3 Technologies.

tion and mapping process. As time passes the errors accumulate, and eventually the map is no longer a good representation of the world. One method to correct this problem is to detect when the robot returns to a place that it has previously visited, i.e. detect that the robot has closed a loop. Because loop closure detection is used to correct errors in the SLAM process, it is of high importance that incorrect, or false, loops are not closed, because this would only increase the errors.

In this thesis the SLAM map consists of individual laser range scans, so called point clouds, acquired at different locations. Loop closure is detected by comparing different point clouds to each other in a pairwise fashion, and classifying them as either being from the same location, or not.

1.3 Multiple target tracking

Multiple target tracking is needed for the robot to be able to move around without constantly running into other objects, and also to make the robot able to follow a moving object while the object moves. The research area target tracking dates back at least to the mid 1900’s, when radar stations were built for the purpose of tracking airplanes, see e.g. the books by Bar-Shalom and Fortmann (1987), Bar-Shalom (1992), Bar-Shalom and Rong Li (1995), Bar-Shalom et al. (2001), and Bar-Shalom et al. (2011). In the typical target tracking scenario it is unknown how many targets there are, it is unknown which target caused which measurement,

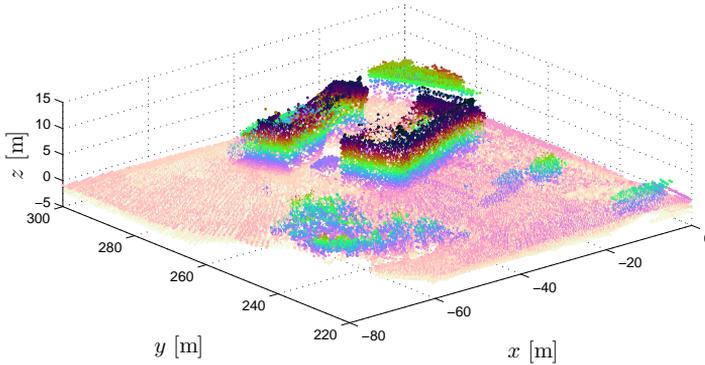


Figure 1.2: Laser range data in 3D, only a portion of the full data set is shown. The scene features a couple of buildings and some vegetation. The data was acquired using laser range sensors mounted on a helicopter. Data courtesy of Piotr Rudol and Mariusz Wzorek, at the Knowledge Processing lab (KPLAB) at the division of Artificial Intelligence and Integrated Computer Systems (AIICS) at the Department of Computer and Information Science (IDA) at Linköping university (LiU).

it is unknown if all targets caused any measurement at all, and there are false, so called clutter, measurements that were not caused by any target at all. In some target tracking scenarios, the target and sensor setup is such that each target generates at most one measurement per time step, in other scenarios each target may generate more than one measurement. In the former case the targets are called point targets, in the latter case the targets are called extended targets.

In this thesis we consider tracking of multiple extended targets, where both the number of targets and each target's state must be found.

1.4 Publications

The following papers, listed in reverse chronological order, have been published:

K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.

K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.

K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012d.

K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012c.

K. Granström, C. Lundquist, F. Gustafsson, and U. Orguner. On extended target tracking using PHD filters. In *Workshop on Stochastic Geometry in SLAM at IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, Minnesota, USA, May 2012.

K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *The International Journal of Robotics Research*, 30(14):1728–1754, December 2011.

U. Orguner, C. Lundquist, and K. Granström. Extended Target Tracking with a Cardinalized Probability Hypothesis Density Filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 65–72, Chicago, IL, USA, July 2011.

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011a.

K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.

K. Granström and T. B. Schön. Learning to Close the Loop from 3D Point Clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, Taipei, Taiwan, October 2010.

K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.

K. Granström, J. Callmer, F. T. Ramos, and J. I. Nieto. Learning to Detect Loop Closure from Range Data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, Kobe, Japan, May 2009.

J. Callmer, K. Granström, J. I. Nieto, and F. T. Ramos. Tree of Words for Visual Loop Closure Detection in Urban SLAM. In *Proceedings of the Australian Conference on Robotics & Automation (ACRA)*, Canberra, Australia, December 2008.

The following paper has been provisionally accepted for publication:

K. Granström and U. Orguner. On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices. *IEEE Transactions on Signal Processing*, 2012e.

The following paper has been revised and resubmitted:

K. Granström and U. Orguner. A New Prediction Update for Extended Target Tracking with Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 2012b.

The following paper has been submitted:

C. Lundquist, K. Granström, and U. Orguner. An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation. *Journal of Selected Topics in Signal Processing*, 2012a.

Finally, the following paper about pedagogic and didactic aspects of undergraduate teaching, has also been published:

C. Lundquist, M. A. Skoglund, K. Granström, and T. Glad. Insights from implementing a system for peer review. *IEEE Transactions on Education*, 2012b. doi: 10.1109/TE.2012.2211876.

1.5 Main contributions

The second part of this thesis contains edited versions of eight of the above listed papers. The scientific contributions contained in these eight papers are summarized in this section.

1.5.1 Loop closure detection

Learning to close loops from range data

Paper A,

K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *The International Journal of Robotics Research*, 30(14):1728–1754, December 2011.

presents a loop closure detection classifier that works for point cloud data in both 2D and 3D. A thorough implementational description is given, and the classifier's properties are evaluated in several different experiments using publicly available data. The pros and cons compared to related work are discussed, and the classifier is shown to compare well to other loop closure detection methods.

1.5.2 Target tracking

Extended Target Tracking using a Gaussian Mixture PHD filter

Paper B,

K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.

presents a Gaussian mixture implementation of an extended target PHD filter. The optimal filter requires a summation over all possible measurement set partitions, which is computationally infeasible in all but the simplest of cases. Suitable partitioning methods are presented, such that the number of partitions that are considered can be kept to a minimum without sacrificing too much tracking performance. The filter is evaluated in both simulations and experiments.

Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements

To keep the presentation simple, the sizes and shapes of the extended targets are not estimated in Paper B. Paper C,

K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.

presents a version of the Gaussian mixture PHD filter that is designed for laser range measurement, with capability to estimate the shape and size of the targets. Two different types of targets are considered, rectangular and elliptical, and the filter is also capable of estimating the target type. Furthermore, the paper shows that the Gaussian mixture PHD filter is not limited to linear motion and measurement models, as in Paper B, it also works for non-linear models. The filter is evaluated in both simulations and experiments.

A PHD filter for tracking multiple extended targets using random matrices

In Paper B and Paper C Gaussian distributions are used to model the extended targets. An alternative to the Gaussian model is to use Gaussian inverse Wishart distributions to model the extended targets, a model in which the extended target shape is assumed to be elliptical. Paper D,

K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.

presents a Gaussian inverse Wishart implementation of the extended target PHD filter. A likelihood function is derived, and the necessary assumptions and approximations are given. Two partitioning methods are presented, in addition to the methods given in Paper B. The filter is evaluated in both simulations and

experiments, and the results show the benefits of estimating the target size and shape, in addition to estimating the position.

Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking

The extended target PHD filters in Paper B, Paper C and Paper D model the number of measurements generated by an extended target as Poisson distributed. In Paper B and Paper D it is noted that correctly setting the filter parameter corresponding to the Poisson rate is necessary in certain circumstances. With an incorrect parameter setting, the filter might estimate the number of targets incorrectly. Paper E,

K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012c.

presents a framework for estimating an individual, possibly time varying, Poisson rate for each extended target. In addition to the already known measurement update, a simple time update is suggested, and a method for mixture reduction is also given. Simulations show that the filter can estimate multiple Poisson rates simultaneously.

On the Reduction of Gaussian inverse Wishart Mixtures

In Paper D a heuristic is used for reduction of Gaussian inverse Wishart mixtures, however it is noted that a better and less approximative method is needed. Paper F,

K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012d.

presents a merging method that can be used for reduction of Gaussian inverse Wishart distribution mixtures. It is shown how a weighted sum of distributions can be approximated with a single distribution, and a criterion is suggested that can be used to determine whether or not two distributions should be merged. Simulations show the merits of the presented merging method.

A New Prediction Update for Extended Target Tracking with Random Matrices

In Paper D it is noted that when two spatially close targets maneuver, the filter often cannot keep the two targets resolved, and as a consequence underestimates the number of targets. Part of the problem is the heuristic prediction method that is used. Paper G,

K. Granström and U. Orguner. A New Prediction Update for Extended Target Tracking with Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 2012b.

presents a prediction method that can handle maneuvering extended targets better, while considering all uncertainty sources. Simulation results show that the presented method outperforms related work on this subject.

On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices

The event that a target launches another target, or that a larger target separates into multiple smaller targets, is called target spawning. The opposite, i.e. that multiple targets merge into a single target, is called target combination. Target spawning is not explicitly modeled in Paper D, however it is noted there that a spawning function could be useful. Paper H,

K. Granström and U. Orguner. On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices. *IEEE Transactions on Signal Processing*, 2012e.

presents spawning and combination of extended targets whose state are modeled as Gaussian inverse Wishart distributed. Limited to the two target case, a model for target combination is first derived. The combination model is then used to derive a model for spawning into two targets. Simulation results show the benefits of the presented functions.

1.6 Thesis outline

The thesis is divided into two parts, with background material in the first part and edited versions of the eight published papers in the second part. It should be noted that while the chapters in the first part give relevant background required for the second part of the thesis, the amount of detail is intentionally kept to a minimum. The reason is that each of the papers in the second part includes background material – repeating this material would cause unnecessary redundancy.

The first part of the thesis is organized as follows. Chapter 2 presents the laser range sensor and the data it produces. Classification, with an emphasis on so called boosting, is the topic of Chapter 3. The machine learning method used in this thesis, called AdaBoost, is presented and illustrated using a number of examples. The estimation problem is introduced in Chapter 4, and filtering solutions and performance metrics are mentioned. Chapter 5 is about the target tracking problem. Data association methods are over-viewed, and performance evaluation is discussed. Random finite sets and the probability hypothesis density filter is introduced in Chapter 6, and extended target tracking is the topic of Chapter 7. The first part of the thesis is ended with Chapter 8, which presents conclusions and discusses future work.

2

The laser range sensor

This chapter presents a brief overview of the laser range sensor and the data it produces. The sensor is described in Section 2.1, and examples of 2D and 3D data are given in Section 2.2 and Section 2.3, respectively. The occlusion problem is described in Section 2.4, and registration of laser range data is discussed in Section 2.5.

2.1 Introduction

In the past 10 year, a vast amount of research has been performed using data from laser range sensors, e.g. mapping, localization and target tracking. There exist different types of laser sensors that produce slightly different types of data, this thesis will be limited to so called sweeping laser sensors. This sensor type works by measuring the distance to the nearest object at different angles, provided that the nearest objects' reflectance properties are good enough. A simulation example of laser range data is given in Figure 2.1.

For an estimation or classification application, the laser range sensor's probabilistic properties need to be modeled. In this thesis, a brief introduction to modeling of the laser range sensor is given. For a more thorough description, chapter 6 in the book by Thrun et al. (2005) is a good starting point. A description of the underlying mechanical and electrical properties goes beyond the scope of this thesis.

In the remainder of this thesis, the output from laser range sensors will be referred to as point clouds, with the following definition:

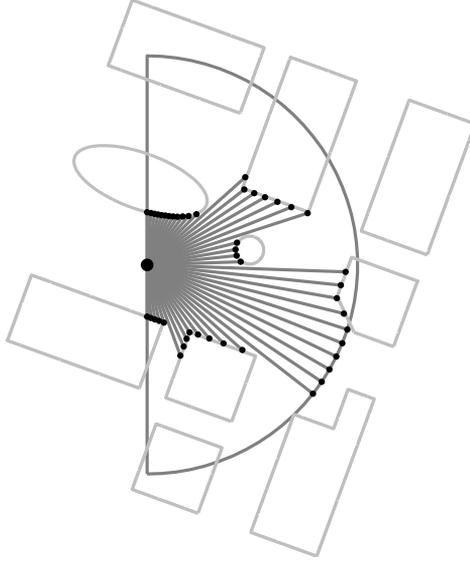


Figure 2.1: Example of laser range data in a 2D simulation environment, where the objects are shown in light gray. The sensor is located at the large black dot, the sensor field of view (180° wide) is shown by the semi-circle. The sensor sweeps right to left, and measures the nearest object every fourth degree. When the nearest object is further away than the boundary of the field of view, the sensor returns maximum range.

Definition 2.1 (Point cloud \mathbf{p}_k). A collection of points in space,

$$\mathbf{p}_k = \{p_i^k\}_{i=1}^N, \quad p_i^k \in \mathbb{R}^D. \quad (2.1)$$

Here, k refers to the acquisition time t_k , N is the number of points p_i^k in the cloud and D is the dimensionality of the data. _____

The name point cloud is inherited from the fact that the sensor measurement defines a point in space which is occupied by an object. It should be noted though that the name point cloud does not capture the so called negative information, i.e. the information about the free space along the laser measurement ray¹. In an application, this negative information about the free-space is important to consider along with the points themselves.

In the applications presented in this thesis, the dimensionality of the data is either $D = 2$ or $D = 3$. Many sensors however, in addition to measuring range, also measure the remission value of the measured point. If the laser range data is fused with image data from a camera, each point may also contain RGB color values. Thus the dimensionality D of the data could be larger than 3. Each measured

¹cf. the gray rays from the sensor to the measurement points in Figure 2.1

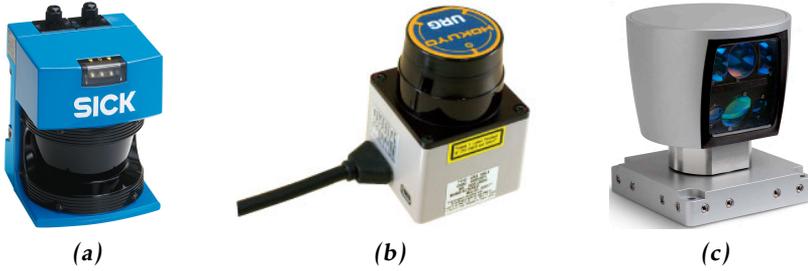


Figure 2.2: Examples of laser range sensors. (a): the SICK LMS200-series. (b): the Hokuyo URG-04Lx-series. (c): the Velodyne HDL-64E. Images are from www.sick.com, www.hokuyo-aut.jp and www.velodyne.com/lidar/, respectively.

point p is a function \mathcal{L} of the sensor position \mathbf{s} and the surrounding environment \mathbf{M} ,

$$p = \mathcal{L}(\mathbf{s}, \mathbf{M}, \mathbf{e}_p), \quad (2.2)$$

where \mathbf{e}_p is random noise. Typically \mathbf{e}_p is modeled as a Gaussian with zero mean and covariance Σ_p ,

$$\mathcal{N}(\mathbf{e}_p; \mathbf{0}, \Sigma_p). \quad (2.3)$$

The properties of laser range sensors vary substantially from sensor to sensor. Maximum measurable range r_{\max} varies from several meters to several kilometers, angular resolution varies from being in the order of one degree to the order of one thousand of a degree. Examples of popular sensors are the LMS200-series sensors manufactured by SICK, see Figure 2.2a, and the sensors manufactured by Hokuyo, see Figure 2.2b. Both these sensors produces planar laser range scans, i.e. they sense the surrounding environment in 2D. Using different pan/tilt units, several 2D scans can be combined to provide 3D laser range data. There are also dedicated 3D laser range sensors, e.g. the HDL-series sensors from Velodyne, see Figure 2.2c.

2.2 Laser range data in 2D

In 2D the points in the point cloud are typically given in polar coordinates as

$$p_i^k = \begin{bmatrix} r_i^k & \varphi_i^k \end{bmatrix}^T, \quad (2.4)$$

where r is the range and φ is the horizontal angle, or bearing, to the measured point. Using the polar to Cartesian transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{T}_{p2c}(r, \varphi) = \begin{bmatrix} r \cos(\varphi) \\ r \sin(\varphi) \end{bmatrix}, \quad (2.5)$$

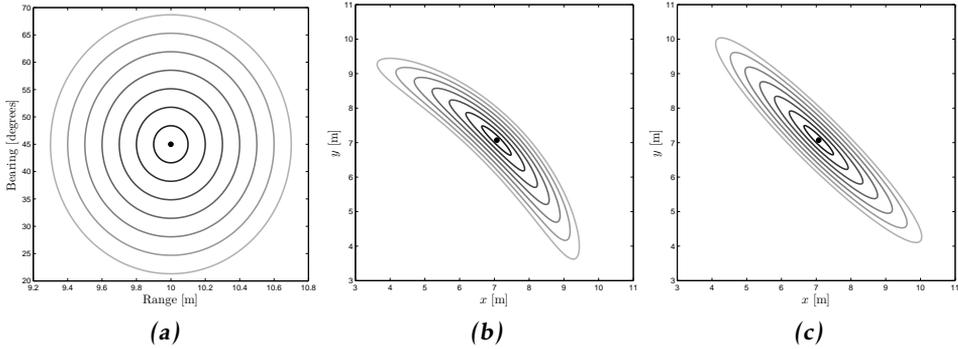


Figure 2.3: Illustration of laser point uncertainty. (a): Uncertainty in polar coordinates. (b): The same uncertainty ellipses as in (a), transformed to Cartesian coordinates using the transform (2.5). (c): First order approximation of transformed uncertainty in (b) using (2.9).

the points can be expressed in Cartesian coordinates as

$$p_i^k = \begin{bmatrix} x_i^k & y_i^k \end{bmatrix}^T. \quad (2.6)$$

The measurement noise covariance matrix is typically a diagonal matrix, where the range and bearing standard deviations can be modeled as functions of range and bearing,

$$\Sigma_p = \begin{bmatrix} \sigma_r^2(r, \varphi) & 0 \\ 0 & \sigma_\varphi^2(r, \varphi) \end{bmatrix}. \quad (2.7)$$

Using the Jacobian J_{p2c} of the polar to Cartesian transformation (2.5),

$$J_{p2c} = \begin{bmatrix} \cos(\varphi) & -r \sin(\varphi) \\ \sin(\varphi) & r \cos(\varphi) \end{bmatrix} \quad (2.8)$$

the covariance can be approximated to first order in Cartesian coordinates as

$$\Sigma_p^c = J_{p2c} \Sigma_p J_{p2c}^T. \quad (2.9)$$

An illustration of the modeled uncertainty is shown in Figure 2.3, where a measurement at range $r = 10\text{m}$ and bearing $\varphi = 45^\circ$, and its corresponding uncertainty, are shown in Figure 2.3a. The results for the non-linear transformation from polar to Cartesian coordinates (2.5) is shown in Figure 2.3b. Compare with the first order approximation in Figure 2.3c.

Figure 2.4 shows a typical outdoor point cloud, both in polar and Cartesian coordinates, Figure 2.4a and Figure 2.4b, respectively. Typical indoor data was shown in Figure 1.1 in Chapter 1. The figures also feature the corresponding 3σ covariance ellipses for every tenth point. For this example, the measurement noise

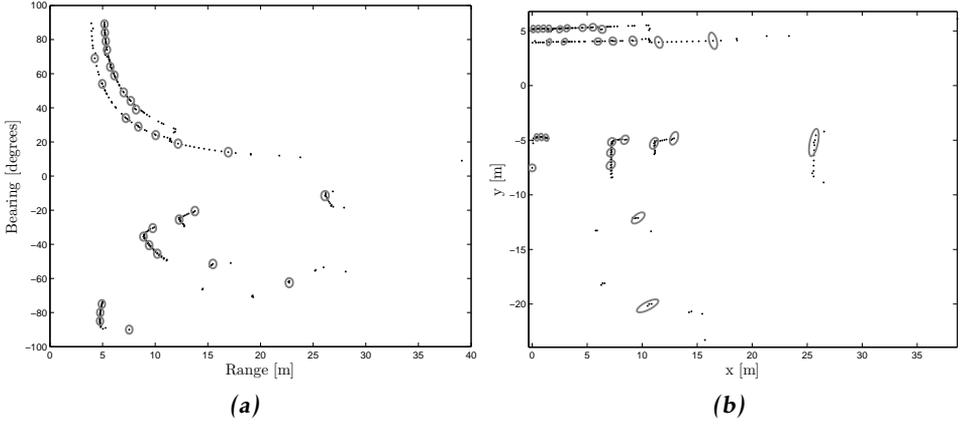


Figure 2.4: Example 2D point cloud, acquired in an outdoor environment. (a): Polar coordinates. (b): Cartesian coordinates. All points measuring range above the maximum range $r_{\max} = 50\text{m}$ have been filtered out. The points are shown in black, for every tenth point the corresponding 3σ covariance ellipse is shown in gray.

covariance is modeled as

$$\Sigma_p = \begin{bmatrix} \sigma_r^2(r, \varphi) & 0 \\ 0 & \sigma_\varphi^2(r, \varphi) \end{bmatrix} = \begin{bmatrix} \left(0.01 \left(1 + \frac{r}{r_{\max}}\right)\right)^2 & 0 \\ 0 & \left(\frac{0.01\pi}{180} \left(1 + \frac{r}{r_{\max}}\right)\right)^2 \end{bmatrix}. \quad (2.10)$$

Thus the uncertainty in range grows larger as the distance to the nearest object along the laser ray grows larger. The bearing noise model can be understood as modeling the laser ray as being shaped as a triangle, with the tip located at the sensor. Thus, the measured point is located on the bottom edge of the triangle.

2.3 Laser range data in 3D

In 3D the points in the point cloud are given in spherical coordinates as

$$p_i^k = \begin{bmatrix} r_i^k & \varphi_i^k & \psi_i^k \end{bmatrix}^T, \quad (2.11)$$

where r is the range, φ is the horizontal angle and ψ is the vertical angle to the measured point. Using the spherical to Cartesian transformation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathcal{T}_{s2c}(r, \varphi, \psi) = \begin{bmatrix} r \sin(\psi) \cos(\varphi) \\ r \sin(\psi) \sin(\varphi) \\ r \cos(\psi) \end{bmatrix}, \quad (2.12)$$

the points can be expressed in Cartesian coordinates as

$$p_i^k = \begin{bmatrix} x_i^k & y_i^k & z_i^k \end{bmatrix}^T. \quad (2.13)$$

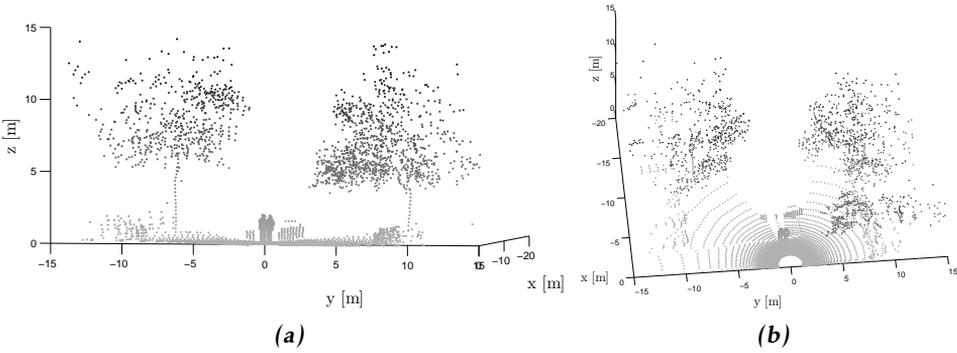


Figure 2.5: Example 3D point cloud in Cartesian coordinates, shown from two different view points in (a) and (b). Grey-scale is used to accentuate height. In the center of the point cloud there are two persons and behind them a car can be seen. To the left and right of the persons and the car there are two trees.

The measurement noise covariance matrix is typically a diagonal matrix, where the range, horizontal angle and vertical angle standard deviations can be modeled as functions of range, horizontal angle and vertical angle,

$$\Sigma_p = \begin{bmatrix} \sigma_r^2(r, \varphi, \psi) & 0 & 0 \\ 0 & \sigma_\varphi^2(r, \varphi, \psi) & 0 \\ 0 & 0 & \sigma_\psi^2(r, \varphi, \psi) \end{bmatrix}. \quad (2.14)$$

Using the Jacobian J_{s2c} of the spherical to Cartesian transformation (2.12),

$$J_{s2c} = \begin{bmatrix} \sin(\psi) \cos(\varphi) & -r \sin(\psi) \sin(\varphi) & r \cos(\psi) \cos(\varphi) \\ \sin(\psi) \sin(\varphi) & r \sin(\psi) \cos(\varphi) & r \cos(\psi) \sin(\varphi) \\ \cos(\psi) & 0 & -r \sin(\psi) \end{bmatrix} \quad (2.15)$$

the covariance can be approximated to first order in Cartesian coordinates as

$$\Sigma_p^c = J_{s2c} \Sigma_p J_{s2c}^T. \quad (2.16)$$

An example point cloud is shown in Figure 2.5. In the figure, gray-scale is used to accentuate height. A 3D point cloud was also shown in Figure 1.2 in Chapter 1.

2.4 Occlusion

Similarly to how a camera needs direct line of sight to the object that is being sensed, so does a laser range sensor. Thus, if an object A is located at the same bearing as another object B , but at larger range than B , then A is occluded by B . Depending on the shape and size of the two objects, A is either partially or fully occluded by B . An example of occlusion is given in Figure 2.6. Occlusion presents

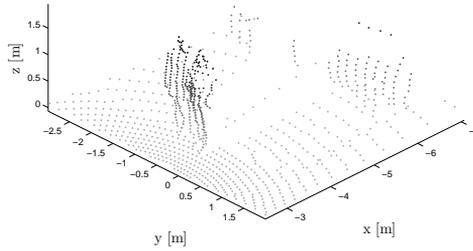


Figure 2.6: Example of the occlusion problem for laser range sensors. The 3D point cloud shows two persons in the foreground, and behind them is a partially occluded vehicle. This point cloud is a portion of the point cloud shown in Figure 2.5. Grey-scale is used to accentuate height, however the gray-scale is different from Figure 2.5.

a considerable challenge in estimation and classification problems where laser range data is used. In the context of target tracking, targets may be lost when they move behind other objects and thus do not generate any measurements. For loop closure detection, occlusions by dynamic objects means that the appearance of the point cloud can be significantly changed.

2.5 Registration

Registration is the process by which two point clouds \mathbf{p}_k and \mathbf{p}_l are fitted to each other with respect to some measure, or cost function, $\mathbf{C}(\mathbf{p}_k, \mathbf{p}_l)$. Typically, the problem is solved by finding a rigid body transformation (R, \mathbf{t}) , where R is a rotation matrix and \mathbf{t} is a translation vector, such that the sum of distances between different point correspondences in the two point clouds is minimized. Point cloud registration is in the literature also referred to as scan matching. Several different methods for finding this rigid body transformation have been suggested, among them the (probably) most popular and well used is the so called Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992; Chen and Medioni, 1992; Zhang, 1994). ICP works by solving the following optimization problem

$$\min_{(R, \mathbf{t})} \mathbf{C}(\mathbf{p}_k, \mathbf{p}_l) = \min_{(R, \mathbf{t})} \sum_{i=1}^{N_k} \sum_{j=1}^{N_l} w_{i,j} \left\| \mathbf{p}_i^k - (R\mathbf{p}_j^l + \mathbf{t}) \right\|^2, \quad (2.17)$$

where $w_{i,j}$ is 1 if point \mathbf{p}_i^k and point \mathbf{p}_j^l describe the same point in space, and 0 otherwise. Finding these point correspondences is typically performed by a nearest neighbor search, and a solution (R, \mathbf{t}) is found by iterating between finding nearest neighbor point pairs and computing the corresponding rigid body transformation. The cost function in (2.17) has many local minimas, and the ICP

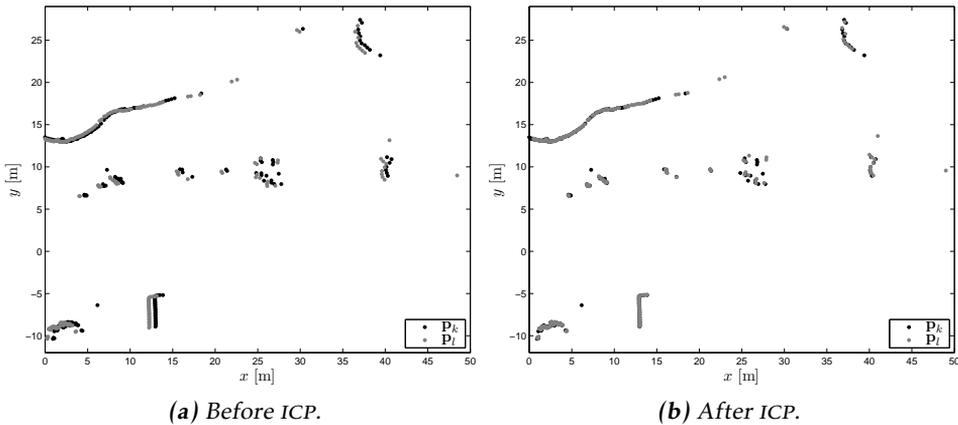


Figure 2.7: Point cloud registration using the ICP algorithm. (a): two point clouds from an outdoor environment before the ICP algorithm is applied. (b): after the ICP algorithm is applied. Note how the rotation and translation that aligns the two point clouds are rather small, thus initializing the ICP algorithm in $(R^0, t^0) = (\mathbf{I}_2, \mathbf{0}_{2 \times 1})$ is sufficient.

algorithm is thus dependent on being initialized in a good point (R^0, t^0) in order to converge. There are a few different ways to implement ICP, and a full overview goes beyond the scope of this thesis. Chapter 4 in the book by Nüchter (2009) is a good starting point for the interested reader.

An illustrative example of the ICP algorithm is given in Figure 2.7, where two point clouds are shown before and after the ICP algorithm is applied. A 3D example of registration is given in Figure 1.2 in Chapter 1, where the point cloud was constructed from 34 smaller point clouds which were registered to each other.

As mentioned above, ICP is a local algorithm in the sense that it, if initialized poorly, often gets stuck in local minimas of the cost function (2.17). To remedy this problem, a method that is able to find a rigid body transformation (R, t) that is close to the ground truth without relying on a good initial guess is needed. Some examples of methods that attempt to improve upon the performance of ICP are the Normal Distributions Transform (NDT) in 2D (Biber and Strasser, 2003) or 3D (Magnusson et al., 2007), CRF-Match in 2D by Ramos et al. (2007), or the approach using histograms in 2D by Bosse and Zlot (2008).

3

Classification

This chapter introduces the classification problem, with an emphasis on boosting methods for finding decision boundaries. The classification problem is defined in Section 3.1, and boosting is presented in Section 3.2. The boosting method of choice, called adaptive boosting, is presented algorithmically, and some key properties of adaptive boosting are highlighted in a series of examples.

3.1 The classification problem

The purpose of a classification method is to take an input data vector

$$\mathbf{f} = [f_1, \dots, f_{n_f}]^T \in \mathbb{R}^{n_f} \quad (3.1)$$

and assign it to one of K classes. Let \mathcal{C}_k denote the class domain, where $k \in \{1, \dots, K\}$ is a class index. In some classification scenarios, the K classes are assumed to be disjoint,

$$\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \quad \forall i \neq j, \quad (3.2)$$

and the input space can therefore be divided into decision regions which are separated by boundaries. These are called decision boundaries, or decision surfaces, see e.g. Bishop (2006). When the decision boundaries are affine functions of the input data \mathbf{f} , the corresponding classifiers are called linear. There are also non-linear classifiers, i.e. classifiers which define decision boundaries that are non-linear functions of the input data. Classes that are disjoint can be separated by linear/non-linear decision boundaries, and are therefore called linearly/non-linearly separable.

However, many problems in classification are neither linearly separable, nor are the true, underlying, data domains \mathcal{C}_k disjoint. For data sets which can not be separated by linear decision boundaries, methods which combine multiple models may be used. Such methods are sometimes called committees, examples include bagging and boosting, see e.g. Bishop (2006). Bagging classifiers are formed by generating M bootstrap data sets from a single data set, and then using each bootstrap data set to train a classifier. Bootstrap data sets are generated by randomly drawing points with replacement from the original data set. Some points from the original data set may thus be drawn more than once in a bootstrap data set, while other points are not drawn at all. The bagging classification is then formed by taking the average of the M bootstrap classifications. In Paper A, boosting, presented in Section 3.2, is used to compute non-linear classifiers.

3.2 Boosting

Boosting is a machine learning method for finding combinations of simple base classifiers in order to produce a form of committee whose performance can be significantly better than any one of the base classifiers used alone. The simple base classifiers need to be just slightly better than a random guess, hence they are often called weak classifiers, see e.g. Bishop (2006). The resulting combination is (typically) better than the best individual weak classifier, and analogously the resulting classifier learned by boosting is thus called strong. The principal difference between boosting and other committee methods such as bagging, is that the training is performed sequentially. Each weak classifier is learned using a weighted form of the data set, where the weighting of each data point depends on the performance of the previous weak classifiers, see e.g. Bishop (2006). There exists a few different boosting methods, here we will limit ourselves to considering adaptive boosting.

3.2.1 Adaptive boosting

A widely used form of boosting is adaptive boosting, abbreviated AdaBoost. It is a machine learning procedure which greedily builds a strong classifier by a linear combination of weak classifiers (Freund and Shapire, 1995). When the weak classifiers are combined into a strong classifier, the resulting decision boundary is non-linear. As more weak classifiers are added, the classification error on the training data converges towards zero, and eventually becomes zero. Although this might be interpreted as over-fitting, AdaBoost has been shown to generalize well on testing data (Freund and Shapire, 1995). A more detailed overview and examination of boosting than can be given here is found in Chapter 10 in the book by Hastie et al. (2009).

Although later generalized to multiple classes, AdaBoost was originally designed for problems with two classes, i.e. $K = 2$. Rather than denoting the two classes as 1 and 2, here they are referred to as the positive class and negative class, or p and n , respectively. As input to the AdaBoost learning algorithm, N hand-labeled

training data pairs are provided,

$$(\mathbf{f}^1, y_1), \dots, (\mathbf{f}^i, y_i), \dots, (\mathbf{f}^N, y_N), \quad (3.3)$$

where each data point \mathbf{f}^i has a corresponding class label y_i . To learn a classifier using AdaBoost, data points from each class are needed. Let N_p and N_n be the number of training data points belonging to \mathcal{C}_p and \mathcal{C}_n , respectively, i.e. $N = N_n + N_p$. The data labels in the two class problem are defined as

$$y_i = \begin{cases} 1 & \text{if } \mathbf{f}^i \in \mathcal{C}_p, \\ 0 & \text{if } \mathbf{f}^i \in \mathcal{C}_n. \end{cases} \quad (3.4)$$

In the AdaBoost algorithm, each data pair (\mathbf{f}^i, y_i) is given a weight w_t^i , where t denotes the specific iteration of the algorithm. The weights are initialized as $w_1^i = \frac{1}{2N_n}$ if $y_i = 0$, or $w_1^i = \frac{1}{2N_p}$ if $y_i = 1$. This initialization ensures that each class is given half the weight of the data, and all data pairs within a class are given an equal weight.

After initialization, AdaBoost iteratively adds weak classifiers to a set of previously added weak classifiers, to find a good combination that together constitutes a strong classifier. The weak classifiers used in this thesis are decision stumps, i.e. one node decision trees, defined as

$$c(\mathbf{f}^i, \theta) = \begin{cases} 1 & \text{if } pf_j^i < p\lambda \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

with parameter $\theta = \{j, p, \lambda\}$, where j is the particular component of \mathbf{f}^i selected, f_j^i , p is the polarity ($p = \pm 1$), and $\lambda \in \mathbb{R}$ is a threshold. The result of a weak classifier (3.5) is that the input space is partitioned into two half spaces, separated by an affine decision boundary which is parallel to one of the input axes.

In each iteration t , the weak classifier that minimizes the weighted classification error with respect to θ is chosen. This is performed by solving an optimization problem. Given the parameters of the best weak classifier θ_t , the training data is classified and the weights of the mis-classified data are increased (or, conversely, the weights of the correctly classified data are decreased). Further, using the classification error ε_t a weight α_t is computed for the best weak classifier. Details on how the weights are computed are given below.

This procedure is repeated until T weak classifiers $c(\mathbf{f}^i, \theta_t)$ have been computed. Weak classifiers can be added several times in each dimension of \mathbb{R}^{n_f} , each time with a new polarity and threshold, i.e. same j and new p and λ . The normalized weighted combination of T weak classifier together create the strong classifier $\mathbf{c}(\mathbf{f}^i)$. The output of the strong classifier is a likelihood, $\mathbf{c}(\mathbf{f}^i) \in [0, 1]$. To obtain a binary classification decision, a threshold $\tau \in [0, 1]$ is used, where the standard choice is $\tau = 0.5$. A detailed presentation of AdaBoost is given in Algorithm 1, and the learning iterations are illustrated in Example 3.1.

Algorithm 1 AdaBoost

Input: Labeled data pairs: $(\mathbf{f}^1, y_1), \dots, (\mathbf{f}^N, y_N)$. Number of training iterations: T .

Initialize weights: $w_1^i = \frac{1}{2N_n}$ if $y_i = 0$, $w_1^i = \frac{1}{2N_p}$ if $y_i = 1$

1: **for** $t = 1, \dots, T$ **do**

2: Normalize the weights:

$$\tilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^{N_n+N_p} w_t^j}, \quad i = 1, \dots, N_n + N_p \quad (3.6)$$

3: Select the best weak classifier with respect to θ ,

$$\theta_t = \arg \min_{\theta} \sum_{i=1}^N \tilde{w}_t^i |c(\mathbf{f}^i, \theta) - y_i| \quad (3.7)$$

4: Define $c_t(\mathbf{f}^i) = c(\mathbf{f}^i, \theta_t)$, and $\varepsilon_t = \sum_{i=1}^N \tilde{w}_t^i |c_t(\mathbf{f}^i) - y_i|$.

5: Update the weights:

$$w_{t+1}^i = \tilde{w}_t^i \beta_t^{1-e_i}, \quad i = 1, \dots, N_n + N_p, \quad (3.8)$$

where $e_i = 0$ if \mathbf{f}^i is classified correctly and 1 otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$. Set

$$\alpha_t = \log \frac{1}{\beta_t}.$$

6: **end for**

Output: Strong classifier

$$\mathbf{c}(\mathbf{f}^i) = \frac{\sum_{t=1}^T \alpha_t c_t(\mathbf{f}^i)}{\sum_{t=1}^T \alpha_t} \in [0, 1] \quad (3.9)$$

Example 3.1: AdaBoost learning

A data set was generated in polar coordinates, where the angle f_φ was sampled uniformly in $[0, 2\pi]$, and the range f_r was sampled from $\mathcal{N}(f_r; 0, 0.10)$ for the positive class and $\mathcal{N}(f_r; 0.50, 0.10)$ for the negative class. Using the transform (2.4), the sampled data was transformed into Cartesian coordinates. Note that since the range components f_r for the two classes are randomly sampled from probability distributions that overlap, the underlying classes are not separable and it is therefore difficult to define a “true” decision boundary.

However, with the known Gaussian distributions for the range components of the data, a probabilistic decision boundary can be defined by considering which class has higher probability in any given data point. Here, the probabilistic decision boundary in the range component is defined as the range for which both classes are equally probable,

$$\frac{1}{\sqrt{2\pi \cdot 0.10^2}} e^{-\frac{(r-0)^2}{2 \cdot 0.10^2}} = \frac{1}{\sqrt{2\pi \cdot 0.10^2}} e^{-\frac{(r-0.50)^2}{2 \cdot 0.10^2}} \Leftrightarrow r = 0.25. \quad (3.10)$$

Thus, for this example, in Cartesian coordinates the probabilistic decision boundary is a circle with radius 0.25. This probabilistic decision boundary is compared with the decision boundary learned by AdaBoost. The data set and a number of

learning iterations are shown in Figure 3.1.

In Figures 3.1b to 3.1f, the decision boundaries of the weak classifiers are shown as black lines and the resulting decision regions are shown in white for \mathcal{C}_p and light purple for \mathcal{C}_n . The increasing weight of the misclassified data is illustrated by making the markers larger.

3.2.2 Examples

In this section we show some of the properties of AdaBoost through some examples. The first two examples are for data from classes that are separable, either linearly or non-linearly. Thus, for these examples true decision boundaries can be defined. The third example is with data from classes that are non-separable, similarly to the data in Example 3.1. The last example bears the largest resemblance to the real world classification problem addressed in this thesis in Paper A.

A very basic requirement for a classification method is that it can handle data which is linearly separable. The AdaBoost algorithm is tested on such data in Example 3.2.

Example 3.2: Linearly separable data

Data points \mathbf{f}^i are generated by uniform sampling in $[0\ 1] \times [0\ 1]$. In total $N = 1000$ data points are generated, and sorted into classes according to

$$\begin{cases} \mathbf{f}^i \in \mathcal{C}_p & \text{if } f_1^i < f_2^i, \\ \mathbf{f}^i \in \mathcal{C}_n & \text{otherwise.} \end{cases} \quad (3.11)$$

Thus, the class regions are well defined in the data space, $\mathbf{f} \in \mathbb{R}^2$, and a true decision boundary can be defined as the line $f_2^i = f_1^i$. Figure 3.2 shows the data and the results. AdaBoost is able to compute a good estimate of the true decision boundary, using $T = 100$ weak classifiers.

For linearly separable data, using AdaBoost can be inefficient from a computational point of view. In Example 3.2 a linear classifier, e.g. Fischer's linear discriminant (see e.g. Bishop (2006)), would be a better choice due to its lower computational demands. An example of data which is separable by a non-linear decision boundary is given in Example 3.3.

Example 3.3: Non-linearly separable data

Data points \mathbf{f}^i are generated by uniform sampling in $[0\ 1] \times [0\ 1]$. In total $N = 1000$ data points are generated, and sorted into classes according to

$$\begin{cases} \mathbf{f}^i \in \mathcal{C}_p & \text{if } |f_1^i - 0.5| \leq 0.25 \text{ AND } |f_2^i - 0.5| \leq 0.25, \\ \mathbf{f}^i \in \mathcal{C}_n & \text{otherwise.} \end{cases} \quad (3.12)$$

Similarly to Example 3.2, the class regions are well defined and the true decision boundary is defined as a square box. The data and the results are shown in Figure 3.3. For the particular example, using just $T = 9$ weak classifiers, AdaBoost finds a good estimate of the true decision boundary.

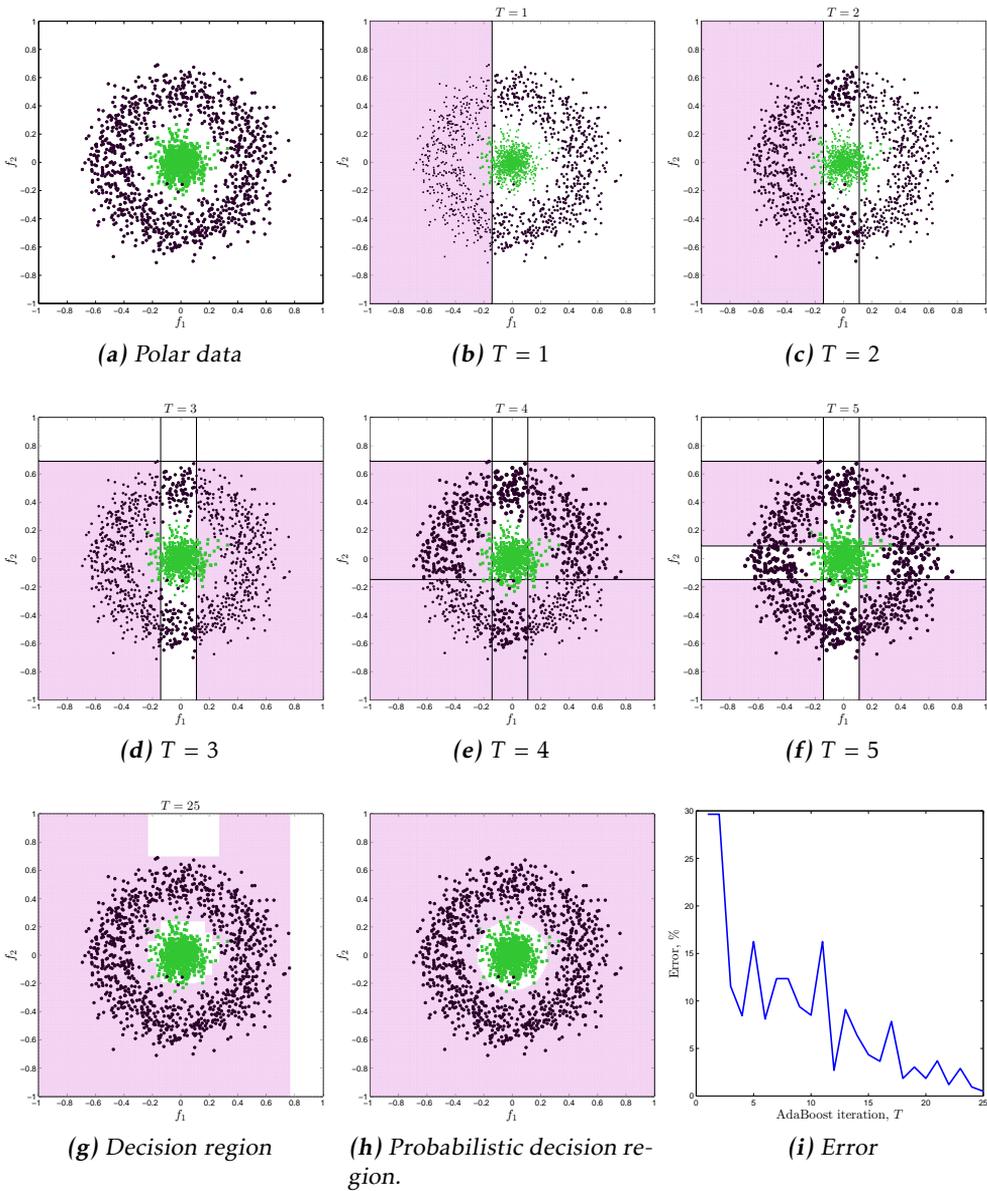


Figure 3.1: AdaBoost learning. (a): polar data set, $\mathbf{f} \in C_p$ in green, $\mathbf{f} \in C_n$ in dark purple. (b) to (f): the first five training iterations. (g) and (h): final decision region, $T = 25$, and true probabilistic decision region. (i): Training error versus number of training iterations T . The error is defined as the percentage of data points that are mis-classified.

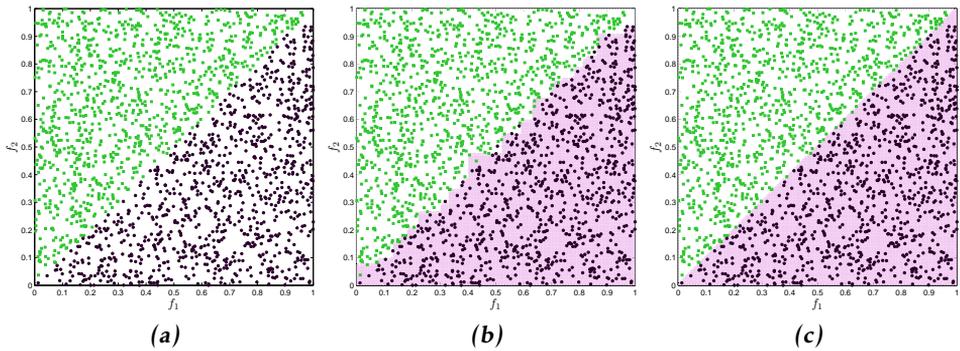


Figure 3.2: AdaBoost with data that is linearly separable. (a): C_p in green, C_n in dark purple. (b): AdaBoost decision region, $T = 100$. (c): True decision region.

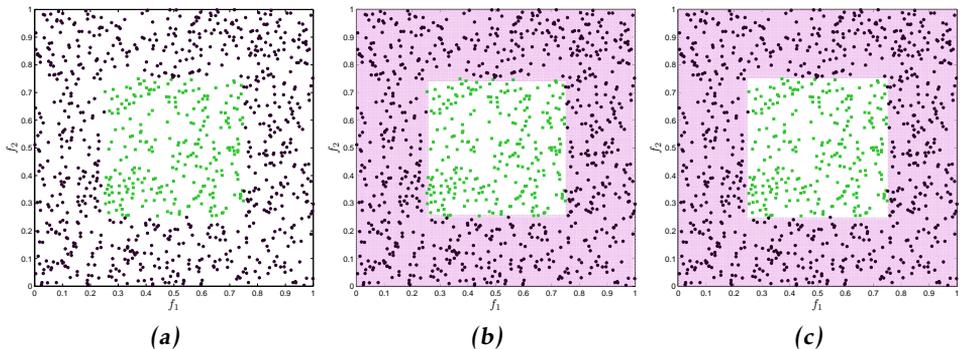


Figure 3.3: AdaBoost with data that is separable by a non-linear decision boundary. (a): C_p in green, C_n in dark purple. (b): AdaBoost decision region, $T = 9$. (c): True decision region.

Most practical classification problems however, are with data sampled from classes that are likely to be non-separable. Example 3.4 presents data, where each class is represented by a Gaussian distribution. Thus, with knowledge of the true underlying class distributions, a decision boundary can be computed from the probability density functions analogously to how a decision boundary was computed in Example 3.1. In the example, there is a large resemblance between the learned decision boundary and the probabilistic decision boundary.

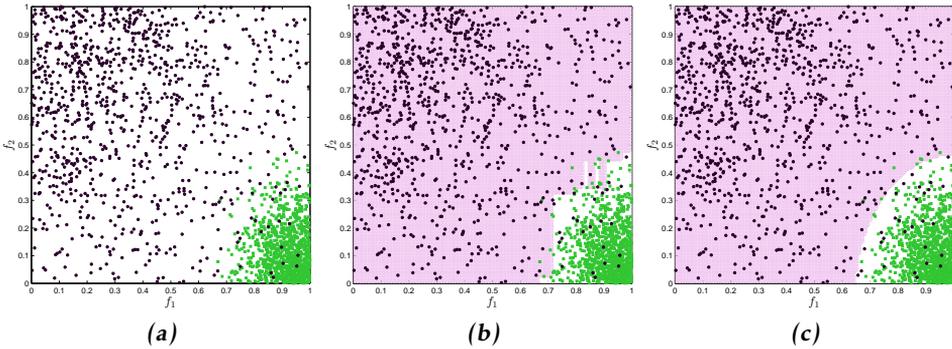


Figure 3.4: AdaBoost with data generated by two Gaussian distributions. (a): C_p in green, C_n in dark purple. (b): AdaBoost decision boundary, $T = 100$. (c): Probabilistic decision region.

Example 3.4: Gaussian data

This example illustrates how AdaBoost finds a decision boundary for data from two non-separable classes. The data points \mathbf{f}^i are generated by sampling from two Gaussian distributions, and keeping only samples that fall in $[0, 1] \times [0, 1]$. Here, $N_p = N_n = 1000$ data points are generated from the following Gaussian distributions

$$\begin{cases} C_p : \text{Samples from } \mathcal{N}\left(\mathbf{f}; \begin{bmatrix} 0.95 \\ 0.05 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 0.02 \end{bmatrix}\right), \\ C_n : \text{Samples from } \mathcal{N}\left(\mathbf{f}; \begin{bmatrix} 0.25 \\ 0.75 \end{bmatrix}, \begin{bmatrix} 0.20 & -0.005 \\ -0.005 & 0.20 \end{bmatrix}\right). \end{cases} \quad (3.13)$$

A probabilistic decision boundary is computed analogously to Example 3.1, i.e. by computing the value of each Gaussian's corresponding probability density function, and for each point in the data space consider which class has higher probability. In Figure 3.4 the data is shown together with the decision boundary learned by AdaBoost and the probabilistic decision boundary.

3.2.3 Properties

Above it was shown that AdaBoost has strong capabilities of finding good non-linear decision boundaries. In each of the three examples however, the number of training data was quite large. In this section, we show what happens when there is few training data available (i.e. N_p and N_n are small), or when the training data are unbalanced ($N_p \ll N_n$ or $N_p \gg N_n$). The important issue of over-fitting is also addressed. Paper A contains experiments where the data is unbalanced, and also contains experiments where the learned classifier is tested for overfitting.

Performance when data is scarce is shown in Example 3.5.

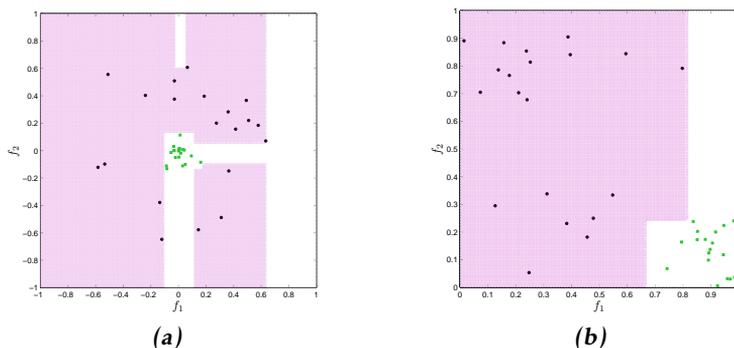


Figure 3.5: AdaBoost when both N_p and N_n are small. A decision boundary has to be learned from an insufficient amount of data. (a): Polar data, $T = 25$. (b): Gaussian data, $T = 3$.

Example 3.5: Few data

Data is generated by randomly selecting $N_p = 20$ and $N_n = 20$ data points from the polar data in Figure 3.1a, and from the Gaussian data in Figure 3.4a. These data sets were previously used in Example 3.1 and Example 3.4, thus the probabilistic decision boundaries are the same as previously. Figure 3.5 shows the results, compare to the true decision boundaries given in Figure 3.1g and Figure 3.4c. It is quite clear from the results that the decision boundary learned by AdaBoost is a poor estimate of the probabilistic decision boundaries. It can be noted though, that such few data gives a rather poor representation of the underlying true distributions, and finding a good decision boundary can be expected to be difficult using any method.

Unbalanced data is presented in Example 3.6, where the number of positive data N_p is 100 times fewer than the number of negative data N_n .

Example 3.6: Unbalanced data

Data is generated by randomly selecting $N_p = 10$ data points from the positive class, and using all $N_n = 1000$ data points from the negative class. As in Example 3.5, both the polar and the Gaussian data sets were used, thus the probabilistic decision boundaries are the same as previously. Figure 3.6 shows the results, compare to the probabilistic decision boundaries given in Figure 3.1g and Figure 3.4c. It is evident that with unbalanced data, AdaBoost no longer finds a decision boundary which resembles the probabilistic one. Instead the learned decision boundary adapts too much to the data, which can be interpreted as over-fitting, an issue which is addressed in the next example.

When learning models for classification, care should be taken to avoid the problem of over-fitting. Over-fitting is when the learned model adapts too much to the

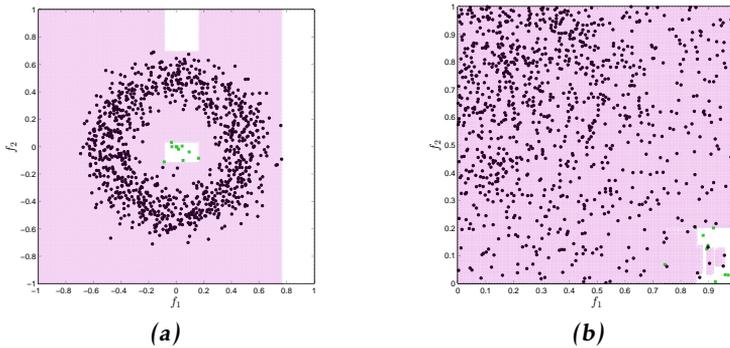


Figure 3.6: AdaBoost when the numbers of data in each class are unbalanced, here $N_p \ll N_n$. (a): Polar data, $T = 25$. (b): Gaussian data, $T = 100$.

training data, and thus does not generalize well to validation data, or the true underlying decision regions. Over-fitting, in the context of AdaBoost, corresponds to using a number of weak classifiers that is too large. Work by Freund and Shapire (1995) has shown that AdaBoost has a strong resistance to over-fitting, indeed experimental results in Paper A confirm this. Attempts have been made to explain AdaBoost, and its reported resistance to over-fitting, in terms of logistic regression, see the paper by Friedman et al. (2000). However, an exhaustive and full technical explanation has to the best of the author's knowledge not been given.

Example 3.7: Overfitting

In this example, the polar data from Example 3.1 and Gaussian data from Example 3.4 were used. For both data sets a classifier was learned using AdaBoost for $T = 1000$ iterations. Figure 3.7 shows the results, compare to the true decision boundaries given in Figure 3.1g and Figure 3.4c.

The results in Example 3.7 show that despite T being excessively large, the resulting decision boundary has not over-fitted to the training data. However, Example 3.5 did show that when data are scarce, the resulting decision boundary adapts too much to training data, i.e. the resistance to overfitting appears to be dependent on the total number of training data.

3.2.4 S-fold cross validation

When solving a classification problem it is important to keep the training data separate from the validation data. If data is scarce, a common practice is to use S -fold cross validation. With this approach, the data is partitioned into S -folds, or subsets. In each of S runs, the S :th fold is reserved for validation, and remaining $S - 1$ folds are used for training. The results from each round are then pooled. This procedure allows a portion $\frac{S-1}{S}$ of the data to be used for training, while the

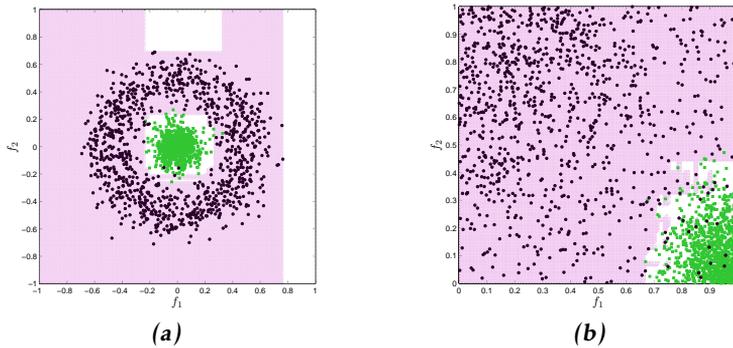


Figure 3.7: Overfitting test. (a): Polar data, $T = 1000$. (b): Gaussian data, $T = 1000$.

whole data set can be used for performance evaluation. While performing S -fold cross validation, it is important to keep the training and validation data fully disjoint. A drawback of S -fold cross validation is that the training procedure has to be repeated S times, which can prove time consuming when the training is computationally expensive, see e.g. Bishop (2006).

A sub-problem of S -fold cross validation is how to partition the data into folds. When data is a sequence over time, training data and validation data can be chosen as different time sequences. This is common practise in e.g. system identification, see e.g. Ljung (1999). In Paper A the data used in classification is not ordered temporally, and thus training and validation data can not be taken as different time sequences. Instead, the data is partitioned into folds by randomly permuting the order of the data, and then dividing the re-ordered data into folds. If the random permutation is performed correctly, each fold should be a good representation of the entire data set. It is also important to consider the labels of the data such that one, or more, of the folds do not represent an unbalanced subset of the whole data set.

3.3 Performance evaluation

This section contains the definition of some quantities related to performance evaluation of binary classifiers. It is shown how these quantities can be used to evaluate classifiers, and compare classifiers to each other. The performance evaluation metrics are used to evaluate and compare classifiers in Paper A.

3.3.1 Basic concepts

For a binary classifier, the true positive rate t_p is the number of positive test data correctly classified as positive. Similarly, the true negative rate t_n is the number of negative test data correctly classified as negative. The false positive

Table 3.1: Binary classifier contingency table.

		Predicted class		Tot.
		p	n	
Actual class	p	True Positive	False Negative	$t_p + f_n = N_p^t$
	n	False Positive	True Negative	$f_p + t_n = N_n^t$
Tot.		$t_p + f_p$	$f_n + t_n$	

rate f_p is the number of negative test data incorrectly classified as positive, and the false negative rate f_n is the number of positive test data incorrectly classified as negative.

Let there be N_p^t and N_n^t number of positive and negative test data, respectively. The four outcomes of a binary classifier can be formulated in a 2×2 contingency table, or confusion matrix, as shown in Table 3.1.

3.3.2 Detection and false alarm

The detection, or true positive, rate D and false alarm, or false positive, rate FA are defined as

$$D = \frac{t_p}{t_p + f_n} = \frac{t_p}{N_p^t}, \quad (3.14a)$$

$$FA = \frac{f_p}{f_p + t_n} = \frac{f_p}{N_n^t}. \quad (3.14b)$$

For an AdaBoost learned classifier, and a given set of test data, the detection and false alarms rates can be computed for different thresholds τ , see Example 3.8.

Example 3.8: Detection and false alarms rates

A set of $N_p^t = N_n^t = 10^4$ test data was generated in the same way as the training data in Example 3.1. Figure 3.8 shows the detection and false alarm rates for different thresholds when the test data is classified using the classifiers learned in Example 3.1 (all training data), Example 3.5 (few training data), Example 3.6 (unbalanced training data), and Example 3.7 (overtraining).

The figure shows that both the detection rate and the false alarm rate decrease with an increasing threshold. This is intuitive, because a higher threshold implies that a higher likelihood is required for the test data point to be classified as belonging to the positive class.

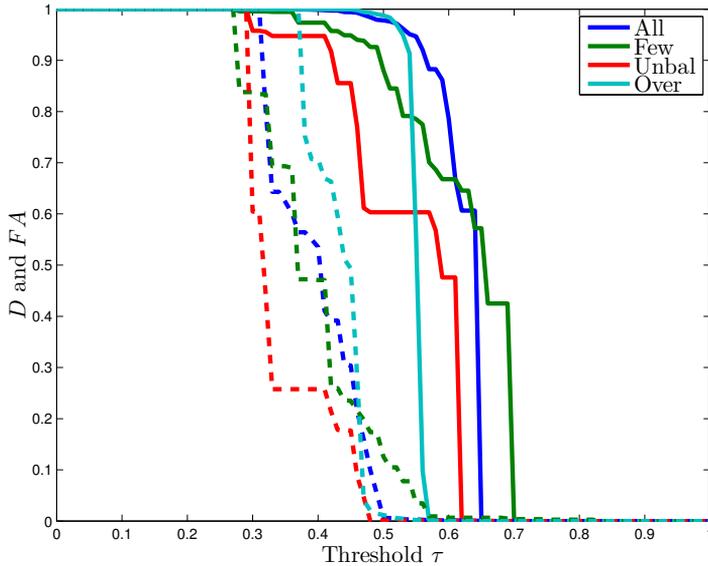


Figure 3.8: Detection (solid) and false alarm (dashed) rates for different thresholds. The legend refers to classifiers learned using the data in Example 3.1 (All), Example 3.5 (Few), Example 3.6 (Unbal), and Example 3.7 (Over).

3.3.3 Receiver operating characteristic

For any binary classifier the detection rate should be high and the false alarm rate should be low. However, as shown in Example 3.8, these two objectives are in conflict. A higher detection rate implies that a lower threshold should be used, but a lower threshold in turn implies a higher false alarm rate. A receiver operating characteristic (ROC) curve is an illustration of a binary classifier's trade off between detection and false alarm. Example 3.9 gives ROC curves for the detection and false alarm curves in Example 3.8.

Example 3.9: Receiver operating characteristic

The detection and false alarm rates in Figure 3.8 are plotted against each other as ROC curves in Figure 3.9. At any given false alarm rate, the detection rate should be as high as possible, which implies that the ROC curve should be as close as possible to the upper left corner of the plot. It can be seen that the overtrained classifier has similar performance as the classifier learned using all data. The classifier learned using few data points has worse performance, but is slightly better than the classifier learned using unbalanced data.

In addition to visually comparing ROC curves in order to compare different classifiers, the area under the ROC curve can be taken as a performance measure. In the best case scenario, the detection rate is 100% for any threshold that gives a

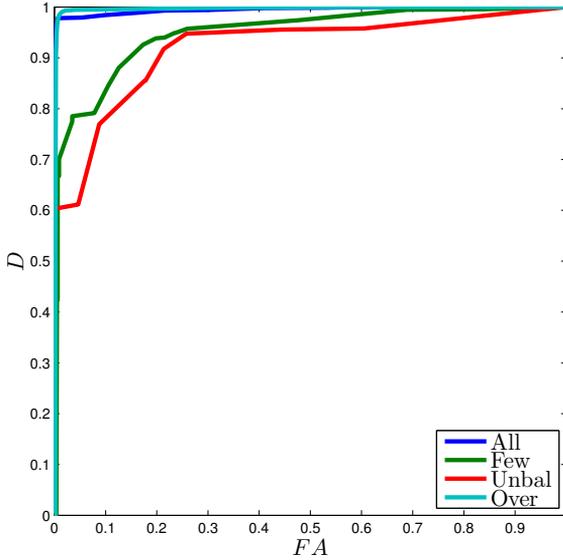


Figure 3.9: ROC curves corresponding to the detection and false alarm curves in Figure 3.8. The legend refers to classifiers learned using the data in Example 3.1 (All), Example 3.5 (Few), Example 3.6 (Unbal), and Example 3.7 (Over).

Table 3.2: Area under the ROC curves in Figure 3.9.

Classifier	All	Few	Unbal	Over
Area under ROC	99.45%	94.85%	92.07%	99.62%

non-zero false alarm rate. In this case the area under the ROC curve would be 1, which means that the larger the area under the ROC curve is, the better the classifier is. The areas under the ROC curves in Figure 3.9 are given in Table 3.2. The results confirm that the classifier learned using the unbalanced training data is the worst, and that the overtrained classifier has equal performance with the classifier learned using all training data.

In certain classification problems it is more important to have a low FA rate than to have a high D rate, one such example is given in Paper A. In this case, different classifiers can be compared by considering the D rate at a specific FA rate. Naturally, the opposite could also be true, i.e. high D rate is more important than a low FA rate. In this case, a comparison of FA rates for specific D rates can be made.

In Table 3.3 the D rates at 0% and 1% FA rate are given for the ROC curves in Figure 3.9. Judging by the D rate at 1% FA rate, the classifier learned using the unbalanced training data is again the worst. However, judging by the D rate at 0% FA rate, the classifier learned using the unbalanced training data is the only one that achieves a non-zero D rate.

Table 3.3: *D* rate at 0% and 1% FA rate for the ROC curves in Figure 3.9.

Classifier	All	Few	Unbal	Over
0% FA	0%	0%	46.94%	0%
1% FA	97.71%	70.75%	60.36%	98.20%

4

Estimation

This chapter is about estimation, a signal processing problem in which an unobserved signal is approximated using an observed signal containing noise. Estimation is an important part of many scientific fields, e.g. sensor fusion and robotics. A common requirement for practical estimation is a mathematical model of the observed and unobserved signals, and the relationship between the signals. In the cases where the mathematical models are linear, and the noise is Gaussian distributed, the Kalman filter is the optimal solution to the estimation problem. In many applications the state-space description is non-linear, for these cases the extended Kalman filter might be an option.

The chapter is organized as follows: the estimation problem is presented and defined in Section 4.1. Section 4.2 is about dynamical models and measurement models, with examples in both continuous and discrete time. Recursive single state Bayes filtering is overviewed in Section 4.3. Linear and non-linear estimation methods are described in Section 4.4 and performance evaluation is presented in Section 4.5.

4.1 The estimation problem

Estimation can, in a general sense, be defined as the problem of approximating, or estimating, a state \mathbf{x} , or a parameter θ , using the noisy measurement \mathbf{z} . In an estimation problem, the quantity of interest, either the state \mathbf{x} or the parameter θ , or the pair \mathbf{x} and θ , is unknown. To keep things simple and uncluttered, in the remainder of the chapter we will assume that it is the state \mathbf{x} that is being estimated. However, note that the presented theory applies equally well to the parameter θ .

Assume, for the sake of simplicity, that the state, parameter and measurement are all vector valued variables, $\mathbf{x} = [x^1, \dots, x^{n_x}]^T \in \mathbb{R}^{n_x}$, $\boldsymbol{\theta} = [\theta^1, \dots, \theta^{n_\theta}]^T \in \mathbb{R}^{n_\theta}$ and $\mathbf{z} = [z^1, \dots, z^{n_z}]^T \in \mathbb{R}^{n_z}$. Estimation can be performed in either continuous or discrete time. Let $\mathbf{x}(t)$ denote the true state in continuous time t , and let \mathbf{x}_k denote the true state at discrete time instant t_k , i.e. $\mathbf{x}_k = \mathbf{x}(t_k)$. In discrete time, let $\hat{\mathbf{x}}_{k|\ell}$ denote the estimate at discrete time t_k , given all measurements up to, and including, time t_ℓ . When $t_\ell > t_k$ ($\ell > k$), the estimation problem is called smoothing, and when $t_\ell < t_k$ ($\ell < k$) the problem is called prediction. However, in this chapter we will limit ourselves to the filtering problem, i.e. when $t_\ell = t_k$ ($\ell = k$).

4.2 Dynamic models and measurement models

To solve the estimation problem it is necessary to model how the estimated quantity \mathbf{x} evolves over time, i.e. to model the state dynamics, which is typically done using differential equations in continuous time. Let

$$\dot{\mathbf{x}}(t) = a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta}(t)) \quad (4.1)$$

denote the dynamic motion model in continuous time. Here, $\dot{\mathbf{x}}(t)$ is the derivative of $\mathbf{x}(t)$ w.r.t. time t , $\mathbf{u}(t)$ is an exogenous input variable, and $\mathbf{w}(t)$ is random noise, often called process noise. Note that estimation of the state \mathbf{x} often assumes that the parameter $\boldsymbol{\theta}$ is known. Often estimation of the state \mathbf{x} cannot be performed in continuous time, instead it has to be performed in discrete time. Let

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}_k) \quad (4.2)$$

be the discrete time counterpart of (4.1). The discrete time steps are related as follows,

$$t_{k+1} = t_k + T_s(k) \quad (4.3)$$

where $T_s(k)$ is the sampling time at time step k . The sampling time can be a function of time, i.e. it can be time varying. However, in the remainder of this chapter we assume constant sampling time and simply write T_s .

In addition to modeling the state dynamics, it is necessary to model the relationship between the measurements \mathbf{z} and the state \mathbf{x} . Let

$$\mathbf{z}(t) = c(\mathbf{x}(t), \mathbf{e}(t), \boldsymbol{\theta}(t)) \quad (4.4)$$

denote the measurement model in continuous time. Here $\mathbf{e}(t)$ is random noise, often called measurement noise. Analogously to the continuous-discrete relationship between (4.1) and (4.2), a measurement model can be given in discrete time as

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{e}_k, \boldsymbol{\theta}_k). \quad (4.5)$$

In the most simple case, the motion and measurement models are both linear, and the process and measurement noises are additive zero mean Gaussian. However, far from all systems can be modeled as linear and Gaussian. In the following, we

will give some simple model examples in continuous and discrete time.

4.2.1 Example in continuous time

Example 4.1 gives a simple linear and noiseless state space system in continuous time.

Example 4.1: Linear state space system, continuous time

Let the state vector contain the one dimensional position p and velocity v of an object, i.e. $\mathbf{x}(t) = [p(t) \ v(t)]^T$. The motion model can be defined as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \begin{bmatrix} \dot{p}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ a(t) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{w}(t) \\ &= A\mathbf{x}(t) + B\mathbf{w}(t)\end{aligned}\tag{4.6}$$

where $\mathbf{w}(t) = a(t)$ is the acceleration. This motion model is often called constant velocity model. Let the measurement be the position, thus the measurement model is

$$\begin{aligned}\mathbf{z}(t) &= z(t) = p(t) \\ &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \\ &= C\mathbf{x}(t).\end{aligned}\tag{4.7}$$

Note that the motion and measurement models (4.6) and (4.7) are modeled as noiseless. This is an atypical choice, because most often systems are assumed to be noisy.

The motion and measurement models used in this thesis are all in discrete time, and in the next section, which presents discrete time models, noise is included. A presentation of random signals in continuous time goes beyond the scope of this thesis, instead we refer the reader to the literature, see e.g. (Jazwinski, 1970).

4.2.2 Examples in discrete time

As was noted above, some estimation problems cannot be solved in continuous time, and instead discrete time models have to be derived for the dynamic motion and measurements. A possible way to discretize a continuous model is to approximate the continuous time derivatives as

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + T_s) - \mathbf{x}(t)}{T_s}\tag{4.8a}$$

$$= \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_s},\tag{4.8b}$$

where T_s is the sample time, and we assume that $t = kT_s$. Note that in the limit, $\lim_{T_s \rightarrow 0}$, the approximation is exact. The approximation (4.8) is also called Euler's approximation. In Example 4.2 Euler's approximation is used to find a discrete time counterpart to the continuous time system presented in Example 4.1.

Example 4.2: Linear Gaussian state space system, discrete time

Using the approximation given in (4.8), the motion model in (4.6) is given in discrete time as

$$\begin{aligned} \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_s} &= \begin{bmatrix} \frac{p_{k+1} - p_k}{T_s} \\ \frac{v_{k+1} - v_k}{T_s} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{w}_k \\ &\Leftrightarrow \\ \mathbf{x}_{k+1} &= \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ T_s \end{bmatrix} \mathbf{w}_k \\ &= F\mathbf{x}_k + G\mathbf{w}_k, \end{aligned} \quad (4.9)$$

with discrete time process noise $p(\mathbf{w}_k) = \mathcal{N}(\mathbf{w}_k; \mathbf{0}, \mathbf{Q}_k)$. The measurement model is given in discrete time as

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \mathbf{e}_k \\ &= H\mathbf{x}_k + \mathbf{e}_k, \end{aligned} \quad (4.10)$$

with discrete time measurement noise $p(\mathbf{e}_k) = \mathcal{N}(\mathbf{e}_k; \mathbf{0}, \mathbf{R}_k)$.

Other continuous to discrete approximations are also possible, see e.g. Gustafson (2010). Depending on the approximation that is used, the discretization of the continuous constant velocity model in Example 4.1 could be different than the one given in Example 4.2.

However, in many cases neither the state dynamics nor the measurements can be modeled accurately as linear systems. Instead non-linear models have to be used. Similarly, the noise processes are not necessarily zero mean Gaussian, but may belong to any other probability distribution. An example of a non-linear, non-Gaussian, state space system is given in Example 4.3.

Example 4.3: Non-linear non-Gaussian state space system, discrete time

A common non-linear motion model is the coordinated turn model with polar velocity. The state is

$$\mathbf{x}_k = \begin{bmatrix} p_k^x & p_k^y & v_k & \phi_k & \omega_k \end{bmatrix}^T, \quad (4.11)$$

where p_k^x and p_k^y are the positions in two dimension, v_k is the velocity, ϕ_k is the heading and ω_k is the turn rate.

The motion model, see e.g. Rong Li and Jilkov (2003), is

$$\mathbf{x}_{k+1} = \begin{bmatrix} p_k^x + \frac{2v_k}{\omega_k} \sin\left(\frac{\omega_k T_s}{2}\right) \cos\left(\phi_k + \frac{\omega_k T_s}{2}\right) \\ p_k^y + \frac{2v_k}{\omega_k} \sin\left(\frac{\omega_k T_s}{2}\right) \sin\left(\phi_k + \frac{\omega_k T_s}{2}\right) \\ v_k \\ \phi_k + \omega_k T_s \\ \omega_k \end{bmatrix} + \mathbf{w}_k, \quad (4.12)$$

where \mathbf{w}_k is random process noise with covariance matrix (Rong Li and Jilkov, 2003)

$$\mathbf{Q}_k = \text{blkdiag}\left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, T_s \sigma_v^2, \begin{bmatrix} \frac{T_s^3 \sigma_\omega^2}{2} & \frac{T_s^3 \sigma_\omega^2}{2} \\ \frac{T_s^3 \sigma_\omega^2}{2} & T_s^3 \sigma_\omega^2 \end{bmatrix}\right). \quad (4.13)$$

The coordinated turn motion model is often used for airplane tracking in air traffic control, and the measurement is then typically the range to the airplane measured by a radar station. If the radar station is located in $\mathbf{s} = [s^x \ s^y]^T$, the measurement model for the state vector in (4.11) is

$$\mathbf{z}_k = \sqrt{(p_k^x - s^x)^2 + (p_k^y - s^y)^2} + \mathbf{e}_k, \quad (4.14)$$

where \mathbf{e}_k is random measurement noise.

4.3 Recursive single state Bayes filter

It is often of interest to use dynamic and measurement models to describe the time evolution of a state. Because of the uncertainties involved, such as process and measurement noise, the knowledge of the state is often described using probability distributions.

The time evolution of the distribution of the state \mathbf{x} can be described in a recursive Bayesian framework. At time step k , assume that we have a prior distribution for the state variable,

$$p(\mathbf{x}_k | \mathbf{z}^k), \quad (4.15)$$

where \mathbf{z}^k is the set of all measurements from time step 0 to time step k

$$\mathbf{z}^k = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{k-1}, \mathbf{z}_k\}. \quad (4.16)$$

The prior can be predicted (i.e. time updated) to the next time step using the Chapman-Kolmogorov equation,

$$p(\mathbf{x}_{k+1} | \mathbf{z}^k) = \int p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}^k) d\mathbf{x}_k, \quad (4.17)$$

where $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ is the transition density from time k to time $k + 1$.

Let $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ be the measurement likelihood. Then, given the predicted prior, the corrected (i.e. measurement updated) posterior is

$$p(\mathbf{x}_{k+1} | \mathbf{z}^{k+1}) = \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{z}^k)}{\int p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{z}^k) d\mathbf{x}_{k+1}}. \quad (4.18)$$

Let $p(\mathbf{x}_0)$ be the prior at the initial time step. The prior $p(\mathbf{x}_0)$, the prediction (4.17), and the correction (4.18), are sufficient to describe the time evolution of the distribution of the state \mathbf{x} given measurements \mathbf{z} ,

$$\begin{aligned} p(\mathbf{x}_0) &\xrightarrow{c} p(\mathbf{x}_0 | \mathbf{z}^0) \xrightarrow{P} p(\mathbf{x}_1 | \mathbf{z}^0) \xrightarrow{c} p(\mathbf{x}_1 | \mathbf{z}^1) \xrightarrow{P} \dots \\ \dots &\xrightarrow{c} p(\mathbf{x}_k | \mathbf{z}^k) \xrightarrow{P} p(\mathbf{x}_{k+1} | \mathbf{z}^k) \xrightarrow{c} p(\mathbf{x}_{k+1} | \mathbf{z}^{k+1}) \xrightarrow{P} \dots \end{aligned} \quad (4.19)$$

where \xrightarrow{P} denotes prediction and \xrightarrow{c} denotes correction.

It is often desired that the propagated distribution over \mathbf{x} has the same functional form, i.e. that $p(\mathbf{x}_k | \mathbf{z}^{k-1})$ and $p(\mathbf{x}_k | \mathbf{z}^k)$ are of the same functional form for all k . For example, if the initial prior is a Gaussian distribution

$$p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; m_0, P_0), \quad (4.20)$$

it is desired that $p(\mathbf{x}_k | \mathbf{z}^{k-1})$ and $p(\mathbf{x}_k | \mathbf{z}^k)$ are Gaussian for all k , i.e.

$$p(\mathbf{x}_k | \mathbf{z}^{k-1}) = \mathcal{N}(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}), \quad (4.21a)$$

$$p(\mathbf{x}_k | \mathbf{z}^k) = \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}). \quad (4.21b)$$

The property that the posterior distribution $p(\mathbf{x}_k | \mathbf{z}^k)$ has the same functional form as the prior distribution $p(\mathbf{x}_k | \mathbf{z}^{k-1})$ is called conjugacy; for a given measurement likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ the prior that gives the same posterior is called conjugate prior.

Worthy of mention for their relevance to this thesis are the conjugate pairs given in Table 4.1. A comprehensive study of conjugate pairs can be found in e.g. the book by Gelman et al. (2004).

Table 4.1: Conjugate prior pairs

Measurement likelihood	Conjugate Prior (variable of interest)
Poisson	Gamma (Poisson rate)
Gaussian	Gaussian (mean vector)
Multivariate Gaussian	Inverse Wishart (covariance matrix)

4.4 Some solutions to the estimation problem

There exists a variety of methods to solve estimation problems, in this section we will briefly review some of them. For linear estimation with Gaussian noise, the Kalman filter provides the optimal solution. For non-linear, non-Gaussian, problems, the extended Kalman filter and the particle filter are two possible methods. The Kalman filter is used in Paper B, Paper D and Paper H, and the extended Kalman filter is used in Paper C and Paper G.

4.4.1 Linear estimation with the Kalman filter

For the case presented in Example 4.2, when the motion and measurement models are linear and the process and measurement noise are Gaussian and independent, the estimation problem can be solved in closed form using the Kalman filter (Kalman, 1960). The Kalman filter propagates in time the first moment $m_{k|k}$ and the second moment $\mathbf{P}_{k|k}$ of the state \mathbf{x}_k ,

$$\begin{aligned} \dots & \xrightarrow{c} \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) & \xrightarrow{p} \mathcal{N}(\mathbf{x}_{k+1}; m_{k+1|k}, P_{k+1|k}) \\ & \xrightarrow{c} \mathcal{N}(\mathbf{x}_{k+1}; m_{k+1|k+1}, P_{k+1|k+1}) & \xrightarrow{p} \dots \end{aligned} \quad (4.22)$$

The Kalman filter prediction and correction equations are given in Algorithm 2. Example 4.4 shows how the Kalman filter is used to estimate the states of the discrete time system given in Example 4.2.

Algorithm 2 Kalman filter

Input: Measurements: $\{\mathbf{z}_k\}_{k=0}^N$. Initial state estimate and covariance: $\{m_0, \mathbf{P}_0\}$. Models: F , G , and H . Parameters: \mathbf{R}_k and \mathbf{Q}_k .

1: **for** $k = 0, \dots, N$ **do**

2: Correction (measurement update)

$$\hat{\mathbf{z}}_{k|k-1} = H m_{k|k-1} \quad (4.23a)$$

$$\mathbf{S}_k = H \mathbf{P}_{k|k-1} H^T + \mathbf{R}_k \quad (4.23b)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} H^T \mathbf{S}_k^{-1} \quad (4.23c)$$

$$m_{k|k} = m_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \quad (4.23d)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k H \mathbf{P}_{k|k-1} \quad (4.23e)$$

3: **if** $k < N$ **then**

4: Prediction (time update)

$$m_{k+1|k} = F m_{k|k} \quad (4.24a)$$

$$\mathbf{P}_{k+1|k} = F \mathbf{P}_{k|k} F^T + G \mathbf{Q}_k G^T \quad (4.24b)$$

5: **end if**

6: **end for**

Output: State estimates and covariances $\{m_{k|k}, \mathbf{P}_{k|k}\}_{k=1}^N$

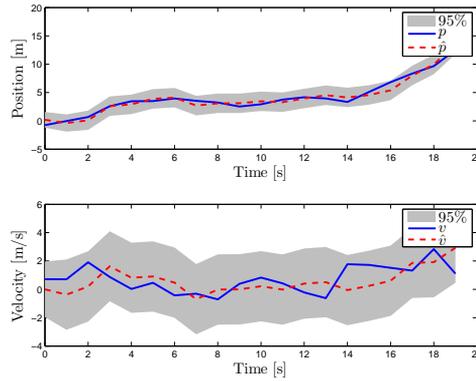


Figure 4.1: Kalman filter state estimates, with 95% confidence intervals.

Example 4.4: Kalman filter

The system in Example 4.2 was initialized at the state $\mathbf{x}_0 = [-0.75, 0.72]^T$, and then simulated with true process noise covariance $\bar{\mathbf{Q}}_k = 1$. $N = 19$ measurements were generated with true measurement noise covariance $\bar{\mathbf{R}}_k = 1$. The solution from the Kalman filter, initialized with

$$\mathbf{m}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (4.25a)$$

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (4.25b)$$

is given in Figure 4.1. In the Kalman filtering the same models were used as were used to simulate the system, and process and measurement noise covariance parameter were set to $\mathbf{Q}_k = \bar{\mathbf{Q}}_k$ and $\mathbf{R}_k = \bar{\mathbf{R}}_k$, respectively.

For certain dynamic motion models, e.g. the constant velocity model in Example 4.2, the state covariance $\mathbf{P}_{k|k}$ will converge to a steady state value, see e.g. Bar-Shalom and Fortmann (1987). In such cases, a class of stationary filters known as α - β filters can be used. The filter recursion consists of the following prediction and correction steps,

$$\mathbf{m}_{k+1|k} = F \mathbf{m}_{k|k}, \quad (4.26a)$$

$$\mathbf{m}_{k+1|k+1} = \mathbf{m}_{k+1|k} + \begin{bmatrix} \alpha \\ \beta \\ \frac{\beta}{T_s} \end{bmatrix} (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}), \quad (4.26b)$$

which, analogously to (4.22), can be expressed as

$$\dots \xrightarrow{c} \mathbf{m}_{k|k} \xrightarrow{p} \mathbf{m}_{k+1|k} \xrightarrow{c} \mathbf{m}_{k+1|k+1} \xrightarrow{p} \dots \quad (4.27)$$

The steady state Kalman filter can be used to determine appropriate values for the parameters α and β , see e.g. Bar-Shalom and Fortmann (1987).

4.4.2 Non-linear estimation

As mentioned above, in many estimation problems the system is neither linear, nor are the noise processes Gaussian. One possible way to solve a non-linear estimation problem is to assume that the process and measurement noises are zero mean Gaussian, and to apply the so called Extended Kalman Filter (EKF), see e.g. Jazwinski (1970). As the name hints, the EKF is an extension of the Kalman filter to non-linear systems. The EKF works by, at each time step, linearizing the non-linear equations around the state estimate via first order Taylor expansion. This linearization introduces linearization errors though. Note that there is no guarantee of convergence of the EKF, however there is much practical experience showing that, if initialized properly, the solution of the EKF often converges.

In the cases when the noise distributions cannot, with reasonable accuracy, be assumed to be Gaussian, the so called Particle Filter (Gordon et al., 1993) is a good alternative to the EKF. In brief, the particle filter provides an approximation of the distribution of the state \mathbf{x}_k conditioned on \mathbf{z}^k . The approximation of the distribution is based on a number of particles, or samples, with associated weights. There are also filters that combine the Kalman filter and the particle filter, these are called Marginalized Particle Filters or Rao-Blackwellized particle filters, see e.g. Schön et al. (2005). In brief, this type of filter is suitable for systems whose state-space models can be separated into linear and nonlinear parts. There are also non-linear filtering methods, good references on non-linear filtering are Simon (2006) and Gustafsson (2010).

4.5 Performance evaluation

When a system is simulated, as in Example 4.4, the true state \mathbf{x}_k is available and can be compared to the state estimate $\hat{\mathbf{x}}_{k|k}$. In contrast to simulations, the true state is often not directly available in experiments, however sometimes an approximation of the true state can be obtained. One such example is outdoor positioning of mobile robots, in which case the position given by a GPS sensor can be used as an approximation of the true position. In other cases the true state can not be approximated, and other more or less subjective performance measures have to be used.

Performance evaluation in the absence of the true state is briefly addressed in Paper A, Paper B, Paper C, and Paper D. In the remainder of this section, it is assumed that the true state is available. When evaluating estimation results, it is important to have a well defined notion of the performance of the estimate. Two performance evaluation methods are presented in this section.

4.5.1 The root mean square error

The estimation error \mathcal{E}_k is defined as

$$\mathcal{E}_k = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}. \quad (4.28)$$

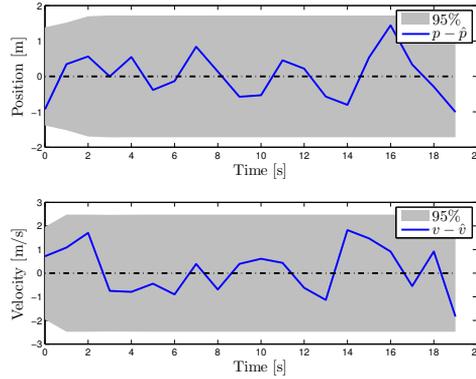


Figure 4.2: Kalman filter estimation errors, with 95% confidence intervals.

The estimation error is thus a vector of the same dimension as the underlying state \mathbf{x}_k , i.e. $\mathcal{E}_k = [\varepsilon_k^1, \dots, \varepsilon_k^{n_x}]^T \in \mathbb{R}^{n_x}$, and each component of the estimation error has the same unit as the corresponding state. If the states are position and velocity, as in Example 4.2, the components of the estimation error are given in meters and meter per second, respectively, see Example 4.5.

Example 4.5: Kalman filter estimation errors

The estimation errors corresponding to the Kalman filter results in Example 4.4 are given in Figure 4.2. The figure shows the estimation errors and the 95% confidence intervals.

A standard performance metric for the estimation error is the root mean square error (RMSE) ρ . Given a time sequence of states \mathbf{x}_k , and the corresponding state estimates $\hat{\mathbf{x}}_{k|k}$, the RMSE's are defined, for each component of the estimation error vector, as

$$\rho^i \triangleq \sqrt{\frac{1}{N} \sum_{k=1}^N (\varepsilon_k^i)^2}. \quad (4.29)$$

Note that the summation is taken over time for each component of the estimation error vector. The RMSE of the estimation error \mathcal{E}_k , i.e. the Euclidean norm of the vector, can be difficult to interpret because often the states have different dimensions¹. An exception to this is when the state vector contains multiple states of the same dimension, in which case an RMSE can be calculated in each time step for those states².

When simulated data is used, Monte Carlo techniques can be used to realize the system with different process and measurement noise sequences. In such a case, the RMSE can be evaluated at each time step over the different simulations. Let

¹cf. Example 4.2: what is the unit of a sum of position squared and velocity squared?

²When \mathbf{x} contains x -position and y -position, the 2D Euclidean norm of \mathbf{x} is the position error.

\mathcal{E}_k^m be the estimation error at time step k for the m :th Monte Carlo simulation run. The RMSE at each discrete time step k is then calculated, for each component i of the estimation error vector, as

$$\rho_k^{i,\text{MC}} = \sqrt{\frac{1}{n_{\text{MC}}} \sum_{m=1}^{n_{\text{MC}}} (\varepsilon_k^{i,m})^2}. \quad (4.30)$$

where $\varepsilon_k^{i,m}$ is component i of \mathcal{E}_k^m , and n_{MC} is the number of Monte Carlo runs. Note that since the estimation error can be negative, calculating the mean estimation error should be avoided. Evaluating the RMSE over time may be of interest when the true target track contains different types of motion, e.g. linear motion and curving motion. In such cases, it is often difficult to model both types of motion using just one motion model.

Note that techniques exist that find the estimate that minimizes the squared RMSE, the so called minimum mean square error (MMSE) estimate $\hat{\mathbf{x}}^{\text{MMSE}}$, see e.g. Bar-Shalom and Fortmann (1987).

4.5.2 The normalized estimation error square

An alternative to the estimation error is the normalized estimation error square (NEES) η_k , defined as

$$\eta_k \triangleq (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T \mathbf{P}_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}). \quad (4.31)$$

The NEES can be understood as being a weighted average of the individual state errors, where the weights are given by the inverse state covariance. Thus, if the variance of a state estimate is high, its inverse weight will be small and a large error will have a smaller contribution to the NEES. Conversely, if the variance of a state is low, its inverse weight will be large and a large error will have a larger contribution to the NEES. Note that the NEES, in contrast to the estimation error and RMSE, is a dimensionless quantity.

When the state estimate $\hat{\mathbf{x}}_k$ is Gaussian distributed, the NEES can be shown to be χ^2 -distributed with n_x degrees of freedom (Bar-Shalom et al., 2001). Thus, using the $\chi^2(n_x)$ -distribution, probability gates can be computed for the NEES. Similarly to (4.30), the NEES can be averaged over Monte Carlo runs. The NEES however, is always positive by definition, and thus the sum can be calculated over the Monte Carlo runs,

$$\eta_k^{\text{MC}} = \sum_{j=1}^{n_{\text{MC}}} \eta_k^j = \sum_{j=1}^{n_{\text{MC}}} (\mathbf{x}_k^j - \hat{\mathbf{x}}_{k|k}^j)^T (\mathbf{P}_{k|k}^j)^{-1} (\mathbf{x}_k^j - \hat{\mathbf{x}}_{k|k}^j). \quad (4.32)$$

The Monte Carlo NEES sum η_k^{MC} is χ^2 -distributed with $n_x \times n_{\text{MC}}$ degrees of freedom. Probability gates g_γ^{max} and g_γ^{min} , corresponding to $\gamma\%$ of the probability mass, can be computed using the $\chi^2(n_x n_{\text{MC}})$ -distribution. For a given discrete time sequence, η_k^{MC} should be within $[g_\gamma^{\text{min}}, g_\gamma^{\text{max}}]$ $\gamma\%$ of the time instances. If it is not, it is a sign that the estimates may be inconsistent. In practical implementa-

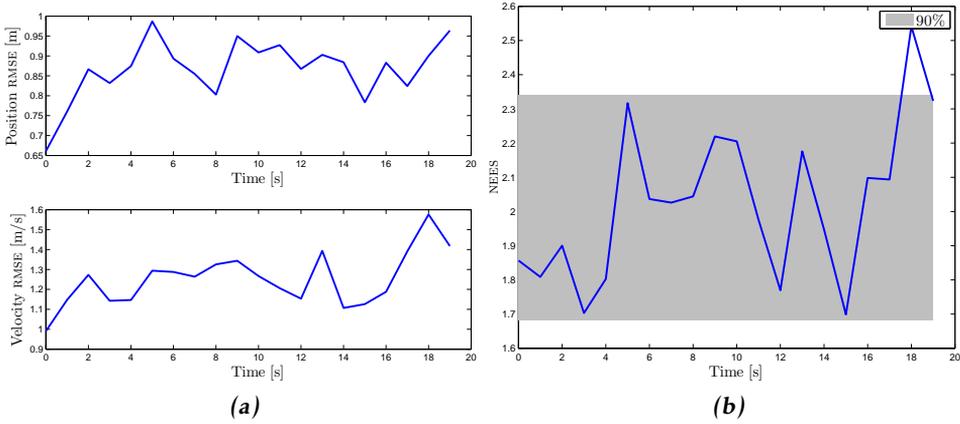


Figure 4.3: Estimation performance evaluation. **(a):** the RMSE for the position and velocity, respectively. **(b):** the NEES with 90% confidence interval.

tions, η_k^{MC} is often divided by the number of Monte Carlo runs n_{MC} . When this is performed, g_γ^{max} and g_γ^{min} must also be divided by n_{MC} . Example 4.6 shows the Monte Carlo average RMSE and NEES for the system in Example 4.4.

Example 4.6: RMSE and NEES

The system presented in Example 4.4 is simulated with 100 unique process and measurement noise sequences, and the Kalman filter was used to compute state estimates. The corresponding RMSE and NEES are shown in Figure 4.3. The NEES is within the 90% confidence interval in 19 out of 20 time steps, or 95% of the time steps.

4.6 Simultaneous localization and mapping

Filtering can be applied to solve a broad variety of problems. An example that is of high relevance to this thesis, is the Simultaneous Localization and Mapping (SLAM) problem. The SLAM problem is a robotics problem that consists of the joint estimation of the robot state ${}^r\mathbf{x}$ and the map state \mathbf{M} . Thus, the full state vector is

$$\mathbf{x} = \begin{bmatrix} {}^r\mathbf{x} \\ \mathbf{M} \end{bmatrix}. \quad (4.33)$$

The robot state typically consists of position and orientation, which is also called the robot pose. The map is often divided into landmarks, sometimes called features, and thus the map state \mathbf{M} can be decomposed as

$$\mathbf{M} = \left[(\mathbf{m}^{(1)})^\top \quad \dots \quad (\mathbf{m}^{(i)})^\top \quad \dots \quad (\mathbf{m}^{(m)})^\top \right]^\top, \quad (4.34)$$

where $\mathbf{m}^{(i)}$ is the i :th landmark state. The landmark state is often given as an (x, y) or (x, y, z) position, however the orientation can also be included. In some SLAM applications the robot trajectory, i.e. a history of robot poses, is of interest and therefore included in the state vector. In the estimation, the estimated quantity is the trajectory state

$${}^t\mathbf{x} = \begin{bmatrix} {}^r\mathbf{x}_0^\top & {}^r\mathbf{x}_1^\top & \dots & {}^r\mathbf{x}_k^\top & \dots & {}^r\mathbf{x}_K^\top & {}^r\mathbf{x}_c^\top \end{bmatrix}^\top, \quad (4.35)$$

where ${}^r\mathbf{x}_k$ is the k th robot pose and ${}^r\mathbf{x}_c$ is the current robot pose. When both the robot trajectory and the whole map is estimated, the problem is called Full-SLAM. The Full-SLAM state vector is

$$\mathbf{x} = \begin{bmatrix} {}^t\mathbf{x} \\ \mathbf{M} \end{bmatrix}. \quad (4.36)$$

Estimating the robot trajectory ${}^t\mathbf{x}$, and not the landmarks in the map state \mathbf{M} , is called trajectory based SLAM. In this case, the state vector is simply $\mathbf{x} = {}^t\mathbf{x}$. In trajectory based SLAM, instead of measuring the landmarks in the map state \mathbf{M} , the robot measures the relative difference between the current pose ${}^r\mathbf{x}_c$ and some previous pose ${}^r\mathbf{x}_k$. A simple example of trajectory based SLAM in 2D is given in Example 4.7.

Example 4.7: Trajectory based SLAM

A true trajectory was simulated,³ and control inputs and measurements were obtained from the simulation. The robot pose is

$${}^r\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \phi_k \end{bmatrix}^\top, \quad (4.37)$$

where (x_k, y_k) is the position and ϕ_k is the orientation. The relative pose

$${}^r\mathbf{x}_{k,c} = \begin{bmatrix} x_{k,c} & y_{k,c} & \phi_{k,c} \end{bmatrix}^\top \quad (4.38)$$

is defined as a rigid body transformation that transforms ${}^r\mathbf{x}_k$ to ${}^r\mathbf{x}_c$. For a pose defined as in (4.37), the relationship between ${}^r\mathbf{x}_k$, ${}^r\mathbf{x}_c$ and ${}^r\mathbf{x}_{k,c}$ is (Smith et al., 1990)

$${}^r\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ \phi_c \end{bmatrix} = \begin{bmatrix} x_k + x_{k,c} \cos(\phi_k) - y_{k,c} \sin(\phi_k) \\ y_k + x_{k,c} \sin(\phi_k) + y_{k,c} \cos(\phi_k) \\ \phi_k + \phi_{k,c} \end{bmatrix} \quad (4.39a)$$

$$= \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \cos(\phi_k) & -\sin(\phi_k) & 0 \\ \sin(\phi_k) & \cos(\phi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k,c} \\ y_{k,c} \\ \phi_{k,c} \end{bmatrix} \quad (4.39b)$$

$$= {}^r\mathbf{x}_k + \begin{bmatrix} R(\phi_k) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} {}^r\mathbf{x}_{k,c}, \quad (4.39c)$$

where $R(\cdot)$ is a rotation matrix, and $\mathbf{0}_{m \times n}$ is an $m \times n$ all zero matrix.

³The trajectory was simulated using the EKF-SLAM toolbox written by Tim Bailey and Juan I. Nieto, Australian Centre for Field Robotics (ACFR), University of Sydney (USYD), Australia.
<http://www-personal.acfr.usyd.edu.au/tbailey/software/index.html>

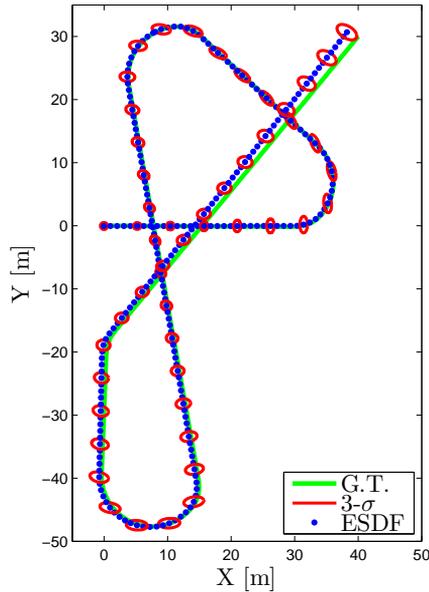


Figure 4.4: Trajectory based SLAM. The true trajectory is given by the line, estimated poses by the dots and the 99% pose uncertainty is shown by the ellipses for every fifth pose.

In this example, measurements are generated when the true current pose is sufficiently close to a previous pose, i.e. when the robot closes a loop. The measurements are of the true relative pose ${}^r\mathbf{x}_{k,c}$, with additive white Gaussian noise. The probability of detecting loop closure was set to one. In reality however, detecting loop closure presents a considerable challenge, and the probability of detection is less than one.

A SLAM trajectory was estimated using an Exactly Sparse Delayed state Filter (ESDF) (Eustice et al., 2006), which is an EKF on information form for state vectors of the type (4.35). The results are shown in Figure 4.4.

The origins of modern SLAM research can be traced back to the mid 80's, when probabilistic methods were first applied to mobile field robotics (Durrant-Whyte and Bailey, 2006). Early work of large impact includes the paper by Smith et al. (1990), which showed that when a robot makes relative observations of landmarks, the estimates of the landmarks are all correlated. This implied that a consistent full solution to the SLAM problem would require a joint state consisting of both robot state ${}^r\mathbf{x}$ and map state \mathbf{M} . A nice overview of SLAM research is given in the two part tutorial by Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006).

A number of different solutions to the SLAM problem have been proposed. In on-line solutions, the data is considered incrementally, i.e. processed one at a time, while in offline solutions the data is typically considered in batch, i.e. all data is processed at the same time, see e.g. Thrun and Leonard (2008). Popular solutions to the SLAM problem include EKF-SLAM, see e.g. Dissanayake et al. (2001), which, as the name reveals, solves the problem using an EKF. Another solution is FAST-SLAM (Montemerlo et al., 2002), which is based on the particle filter. FAST-SLAM has been shown to suffer from particle depletion when the robot's mission grows longer in time, which results in inconsistent estimates (Bailey et al., 2006). There are, however, many practical examples where FAST-SLAM has provided good results. A third family of solutions to the SLAM problem are the graph-based solutions, pioneered by Lu and Milios (1997). The graph-based solutions solve the SLAM problem offline in batch, either as trajectory based SLAM, or as Full-SLAM.

An important part of any SLAM solution is the data association, i.e. associating measurements to the right landmark estimates, or associating relative pose estimates to the correct previous pose. Data association in SLAM is very important, because incorrect associations can lead to inconsistent SLAM-estimates. In Paper A data association for trajectory based SLAM is posed as a loop closure detection problem, and a detection classifier is learned using AdaBoost.

5

Target tracking

This chapter is about target tracking, which is a type of estimation problem. The target tracking problem is defined in Section 5.1, and some common data association methods are presented in Section 5.2. The optimal sub-pattern assignment metric for target tracking performance evaluation is defined in Section 5.3.

5.1 The target tracking problem

In this thesis, we will focus on targets that are moving objects, such as airplanes, cars and humans. Early target tracking research was motivated by, among other things, tracking of airplanes using radars, see e.g. Bar-Shalom and Fortmann (1987). When airplanes are tracked using radar stations, the distance between the sensor and target is often such that the target only occupies one resolution cell of the sensor. Due to this, each target generates at most one radar measurement. Because the targets generate at most one measurement per time step, they effectively behave as points in the surveillance space and can thus be modeled as such. This leads to the point target assumption, with the following definition:

Definition 5.1 (Point target). A target that gives rise to at most one measurement per time step. _____

In the following two subsections some properties of target tracking are listed, and problem formulations are given for both the case of a single target, and the case of multiple targets.

5.1.1 Single target tracking

Single target tracking can be defined as the processing of measurements in order to maintain an estimate of a target's current state. As in any estimation problem,

it holds that

I each target generated measurement is corrupted by noise.

However, when the estimation problem was introduced to the reader it was implicitly assumed that in each time step there is a single state generated measurement. In contrast to this, single target tracking is complicated by the fact that

II the probability of detection for each target is less than one, i.e. in each time step it is not known whether or not the target generated a measurement,

III there are false, so called clutter, measurements, and

IV measurement origin is unknown, i.e. it is not known which measurements are target generated, and which measurements are clutter.

Each time step, a sensor delivers $N_{z,k}$ measurements $\mathbf{z}_k^{(j)}$. Let the set of measurements at time k be denoted

$$\mathbf{Z}_k = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}}. \quad (5.1)$$

Further, let \mathbf{Z}^k be all sets of measurements from time 1 to time k , i.e. $\mathbf{Z}^k = \{\mathbf{Z}_i\}_{i=1}^k$. The objective of single target tracking is to use \mathbf{Z}^k to determine whether or not there is a target present, and if a target is present, to estimate the target state \mathbf{x}_k .

5.1.2 Multiple target tracking

Multiple target tracking is an extension of single target tracking, and can be defined as the processing of multiple measurements obtained from multiple targets in order to maintain estimates of the targets' current states, see e.g. Bar-Shalom and Fortmann (1987). At the heart of multiple target tracking lies a joint estimation problem, namely estimating the number of targets, and estimating the states of each target. In addition to the properties I to IV listed above, in multiple target tracking

V the number of present targets is unknown.

Let $N_{x,k}$ denote the unknown number of targets present at time k , and let $\mathbf{x}_k^{(i)}$ denote the state of target i at time k . At time k the set of all present targets \mathbf{X}_k is given by

$$\mathbf{X}_k = \left\{ \mathbf{x}_k^{(i)} \right\}_{i=1}^{N_{x,k}}. \quad (5.2)$$

The objective of multiple target tracking is to estimate \mathbf{X}_k given \mathbf{Z}^k , i.e. to determine how many targets there are, and for the targets that are present, to estimate the target states $\mathbf{x}_k^{(i)}$.

5.2 Data association methods

An important part of solving a target tracking problem is solving the data association problem, sometimes also referred to as the correspondence problem. Data association means to associate each measurement to one of the measurement generating sources, i.e. either to a target or a clutter source.

Data association is an integral part of target tracking, because incorrect data association can result in disastrous tracking performance. In this section, we will briefly overview some data association methods often used for point target tracking. In each time step, each measurement is either clutter, or generated by a target. For the measurements that are generated by targets, a decision has to be made as to which measurements belong to already existing targets, and which measurements belong to new targets. Handling the data association problem is easier when the number of targets is limited to at most one. Hence, we will review single target tracking association methods first before reviewing association methods for multiple target tracking.

5.2.1 Single target tracking

When at most one target is present, i.e. single target tracking, the data association problem comes down to deciding if there is a target present, and if so, which measurement belongs to the target. Since there is at most one target present, the association can be handled locally, i.e. only the measurements closest to the target estimate are considered as potentially having been generated by the target.

Nearest neighbor

In nearest neighbor (NN) data association, the target is associated to the nearest measurement such that

$$\left(\mathbf{z}_k^{(i)} - \hat{\mathbf{z}}_{k|k-1} \right)^T S_k^{-1} \left(\mathbf{z}_k^{(i)} - \hat{\mathbf{z}}_{k|k-1} \right) \quad (5.3)$$

is minimized. Here, the notation $\hat{\mathbf{z}}_{k|k-1}$ and S_k was introduced in Algorithm 2, and i is an index which spans over all measurements that fall within the gate. A gate, or validation region, is a part of the measurement space where the specific measurement is found with some (high) probability (Bar-Shalom and Fortmann, 1987). Gating is used as a means to reduce the risk that the target is corrected using a clutter measurement. Each measurement has a probability of being target generated, and a probability of being clutter. The measurements that fall inside the gate have a probability of being target generated that is relatively high.

One of the more common gating methods is ellipsoidal gating, which checks whether the quantity in (5.3) is larger or smaller than some gate threshold g_γ , which corresponds to $\gamma\%$ of the probability mass.¹ Ellipsoidal gating thus corresponds to an ellipsoidal region in the measurement space. Note that NN makes a hard decision in the sense that only the measurement that minimizes (5.3) is

¹The quantity in (5.3) is χ^2 -distributed, cf. the normalized estimation error square in (4.31).

considered, the remaining measurements that fall inside the gate are ignored in the correction step.

Probabilistic data association

Probabilistic data association (PDA) is a soft version of NN. No hard decision is made at all, instead all measurements that fall inside the gate are used to the extent that they suit the prediction. Let $\left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k}$ be the measurements that fall within the gate. The following hypotheses are considered,

$$\mathcal{H}_0 : \text{All of } \left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k} \text{ are clutter} \quad (5.4a)$$

$$\mathcal{H}_j : \mathbf{z}_k^{(j)} \text{ was target generated and } \left\{ \mathbf{z}_k^{(i)} \right\}_{i \neq j} \text{ are clutter. } \quad j = 1, \dots, m_k \quad (5.4b)$$

Using the total probability theorem, the probability $\mathcal{P}\left(\mathbf{x}_k \left| \left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k} \right.\right)$ is calculated as

$$\mathcal{P}\left(\mathbf{x}_k \left| \left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k} \right.\right) = \sum_{j=0}^{m_k} \mathcal{P}\left(\mathbf{x}_k \left| \mathcal{H}_j, \left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k} \right.\right) \mathcal{P}\left(\mathcal{H}_j \left| \left\{ \mathbf{z}_k^{(i)} \right\}_{i=1}^{m_k} \right.\right). \quad (5.5)$$

Details on how PDA is implemented can be found in the literature, see e.g. Bar-Shalom and Fortmann (1987).

5.2.2 Multiple target tracking

In multiple target tracking, i.e. when more than one target may be present, data association is more complicated, and using local methods is insufficient. Instead a global association decision must be made, meaning that all measurement-to-target-estimate associations have to be considered jointly.

Global nearest neighbor

Global nearest neighbor (GNN) data association considers all measurement-to-clutter/existing track/new track associations, and selects the best overall hypothesis. In an implementation, the clutter and new target tracks are typically handled by so called track initiators, and they are therefore combined into a category called external sources. At each time step, an association matrix containing measurement-to-source likelihoods is formed, and the assignment problem is then solved as a convex optimization problem, e.g. using the auction algorithm (Blackman and Popoli, 1999). While being global, and thus superior to NN, GNN represents a hard decision for each measurement and only one data association hypothesis is thus considered. In some more complex scenarios, making a hard decision may be insufficient.

Joint probabilistic data association

Joint probabilistic data association (JPDA) is a soft version of GNN analogously to how PDA is a soft version of NN. Measurement-to-target probabilities are computed jointly over all targets, and only measurements from the last time step are

considered, see e.g. Bar-Shalom and Fortmann (1987).

Multiple hypothesis tracking

Multiple hypothesis tracking (MHT) is a data association method that considers association of sequences of measurements and evaluates the probabilities of all hypotheses. Quickly, the number of possible hypotheses grows very large, and therefore methods to reduce the number of hypotheses have been suggested. These include clustering to reduce combinatorial complexity, pruning of low probability hypotheses and merging of similar hypotheses, see e.g. Bar-Shalom and Rong Li (1995).

5.3 Performance evaluation

In Section 4.5 performance evaluation methods for the estimation problem were presented. While state estimation is a central part of target tracking, in many scenarios performance indicators, such as RMSE and NEES, can not be applied directly to a target tracking problem. Some typical difficulties are highlighted in Example 5.1.

Example 5.1: Multiple target tracking data association difficulties

Let the true number of targets be 2, and let each state consist of Cartesian x and y position. Consider the scenarios given in Figure 5.1. Note that the index numbers for the estimates are not used as track labels. In Figure 5.1a the targets and the estimates can be arranged in vectors according to the respective index numbers, and the total position RMSE can be computed in a straightforward manner as the Euclidean norm of the estimation error. In Figure 5.1b, the indexing of the targets and the estimates does not coincide, and a straightforward computation of the RMSE does not make sense. In Figures 5.1c and 5.1d, the problem is complicated further because the number of targets is either under- or overestimated.

Thus, a multiple target tracking performance evaluation method must capture both the error in the estimated number of targets, as well as the state estimation error. Further, the performance evaluation should consider, globally, which estimate is associated to which target. One such method is the optimal subpattern assignment (OSPA) metric (Schuhmacher et al., 2008). Let

$$d^{(c)}(\mathbf{x}^{(k)}, \mathbf{x}^{(l)}) \triangleq \min(c, d(\mathbf{x}^{(k)}, \mathbf{x}^{(l)})) \quad (5.6)$$

be the distance between $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(l)}$, cut off at $c > 0$. Here $d(\cdot, \cdot)$ is any metric, in target tracking typically the Euclidean metric. Let $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ and $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(n)}\}$, where $m, n \in \{0, 1, 2, \dots\}$. Let Π_k denote the set of permutations on $\{1, 2, \dots, k\}$ for any $k \in \{1, 2, \dots\}$. For example, if $k = 3$ then Π_3 is

$$\Pi_3 = \{[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]\}. \quad (5.7)$$

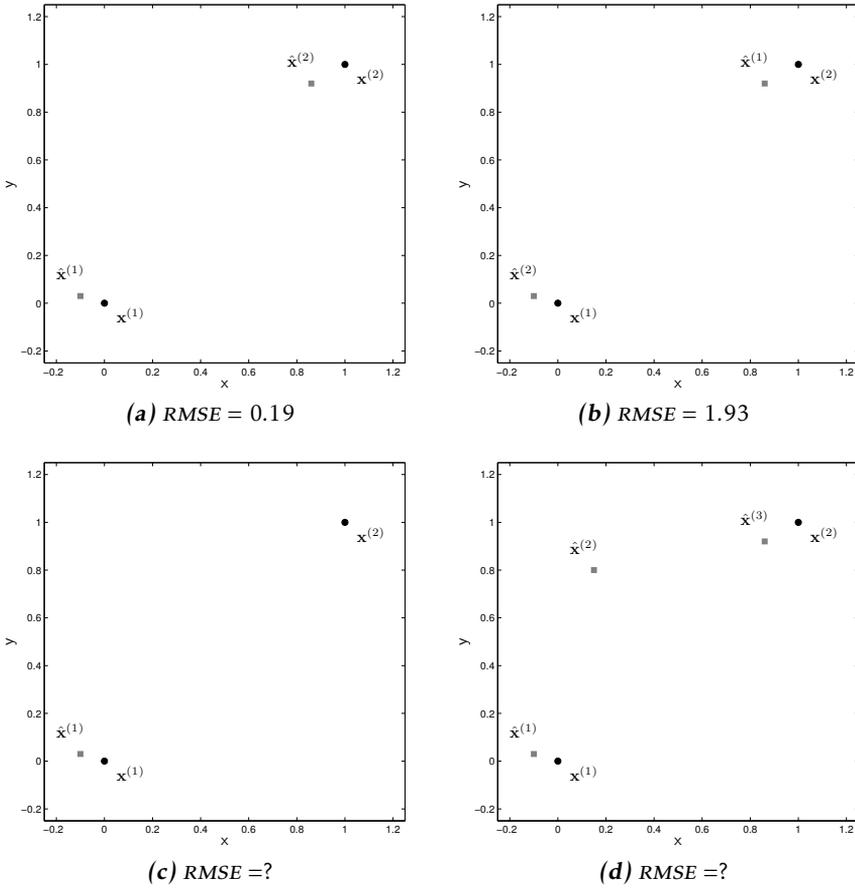


Figure 5.1: Multiple target tracking data association difficulties. There are two true targets, marked with black dots, and the estimated targets are marked with gray squares.

Define $\bar{d}_p^{(c)}(\mathbf{X}, \hat{\mathbf{X}})$, called the OSPA metric of order p with cut-off c , as

$$\bar{d}_p^{(c)}(\mathbf{X}, \hat{\mathbf{X}}) \triangleq \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(\pi(i))})^p + c^p (n - m) \right) \right)^{\frac{1}{p}} \quad \text{if } m \leq n \quad (5.8a)$$

$$\bar{d}_p^{(c)}(\mathbf{X}, \hat{\mathbf{X}}) \triangleq \bar{d}_p^{(c)}(\hat{\mathbf{X}}, \mathbf{X}) \quad \text{if } m > n \quad (5.8b)$$

$$\bar{d}_p^{(c)}(\mathbf{X}, \hat{\mathbf{X}}) \triangleq 0 \quad \text{if } m = n = 0 \quad (5.8c)$$

Note that if $d(\cdot, \cdot)$ is the Euclidean metric, $n = m$, $p = 2$, $c = \infty$ and the optimal permutation is $\pi(i) = i$, the OSPA $\bar{d}_2^{(\infty)}$ reduces to the RMSE multiplied with $\sqrt{n_x}$. It is shown by Schuhmacher et al. (2008) that (5.8) is indeed a metric. Similarly

to how a minimum mean square error estimate can be found, see Section 4.5, a minimum mean OSPA (MMOSPA) estimate $\hat{\mathbf{x}}^{\text{MMOSPA}}$ can be found (Guerriero et al., 2010). In Example 5.2 the OSPA metric is evaluated for the scenarios presented in Example 5.1.

Example 5.2: OSPA metric

For the multiple target tracking scenarios given in Figure 5.1, the OSPA metric, evaluated with $c = 0.5$ and $p = 2$, is

$$\text{(a): } \bar{d}_2^{(0.5)}(\mathbf{X}, \hat{\mathbf{X}}) = 0.14 \quad (5.9a)$$

$$\text{(b): } \bar{d}_2^{(0.5)}(\mathbf{X}, \hat{\mathbf{X}}) = 0.14 \quad (5.9b)$$

$$\text{(c): } \bar{d}_2^{(0.5)}(\mathbf{X}, \hat{\mathbf{X}}) = 0.36 \quad (5.9c)$$

$$\text{(d): } \bar{d}_2^{(0.5)}(\mathbf{X}, \hat{\mathbf{X}}) = 0.31 \quad (5.9d)$$

6

Random finite sets and the probability hypothesis density

This chapter is about random finite sets and the probability hypothesis density. Random finite sets and multi-target calculus are reviewed in Section 6.1. These mathematical concepts can be used to derive a recursive multi-state Bayes filter. Multi-state Bayes filters is the topic of Section 6.2, with an emphasis on the probability hypothesis density filter.

6.1 Introduction

When the estimation problem was presented, see Chapter 4, the notion of a random vector \mathbf{x} was implicitly assumed to be known to the reader. In this section the reader is introduced to the perhaps less familiar concept of a random finite set, and is also given a short overview of multi-target calculus.

6.1.1 Random finite sets

In the previous chapter both the measurements and targets were seen as sets, see (5.1) and (5.2). Considering the properties I to V listed in Section 5.1, in multiple target tracking it becomes suitable to model the numbers of elements in the measurement and target sets as random variables, and to model all elements in the sets as random variables.

A random finite set (RFS) is a set where each element in the set is a random variable, and where the number of elements in the set is a non-negative integer valued random variable. Mahler (2007b) defines random finite set as follows.

Definition 6.1 (Random finite set). A random variable Ξ that draws its instantiations $\Xi = \mathbf{X}$ from the hyperspace \mathcal{X} of all finite subsets \mathbf{X} (the null set \emptyset included) of some underlying space \mathcal{X}_0 . _____

One of the benefits of using RFS:s to model multiple target tracking is that doing so simplifies the performance evaluation, as highlighted in the examples in Section 5.3. The number of elements in an RFS is also called the set cardinality, and is denoted $|\mathbf{X}|$. Note that an RFS \mathbf{X} is without ordering, which implies that

$$\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\} = \{\mathbf{x}^{(2)}, \mathbf{x}^{(1)}\} \quad (6.1)$$

for $|\mathbf{X}| = 2$. The RFS concept is used for multiple target tracking in Paper B to Paper H. In Example 6.1 we give an RFS example whose elements are random state vectors.

Example 6.1: Random finite set in Euclidean vector space

In multiple target tracking the target states can typically be defined as vectors in a Euclidean vector space \mathbb{R}^{n_x} , e.g. this is the case in the examples in Chapter 4.

Let the underlying state space be a Euclidean vector space, i.e. $\mathcal{X}_0 = \mathbb{R}^{n_x}$. Then the hyperspace \mathcal{X} consists of the following finite sets:

$$\begin{aligned} \mathbf{X} &= \emptyset \\ \mathbf{X} &= \mathbf{x}^{(1)} \in \mathbb{R}^{n_x} \\ &\vdots \\ \mathbf{X} &= \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_x)}\}, \quad \mathbf{x}^{(i)} \in \mathbb{R}^{n_x} \quad \forall i \\ &\vdots \end{aligned} \quad (6.2)$$

where $N_x \geq 0$ is the set cardinality.

Thus, in multiple target tracking the target set (5.2) can be modeled as a random finite set $\mathbf{X}_k \in \mathcal{X}$, where $\mathcal{X}_0 = \mathbb{R}^{n_x}$. This has a simple interpretation: there could be any number of targets present, and the state of each present target is a vector in \mathbb{R}^{n_x} .

6.1.2 A brief overview of multi-target calculus

The probability mass function and probability density function are two important concepts in the context of random variables. In this section the RFS generalizations of these concepts, given by Mahler (2007b), are introduced.

For a random variable $\mathbf{x} \in \mathcal{X}_0$ the probability mass function $p_{\mathbf{x}}(S)$ gives the probability of \mathbf{x} being in some part $S \subseteq \mathcal{X}_0$,

$$P_{\mathbf{x}}(S) = \Pr(\mathbf{x} \in S). \quad (6.3)$$

The probability density function $p_{\mathbf{x}}(x)$ describes the relative likelihood of \mathbf{x} to occur at a given point x , and it is related to the probability mass function by an integral

$$P_{\mathbf{x}}(S) = \int_S p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \quad (6.4)$$

and a derivative

$$p_{\mathbf{x}}(x) = \left. \frac{dP_{\mathbf{x}}(S)}{d\mathbf{x}} \right|_{S=x}. \quad (6.5)$$

The probability mass function for a random vector can be generalized to the belief mass function for an RFS. The belief-mass function is denoted $\beta_{\Xi}(S)$, and is defined as the probability that the random finite set Ξ on \mathcal{X}_0 is within some region S ,

$$\beta_{\Xi}(S) = \Pr(\Xi \subseteq S). \quad (6.6)$$

Similarly, the probability density function for a random vector can be generalized to the probability density function $p_{\Xi}(\mathbf{X})$ of a random finite set Ξ .

A multi-state density function on the underlying space \mathcal{X}_0 is a real valued function $p(\mathbf{X})$ of a finite subset variable $\mathbf{X} \subseteq \mathcal{X}_0$ such that, if \mathcal{X}_0 has a unit of measurement u , the unit of measurement of $p(\mathbf{X})$ is $u^{|\mathbf{X}|}$. Furthermore a multi-state density function $p(\mathbf{X})$ is a multi-state probability density function if $p(\mathbf{X}) \geq 0$ for all \mathbf{X} and if

$$\int p(\mathbf{X}) \delta \mathbf{X} = 1. \quad (6.7)$$

The probability density function of an RFS Ξ is, if it exists, the function $p_{\Xi}(\mathbf{X})$, such that

$$\int_S p_{\Xi}(\mathbf{X}) \delta \mathbf{X} = \Pr(\Xi \subseteq S), \quad \forall S. \quad (6.8)$$

The relation between $p_{\Xi}(\mathbf{X})$ and $\beta_{\Xi}(S)$ is thus given by a set integral

$$\beta_{\Xi}(S) = \int_S p_{\Xi}(\mathbf{X}) \delta \mathbf{X} \quad (6.9)$$

and a set derivative

$$p_{\Xi}(\mathbf{X}) = \left. \frac{\delta \beta_{\Xi}(S)}{\delta \mathbf{X}} \right|_{S=\emptyset}. \quad (6.10)$$

Please refer to the book by Mahler (2007b) for the definitions of the set integral and the set derivative. Multi-target distributions can also be expressed in vector notation, e.g.

$$p_{\Xi}(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = 2p_{\mathbf{x}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}), \quad (6.11)$$

since the probability density must be distributed equally over the two possible vectors $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$ and $[\mathbf{x}^{(2)}, \mathbf{x}^{(1)}]$. In general it holds that

$$p_{\Xi}(\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}) = n! p_{\mathbf{x}}(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}). \quad (6.12)$$

As mentioned earlier in the chapter, the cardinality of an RFS is a random variable, and it is necessary to model its distribution. The cardinality distribution of the

finite random set Ξ is

$$p_{\Xi}(n) \triangleq p_{|\Xi|}(n) \quad (6.13a)$$

$$\triangleq \Pr(|\Xi| = n) \quad (6.13b)$$

$$= \frac{1}{n!} \int_{\mathcal{X}_0} p(\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}) d\mathbf{x}^{(1)} \dots d\mathbf{x}^{(n)}. \quad (6.13c)$$

Let $p(n)$ be a probability distribution on the non-negative integers and let $p_{\mathbf{x}}(\mathbf{x})$ be a probability density function on \mathcal{X}_0 . For any $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$ with $|\mathbf{X}| = n$, define

$$p_{\Xi}(\mathbf{X}) \triangleq n! p(n) \prod_{i=1}^n p_{\mathbf{x}}(\mathbf{x}^{(i)}). \quad (6.14)$$

An i.i.d. cluster process is any random finite set Ξ that has $p_{\Xi}(\mathbf{X})$ as its distribution, for some choice of $p(n)$ and $p_{\mathbf{x}}(\mathbf{x})$. In Example 6.2 a type of i.i.d. cluster process called Poisson process is given.

Example 6.2: Multitarget Poisson process

If $p(n)$ is the Poisson probability mass function with rate parameter γ ,

$$p(n) = \mathcal{PS}(n; \gamma) \quad (6.15a)$$

$$= \frac{\gamma^n e^{-\gamma}}{n!}, \quad (6.15b)$$

then the RFS \mathbf{X} with pdf

$$p_{\Xi}(\mathbf{X}) = e^{-\gamma} \gamma^n \prod_{i=1}^n p_{\mathbf{x}}(\mathbf{x}^{(i)}) \quad (6.16)$$

is a multi-target Poisson process.

The concepts i.i.d. cluster process and Poisson process can be used to derive practical recursive multi-state Bayes filters, e.g. the probability hypothesis density filter and the cardinalized probability hypothesis density filter, which are given in the next section.

For a more in depth description of multi-target calculus, please refer to the book by Mahler (2007b).

6.2 Recursive multi-state Bayes filter

In Section 4.3 we described the time evolution of the probability distribution of a single target. In this section we generalize this to the multi-target case using the concepts and tools introduced previously in this chapter. For the sake of simplicity and brevity, we drop the subscript Ξ from the multi-target pdf, i.e.

$$p(\mathbf{X}) = p_{\Xi}(\mathbf{X}). \quad (6.17)$$

At time step k , assume that we have a posterior distribution for the multi-target set,

$$p(\mathbf{X}_k | \mathbf{Z}^k). \quad (6.18)$$

The posterior can be predicted (time updated) to the next time step using the set equivalent of (4.17),

$$p(\mathbf{X}_{k+1} | \mathbf{Z}^k) = \int p(\mathbf{X}_{k+1} | \mathbf{X}_k) p(\mathbf{X}_k | \mathbf{Z}^k) \delta \mathbf{X}_k, \quad (6.19)$$

where $p(\mathbf{X}_{k+1} | \mathbf{X}_k)$ is the multi-target transition density from time k to time $k+1$.

Let $p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1})$ be the multi-target measurement likelihood. Then the set equivalent of (4.18) is

$$p(\mathbf{X}_{k+1} | \mathbf{Z}^{k+1}) = \frac{p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1}) p(\mathbf{X}_{k+1} | \mathbf{Z}^k)}{\int p(\mathbf{Z}_{k+1} | \mathbf{X}_{k+1}) p(\mathbf{X}_{k+1} | \mathbf{Z}^k) \delta \mathbf{X}_{k+1}}. \quad (6.20)$$

The prior at the initial time step $p(\mathbf{X}_0)$, and the prediction and correction equations, (6.19) and (6.20) respectively, is sufficient to describe the time evolution of the distribution of the state \mathbf{X} given measurements \mathbf{Z} ,

$$\begin{aligned} p(\mathbf{X}_0) &\xrightarrow{c} p(\mathbf{X}_0 | \mathbf{Z}^0) \xrightarrow{P} p(\mathbf{X}_1 | \mathbf{Z}^0) \xrightarrow{c} p(\mathbf{X}_1 | \mathbf{Z}^1) \xrightarrow{P} \dots \\ \dots &\xrightarrow{c} p(\mathbf{X}_k | \mathbf{Z}^k) \xrightarrow{P} p(\mathbf{X}_{k+1} | \mathbf{Z}^k) \xrightarrow{c} p(\mathbf{X}_{k+1} | \mathbf{Z}^{k+1}) \xrightarrow{P} \dots \end{aligned} \quad (6.21)$$

From the above we see that there are many conceptual similarities between single target Bayes filtering, presented in Section 4.3, and multi-target Bayes filtering. However, while the single target Bayes filter is computationally tractable, its multi-target generalization is typically not, due to the need to compute set integrals. Because of this approximations are needed, in the next section we will introduce a first order approximation of the full multi-target Bayes filter.

6.2.1 The probability hypothesis density filter

For a random vector $\mathbf{x} \in \mathbb{R}^{n_x}$ with conditional pdf $p_{\mathbf{x}}(\mathbf{x}|\mathbf{z})$, the first order moment, also called the expected value, is defined as

$$\mathbb{E}_{p_{\mathbf{x}}}[\mathbf{x}|\mathbf{z}] \triangleq \int_{\mathbb{R}^{n_x}} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}|\mathbf{z}) d\mathbf{x} \quad (6.22a)$$

$$= \hat{\mathbf{x}}. \quad (6.22b)$$

For a random vector with Gaussian pdf $p_{\mathbf{x}}(\mathbf{x}_k | \mathbf{z}^k) = \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k})$, the expected value is equal to the mean vector, $\hat{\mathbf{x}}_{k|k} = m_{k|k}$. The α - β -filter recursion (4.27) corresponds to propagating the expected value,

$$\dots \xrightarrow{c} \hat{\mathbf{x}}_{k|k} \xrightarrow{P} \hat{\mathbf{x}}_{k+1|k} \xrightarrow{c} \hat{\mathbf{x}}_{k+1|k+1} \xrightarrow{P} \dots \quad (6.23)$$

The first-order moment of a multi-target pdf is a density function defined on single target states $\mathbf{x} \in \mathcal{X}_0$ (Mahler, 2007b),

$$D_{k|k}(\mathbf{x}). \quad (6.24)$$

In point process theory $D_{k|k}(\mathbf{x})$ is called first-moment density or intensity density, see e.g. Mahler (2007b), however for target tracking the name probability hypothesis density (PHD) was adopted by Mahler (2007b). Despite its name, the PHD should not be confused with a probability density function. It is uniquely determined by the property that, given any region S in single target state space \mathcal{X}_0 , i.e. $S \subseteq \mathcal{X}_0$, the integral

$$\int_S D_{k|k}(\mathbf{x}) d\mathbf{x} \quad (6.25)$$

is the expected number of targets in S (Mahler, 2007b). Especially, if $S = \mathcal{X}_0$ is the entire state space then

$$N_{k|k} \triangleq \int D_{k|k}(\mathbf{x}) d\mathbf{x} \quad (6.26)$$

is the expected total number of targets (Mahler, 2007b).

The PHD filter propagates the first order multi-target moment (6.24) through time,

$$\begin{aligned} D_0(\mathbf{x}) &\xrightarrow{c} D_{0|0}(\mathbf{x}) \xrightarrow{p} D_{1|0}(\mathbf{x}) \xrightarrow{c} D_{1|1}(\mathbf{x}) \xrightarrow{p} \dots \\ \dots &\xrightarrow{c} D_{k|k}(\mathbf{x}) \xrightarrow{p} D_{k+1|k}(\mathbf{x}) \xrightarrow{c} D_{k+1|k+1}(\mathbf{x}) \xrightarrow{p} \dots \end{aligned} \quad (6.27)$$

and can be interpreted as an RFS equivalent to the α - β -filter for state vector estimation, see Section 4.4.1. It has been noted that, in principle, it is possible to derive predictor and corrector equations for a second order multi-target filter, however such a filter is unlikely to be computationally tractable (Mahler, 2007b).

PHD filter initialization consists of choosing a prior PHD (Mahler, 2007b)

$$D_0(\mathbf{x}) = n_0 \times s_0(\mathbf{x}), \quad (6.28)$$

where n_0 is the initial expected number of targets, and $s_0(\mathbf{x})$ is a pdf with peaks that correspond to likely initial target locations.

Given a PHD $D_{k|k}(\mathbf{x})$, the predicted PHD is (Mahler, 2007b)

$$\begin{aligned} D_{k+1|k}(\mathbf{x}) = & \underbrace{D_{k+1|k}^b(\mathbf{x})}_{\text{Birth of new targets}} + \underbrace{\int p_S(\mathbf{x}') p_{k+1|k}(\mathbf{x}|\mathbf{x}') D_{k|k}(\mathbf{x}') d\mathbf{x}'}_{\text{Prediction of existing targets}} \\ & + \underbrace{\int p_{k+1|k}^s(\mathbf{x}|\mathbf{x}') D_{k|k}(\mathbf{x}') d\mathbf{x}'}_{\text{Target spawning}} \end{aligned} \quad (6.29)$$

where

- $D_{k+1|k}^b(\mathbf{x})$ is the likelihood that new targets will appear at time $k + 1$,
- $p_S(\mathbf{x}')$ is the probability that a target with state \mathbf{x}' at time k will survive to time $k + 1$,
- $p_{k+1|k}(\mathbf{x}|\mathbf{x}')$ is a single target Markov transition density, and
- $p_{k+1|k}^s(\mathbf{x}|\mathbf{x}')$ is the likelihood that a target with state \mathbf{x}' at time k will spawn a target with state \mathbf{x} at time $k + 1$.

Under the assumption that the multi-target distribution corresponding to the predicted PHD is approximately Poisson, the corrected PHD is (Mahler, 2007b)

$$D_{k+1|k+1}(\mathbf{x}) \approx \underbrace{(1 - p_D(\mathbf{x})) D_{k+1|k}(\mathbf{x})}_{\text{Not detected targets}} + \underbrace{\sum_{\mathbf{z} \in \mathcal{Z}} \frac{p_D(\mathbf{x}) p_{k+1}(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z}) + \int p_D(\mathbf{x}') p_{k+1}(\mathbf{z}|\mathbf{x}') D_{k+1|k}(\mathbf{x}') d\mathbf{x}'} D_{k+1|k}(\mathbf{x})}_{\text{Detected targets}} \quad (6.30)$$

where

- $p_D(\mathbf{x})$ is the probability that a measurement will be collected at time step $k + 1$ from a target with state \mathbf{x} ,
- $p_{k+1}(\mathbf{z}|\mathbf{x})$ is the sensor likelihood function,
- λ is the average of the Poisson distributed number of false alarms collected by the sensor, and
- $c(\mathbf{z})$ is the spatial distribution of the false alarms.

In (6.29) and (6.30) the theoretical equations for PHD prediction and correction are given, however these equations must be implemented to be useful in a practical setting. One alternative is to use particle filters, e.g. a sequential Monte Carlo filter; another alternative is to approximate the PHD with a weighted mixture of Gaussian distributions and use the Kalman filter, or one of its non-linear extensions.

A sequential Monte Carlo implementation of the PHD filter for point targets is presented by Vo et al. (2005), with a convergence analysis given by Vo et al. (2005); Clark and Bell (2006); Johansen et al. (2006). The Gaussian mixture PHD (GM-PHD) filter for point targets assumes that at time step k the PHD has the following Gaussian mixture representation (Vo and Ma, 2006),

$$D_{k|k}(\mathbf{x}) = \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}). \quad (6.31)$$

Thus, because $\int \mathcal{N}(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}) d\mathbf{x} = 1$, the number of present targets is readily given as $N_{k|k} = \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$. To arrive at predictor and corrector equations, the following assumptions were made by Vo and Ma (2006), repeated below for clarity:

Assumption A1. Each target evolves and generates observations independently of one another. _____

Assumption A2. Clutter is Poisson and independent of target-originated measurements. _____

Assumption A3. The predicted multiple-target RFS is Poisson. _____

Assumption A4. Each target follows a linear Gaussian dynamical model and the sensor has a linear Gaussian measurement model. _____

Assumption A5. The survival and detection probabilities are state independent. _____

Assumption A6. The intensities of the birth and spawn RFS:s are Gaussian mixtures. _____

As mentioned by Vo and Ma (2006), Assumption A1, A2, A4 and A5 are standard in many target tracking applications, see e.g. Bar-Shalom and Fortmann (1987). The third assumption is reasonable in applications where target interactions are negligible. Extended, more complete, remarks on the assumptions are given in Vo and Ma (2006).

The full prediction and correction equations are not repeated here, instead we refer the reader to the paper by Vo and Ma (2006). Convergence analysis of the GM-PHD filter is given in Clark and Vo (2007).

6.2.2 The cardinalized probability hypothesis density filter

A known drawback of the PHD filter is that the cardinality is estimated using a single parameter (the mean), resulting in the cardinality distribution being approximated with a Poisson distribution. Because the Poisson mean and variance are equal, when the true cardinality is high the corresponding estimate has a high variance. In practice, this results in an oversensitive cardinality estimate (Erdinc et al., 2005), e.g. seen when there are missed detections. To improve upon this, the cardinalized probability hypothesis density (CPHD) filter was introduced (Mahler, 2007a). In addition to propagating the PHD $D_{k|k}(\mathbf{x})$ in time,

the CPHD filter also propagates the full cardinality distribution $P_{k|k}(n)$,

$$\begin{aligned} \left\{ \begin{array}{l} D_0(\mathbf{x}) \\ P_0(n) \end{array} \right\} &\xrightarrow{c} \left\{ \begin{array}{l} D_{0|0}(\mathbf{x}) \\ P_{0|0}(n) \end{array} \right\} \xrightarrow{p} \left\{ \begin{array}{l} D_{1|0}(\mathbf{x}) \\ P_{1|0}(n) \end{array} \right\} \xrightarrow{c} \left\{ \begin{array}{l} D_{1|1}(\mathbf{x}) \\ P_{1|1}(n) \end{array} \right\} \xrightarrow{p} \dots \\ \dots &\xrightarrow{c} \left\{ \begin{array}{l} D_{k|k}(\mathbf{x}) \\ P_{k|k}(n) \end{array} \right\} \xrightarrow{p} \left\{ \begin{array}{l} D_{k+1|k}(\mathbf{x}) \\ P_{k+1|k}(n) \end{array} \right\} \xrightarrow{c} \left\{ \begin{array}{l} D_{k+1|k+1}(\mathbf{x}) \\ P_{k+1|k+1}(n) \end{array} \right\} \xrightarrow{p} \dots \end{aligned} \quad (6.32)$$

To initialize the CPHD filter an initial PHD

$$D_0(\mathbf{x}) = n_0 \times s_0(\mathbf{x}) \quad (6.33)$$

and an initial cardinality distribution $P_0(n)$ have to be chosen (Mahler, 2007b), where $s_0(\mathbf{x})$ is a pdf with peaks that correspond to likely initial target locations, and $P_0(n)$ is a pmf defined on $n \in \mathbb{N} = \{0, 1, 2, 3, \dots\}$ such that the expected value is n_0 ,

$$n_0 = \sum_{n=0}^{\infty} n P_0(n). \quad (6.34)$$

The CPHD prediction and correction equations are more intricate than their PHD counterparts, and they are not repeated here. However, we note that the CPHD prediction and correction require the assumption that the propagated multi-target distribution is approximated with an i.i.d. cluster process, cf. (6.14). A GM implementation of the CPHD filter for point targets can be found in the paper by Vo et al. (2007).

6.3 A brief revisit to the SLAM problem

Random finite sets were introduced in this chapter in part because they are a suitable remedy to the intricacies of performance evaluation of multiple object estimation, as highlighted in Section 5.3. Similarly to multiple target tracking, in SLAM the map state \mathbf{M} consists of an unknown number of landmarks, each with an unknown state. Thus, modeling the landmarks as an RFS

$$\mathbf{M} = \left\{ \mathbf{m}^{(i)} \right\}_{i=1}^m, \quad (6.35)$$

rather than a vector, as in (4.34), can simplify performance evaluation of the SLAM map. A further benefit of the RFS model is that the measurement to landmark data association can become more robust against high clutter rate and measurement noise.

An RFS formulation for SLAM was first proposed by Mullane et al. (2008), who model the robot state and map as a joint finite set valued variable and give a Gaussian mixture implementation of the proposed PHD-SLAM filter. A Rao-Blackwellized implementation of PHD-SLAM was given by Mullane et al. (2011), with early results presented by Mullane et al. (2010). The implemented PHD-SLAM filter is based on using the GM-PHD filter for the map and a particle filter

for the robot trajectory. A similar approach is taken by Lee et al. (2012).

The RFS approaches to SLAM share most similarities with FAST-SLAM (Montemerlo et al., 2002). Comparisons show that the RFS approach outperforms FAST-SLAM in scenarios with high levels of clutter measurements.

7

Extended target tracking

In this chapter we revisit the target tracking problem, this time with an emphasis on a type of target that is called extended. We give a definition of extended target in Section 7.1 and provide some extended target models in Section 7.2. Measurement set partitioning is presented in Section 7.3, and performance evaluation for extended target estimates is discussed in Section 7.4.

7.1 Introduction

In many modern target tracking applications the point target assumption, see Definition 5.1 in Chapter 5, is not valid. Examples of such applications include vehicle tracking using automotive radars, people tracking using laser range sensors or object tracking using vision sensors, e.g. cameras. See Figure 7.1 for two examples. This prompts us to make the following definition:

Definition 7.1 (Extended target). A target that potentially gives rise to more than one measurement per time step. _____

It is important to note here that the target tracking properties I to V listed in Section 5.1 apply to extended targets too. In addition to those properties, for extended target tracking it also holds that

VI the number of measurements generated by each target is unknown.

The multiple measurements per target raise interesting potentials for the estimation of target shape and size. While the single measurement setting is sufficient to track the targets' centers of mass, multiple measurements allow certain properties of the targets to be estimated too, e.g. shape, size and orientation. With this added knowledge, differentiation between different target types is possible.

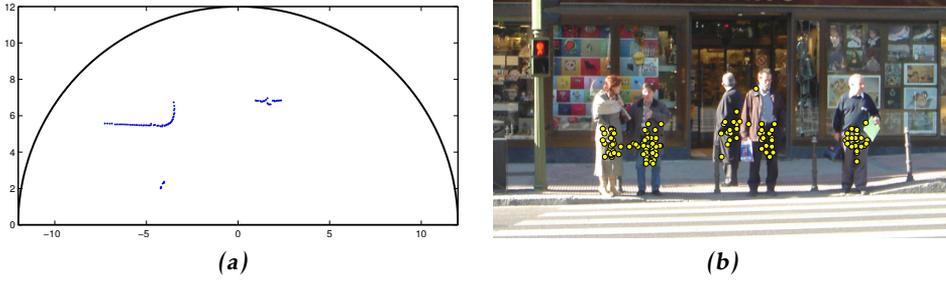


Figure 7.1: (a) Laser data with car, bicycle and pedestrian. (b) Camera image with pedestrian detections. Detections obtained using the classifier by Maji et al. (2008).

7.2 Extended target modeling

Naturally there are more than one possible way to model extended targets. In this section we will briefly overview a few different alternatives.

7.2.1 Extended target measurements

Let ξ_k denote the extended target state at time t_k . Modeling the distribution of the set of target generated measurements means to model the distribution

$$p(\mathbf{Z}_k | \xi_k) = p(\mathbf{Z}_k | N_{z,k}, \xi_k) p(N_{z,k} | \xi_k), \quad (7.1)$$

where the measurement set \mathbf{Z}_k was defined in (5.1). The measurements $\mathbf{z}_k^{(j)}$ are often assumed to be i.i.d.,

$$p(\mathbf{Z}_k | N_{z,k}, \xi_k) = \prod_{j=1}^{N_{z,k}} p(\mathbf{z}_k^{(j)} | \xi_k), \quad (7.2)$$

where $p(\mathbf{z}_k | \xi_k)$ is a likelihood function for a single target generated measurement.

In the extended target model suggested by Gilholm and Salmond (2005) the number $N_{z,k}$ of target generated measurements is Poisson distributed with a rate parameter λ_k . The probability mass function for $N_{z,k}$ is

$$p(N_{z,k} | \lambda_k) = \mathcal{PS}(N_{z,k}; \lambda_k) \quad (7.3a)$$

$$= \frac{e^{-\lambda_k} \lambda_k^{N_{z,k}}}{N_{z,k}!}. \quad (7.3b)$$

Each of the $N_{z,k}$ measurements are then distributed according to a spatial extent model. The analysis in Gilholm and Salmond (2005) is limited to the single target case, and a multiple hypothesis Kalman filter implementation and a particle filter implementation is given.

The model by Gilholm and Salmond (2005) has been extended to modeling the generation of target measurements as an inhomogeneous Poisson point process (Gilholm et al., 2005). The probability of n measurements falling in a region A of the surveillance space is

$$P(N_{z,k} = n) = \mathcal{PS}(n; \lambda_k(A)), \quad (7.4a)$$

where

$$\lambda_k(A) = \int_A \Lambda_k(\mathbf{z} | \xi_k) d\mathbf{z} \quad (7.4b)$$

is the expected number of measurements falling in A , and $\Lambda_k(\mathbf{z} | \xi_k)$ is the spatially dependent intensity of the Poisson process. The likelihood function for a single measurement in the region A is

$$p(\mathbf{z} | \xi_k) = \frac{\Lambda_k(\mathbf{z} | \xi_k)}{\lambda_k(A)}, \quad (7.5)$$

i.e. given ξ_k , any measurement \mathbf{z} is a random draw from this pdf. For this extended target model, Gilholm et al. (2005) gave a particle filter implementation for the multiple target case, and an extended target PHD filter for this model was developed by Mahler (2009).

A slightly different approach is taken by Swain and Clark (2010). They assume that at most one measurement is generated by a point target, however the point targets belong to groups, or clusters. The cluster centers are referred to as parent processes, and their point target processes are daughter processes. The resulting filter presented by Swain and Clark (2010) is similar to the extended target PHD filter (Mahler, 2009).

In all papers in the second part of this thesis, except Paper A and Paper F, the number of extended target generated measurements is modeled as Poisson distributed, with a target state dependent measurement rate $\lambda_k(\xi_k)$. In the next section, different models for the extended target state are given, together with corresponding measurement models.

7.2.2 Extended target state

For the extended target state, a common choice is to let the extended target state be a vector $\xi_k \in \mathbb{R}^{n_x}$ that is Gaussian distributed,

$$p(\xi_k | \mathbf{Z}^k) = \mathcal{N}(\xi_k; m_{k|k}, P_{k|k}). \quad (7.6)$$

The measurement distribution is then typically modeled as

$$p(\mathbf{z}_k^{(j)} | \xi_k) = \mathcal{N}(\mathbf{z}_k^{(j)}; h_k(\xi_k), R_k), \quad (7.7)$$

where $h_k(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ is a non-linear measurement function. In this case, the extended state vector contains all states that are of interest, such as position, velocity, acceleration and the states that govern the shape and size of the extended target. A Gaussian state vector is used in Paper B and Paper C, and is also used by

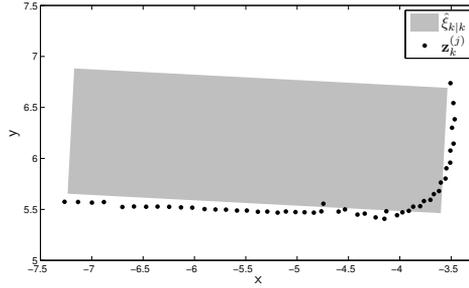


Figure 7.2: State estimate with rectangular shape for the extended target, and measurements from a car.

e.g. Salmond and Parr (2003); Gilholm and Salmond (2005); Baum et al. (2010b); Baum and Hanebeck (2011); Baum et al. (2011).

In Example 7.1 a Gaussian state vector is applied to the car data from Figure 7.1a, under the assumption that the target is shaped as a rectangle.

Example 7.1: Gaussian state

Consider the laser data in Figure 7.1a, which features measurements from a car, a bicycle and a pedestrian. For the car a rectangle appears to be a suitable shape model. Let the extended target state be a Gaussian distributed vector,

$$\xi_k = [x_k, y_k, \ell_k, w_k, \phi_k]^T \quad (7.8)$$

where (x_k, y_k) is the position, (ℓ_k, w_k) is the length and width of the rectangle, and ϕ_k is the orientation of the rectangle.

Using the knowledge that the laser range sensor measures along the edge of the shape, i.e. along the edges of the rectangle, Figure 7.2 shows the result of this extended target state applied to the car measurements from Figure 7.1a. The rectangle is not an exact fit to the measurements, however it is a reasonable approximation of the shape.

Note that for this type of example, modeling the measurement function $h_k(\cdot)$ can be quite complicated. Modeling of the measurement function for laser range data is considered in Paper C.

An alternative to the Gaussian model (7.6) was proposed by Koch (2008). This model is sometimes referred to as the random matrix model, or random matrix framework. The random matrix model defines the extended target state as the combination of a kinematical state vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$, and an extension state matrix $X_k \in \mathbb{S}_{++}^d$ representing the size of the target. Modeling the extension as a random matrix means that the shape is modeled as elliptical.

The extended target state $\xi_k = (\mathbf{x}_k, X_k)$ is modeled as Gaussian inverse Wishart (GIW) distributed (Koch, 2008),

$$p(\xi_k | \mathbf{Z}^k) = p(\mathbf{x}_k | X_k, \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (7.9a)$$

$$= \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k} \otimes X_k) \mathcal{IW}(X_k; v_{k|k}, V_{k|k}), \quad (7.9b)$$

and the measurement distribution can be modeled as

$$p(\mathbf{z}_k^{(j)} | \xi_k) = \mathcal{N}(\mathbf{z}_k^{(j)}; h_k(\mathbf{x}_k), X_k). \quad (7.10)$$

Koch (2008) uses a linear function $h_k(\mathbf{x}_k) = H_k \mathbf{x}_k$, which results in linear correction for the extended target state estimate. The GIW model is used in Paper D, and is also used by e.g. Koch and Feldmann (2009); Wieneke and Koch (2010); Lan and Rong Li (2012).

In this model, the measurement covariance is given by the extension matrix, and the measurements are assumed to be spread across the target surface. The kinematic state vector contains states that are derivatives of the spatial state component, denoted \mathbf{r}_k (Koch, 2008). For example, if the measurements are of the extended target's Cartesian (x_k, y_k) -position, then the spatial state component is $\mathbf{r}_k = [x_k, y_k]^T$. In this case the kinematic state vector \mathbf{x}_k contains the position \mathbf{r}_k , and possibly also higher derivatives of the spatial components such as velocity $d\mathbf{r}_k/dt_k$ and acceleration $d^2\mathbf{r}_k/dt_k^2$.

As mentioned above, in the random matrix model the extended target shape is assumed to be an ellipse (Koch, 2008). While this assumption is limiting, in many scenarios the ellipse is a sufficient approximation of the true extended target shape. Example 7.2 shows the GIW model applied to the laser data in Figure 7.1a.

Example 7.2: GIW state

Consider again the laser data in Figure 7.1a. Let the extended target be GIW distributed, with kinematical state

$$\mathbf{x}_k = [x_k, y_k]^T, \quad (7.11)$$

and extension state $X_k \in \mathbb{S}_{++}^2$. Using a linear measurement function $H_k \mathbf{x}_k = \mathbf{x}_k$, in Figure 7.3 this extended target state is applied to the car, bicycle and pedestrian measurements from Figure 7.1a, respectively.

For the bicycle and the pedestrian an ellipse appears to be a reasonable model for the shape, however it is a poor model for the car.

The measurement model (7.10) implicitly assumes that the sensor noise is negligible in comparison to the target extent (Feldmann and Fränken, 2008; Feldmann et al., 2011), a relation that does not hold in all scenarios.

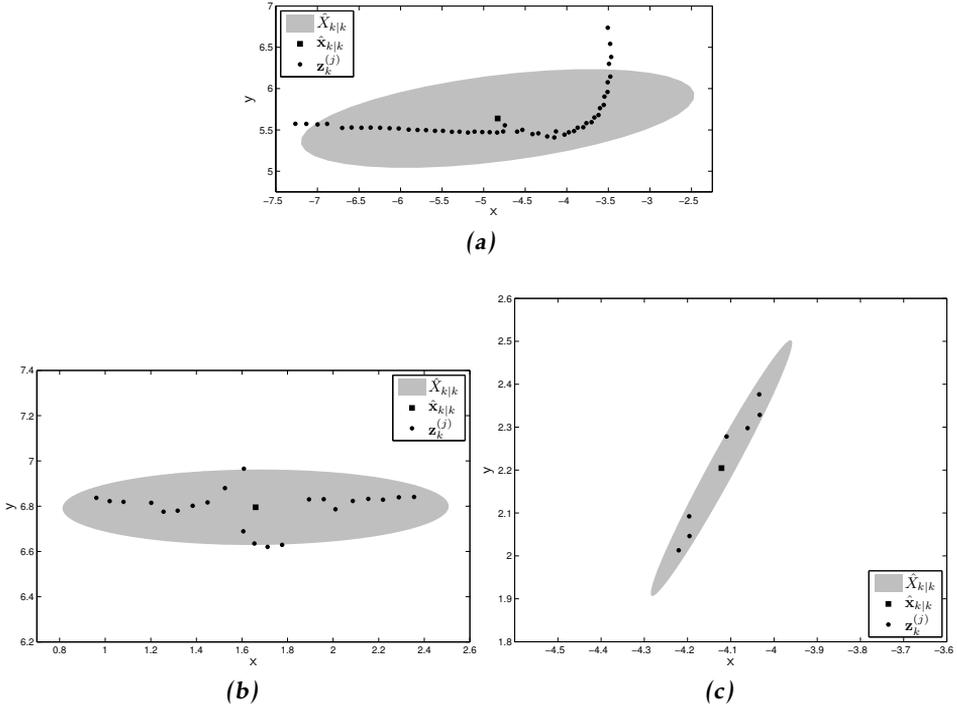


Figure 7.3: The GIW extended target state model applied to the laser data from Figure 7.1a. (a), (b) and (c) Model applied to measurements from a car, a bicycle and a pedestrian, respectively.

Feldmann and Fränken (2008), see also the work by Feldmann et al. (2011), proposed to approximate the kinematical and extension state as independent

$$p(\xi_k | \mathbf{Z}^k) = p(\mathbf{x}_k | X_k, \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (7.12a)$$

$$\approx p(\mathbf{x}_k | \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (7.12b)$$

$$= \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \mathcal{IW}(X_k; v_{k|k}, V_{k|k}), \quad (7.12c)$$

with measurement distribution model

$$p(\mathbf{z}_k^{(j)} | \xi_k) = \mathcal{N}(\mathbf{z}_k^{(j)}; H_k \mathbf{x}_k, z X_k + R_k), \quad (7.13)$$

where z is a scaling factor, and $R_k \in \mathbb{S}_+^d$. This measurement model can be interpreted as meaning that X_k is the true extension of the extended target, while R_k is the sensor error covariance matrix. Note that with this measurement distribution, the correction for the extended target state estimate is no longer linear, but has to be approximated (Feldmann and Fränken, 2008; Feldmann et al., 2011). For a comparison of the model (7.9), (7.10) and the model (7.12), (7.13), see the paper by Feldmann et al. (2011).

In comparison to (7.9), in the GIW model (7.12) the kinematic state vector can be defined independently of the spatial state component. This includes the possibility of states that represent, e.g., the heading and the turn-rate. This variant of the GIW model is used in Paper F and Paper G.

A third model for the measurement distribution is suggested by Lan and Rong Li (2012),

$$p\left(\mathbf{z}_k^{(j)} \mid \xi_k\right) = \mathcal{N}\left(\mathbf{z}_k^{(j)}; H_k \mathbf{x}_k, B_k X_k B_k^T\right), \quad (7.14)$$

where B_k is a $d \times d$ non-singular transformation matrix. The transformation that B_k represents could, e.g., be a rotation or a scaling of the extension matrix X_k . Setting the two models (7.13) and (7.14) equal, we have the following relation (Lan and Rong Li, 2012),

$$B_k X_k B_k^T = z X_k + R_k \quad (7.15a)$$

$$= (z X_k + R_k)^{\frac{1}{2}} X_k^{-\frac{1}{2}} X_k X_k^{-\frac{1}{2}} (z X_k + R_k)^{\frac{T}{2}}, \quad (7.15b)$$

$$\Leftrightarrow B_k = (z X_k + R_k)^{\frac{1}{2}} X_k^{-\frac{1}{2}}. \quad (7.15c)$$

Under the assumption $X_k \approx \hat{X}_{k|k-1} = \mathbb{E}\left[X_k \mid \mathbf{Z}^{k-1}\right]$, Lan and Rong Li (2012) approximate (7.15c) as

$$B_k \approx \left(z \hat{X}_{k|k-1} + R_k\right)^{\frac{1}{2}} \hat{X}_{k|k-1}^{-\frac{1}{2}}, \quad (7.16)$$

which gives a matrix B_k that is not a function of the extended target state.

Both the state vector representation (7.6) and the random matrix representation (7.9), (7.12) can be augmented with a state variable $\gamma_k > 0$, where γ_k is related to measurement generating Poisson rate as follows,

$$\lambda_k(\xi_k) = \gamma_k. \quad (7.17)$$

The conjugate prior for the Poisson rate is the gamma distribution, see Table 4.1 in Chapter 4. In this case it is natural to model $\xi_k = (\gamma_k, \mathbf{x}_k)$ as gamma Gaussian (GG) distributed,

$$p\left(\xi_k \mid \mathbf{Z}^k\right) = p\left(\gamma_k \mid \mathbf{Z}^k\right) p\left(\mathbf{x}_k \mid \mathbf{Z}^k\right) \quad (7.18a)$$

$$= \mathcal{GAM}\left(\gamma_k; \alpha_{k|k}, \beta_{k|k}\right) \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, P_{k|k}\right), \quad (7.18b)$$

and to model $\xi_k = (\gamma_k, \mathbf{x}_k, X_k)$ as gamma Gaussian inverse Wishart (GGIW) distributed,

$$p\left(\xi_k \mid \mathbf{Z}^k\right) = p\left(\gamma_k \mid \mathbf{Z}^k\right) p\left(\mathbf{x}_k \mid X_k, \mathbf{Z}^k\right) p\left(X_k \mid \mathbf{Z}^k\right) \quad (7.19a)$$

$$= \mathcal{GAM}\left(\gamma_k; \alpha_{k|k}, \beta_{k|k}\right) \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, P_{k|k} \otimes X_k\right) \mathcal{IW}\left(X_k; v_{k|k}, V_{k|k}\right), \quad (7.19b)$$

or

$$p(\xi_k | \mathbf{Z}^k) = p(\gamma_k | \mathbf{Z}^k) p(\mathbf{x}_k | \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (7.20a)$$

$$= \mathcal{GAM}(\gamma_k; \alpha_{k|k}, \beta_{k|k}) \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \mathcal{IW}(X_k; v_{k|k}, V_{k|k}). \quad (7.20b)$$

A Bayesian recursion for estimation of the Poisson rate γ_k is given in Paper E. The GGIW model (7.19) is used in Paper H.

7.2.3 Extension shape models

Closely related to modeling the distribution of the extended target measurements is modeling the extended target shape. Dezert (1998) models the extension as a collection of points, which together with the target kinematics are jointly estimated from the measurements. In this case the target shape is given by the shape of the points. Gilholm and Salmond (2005) give an example in which the target extension is an infinitely thin stick of length ℓ , a similar example is used by Boers et al. (2006).

The GIW measurement models suggested by Koch (2008), Feldmann et al. (2011), and Lan and Rong Li (2012), assume that the extended target is shaped like an ellipse, and that the measurements are spread across the target surface. Inspired by Koch (2008), Degerman et al. (2011) decompose the extension matrix into principal components, and design a heuristic Kalman filter for tracking the extension. Zhu et al. (2011) model the extension as a combination of two ellipses, more specifically two Gaussian distributions. The random hypersurface model by Baum and Hanebeck (2009) models the measurements as random samples of measurement generating points on the target surface. A random hypersurface model is given for elliptic targets by Baum et al. (2010b), and a comparison between the elliptic random hypersurface model and the GIW model is given by Baum et al. (2010a).

Modeling the measurements as being spread across the target surface is appropriate, e.g., when airborne radars are used to track ground vehicles. For other sensors, e.g. the laser range sensor, the measurements are better modeled as being spread along the edge of the target surface. Rectangular and elliptical shape models are given for the laser range sensor in Paper C, Lundquist et al. (2011a) give a more general shape model for laser range type sensors, capable of estimating arbitrary shapes. Note that the methods presented by Lundquist et al. (2011a) and in Paper C are capable of estimating the entire extension, even when only parts of the extension are seen. For similar scenarios, Petrov et al. (2011) give a sampling based measurement model for extended targets whose extensions are measured across so called regions of interest. They illustrate their approach by considering circular objects. Lundquist et al. (2011b) model the extended targets using polynomials, an approach that is shown to be applicable to road-mapping using vehicle radars.

7.3 Measurement set partitioning

In Chapter 5 the data association problem was mentioned, and some data association methods for multiple point target tracking in clutter were given in Section 5.2. The data association problem must be solved also for multiple extended target tracking in clutter. Because more than one measurement may originate from the same target, one approach to solving the data association problem is to divide the set of measurements into non-empty subsets, where each subset is assumed to contain measurement that are all from the same source. The subsets can then be associated to the extended targets, similarly to how single measurements are associated to the point targets.

In this thesis, we refer to the division of the measurement set into non-empty subsets as partitioning the measurement set, a particular partitioning of the measurement set is called a partition, and the non-empty subsets are denoted cells. Note that in the literature, partitioning is sometimes called clustering or cluster analysis, and the cells are then typically called clusters. For a given partition, the cells can be interpreted as containing measurements that all stem from either a single extended target or a clutter source.

There are different ways to approach the measurement partitioning, e.g. the measurement-to-cell association could be either hard or soft. Hard association means that the measurement belongs to the cell or not, similarly to NN and GNN data association. Soft association means that the measurement belongs to the cell to a certain degree. This thesis will only consider hard measurement-to-cell associations.

When it comes to computing the partitions, one method is to assume that the partition should have K cells, and then assign the measurements to the K cells by minimizing a cost function. One such partitioning method is K -means clustering, see e.g. the textbooks by Bishop (2006); Hastie et al. (2009). K -means clustering is a type of combinatorial algorithm (Hastie et al., 2009), meaning that it works directly on the data and does not have an underlying probability model. Given a desired number of cells, the algorithm assigns the data to the cells by iteratively minimizing a cost function. One of K -means clustering's drawbacks is its tendency to get stuck in local optimas during the cost function minimization. An improved version, called K -means++ clustering, is reported to have an initialization that better avoids local optimas (Arthur and Vassilvitskii, 2007; Ostrovsky et al., 2006).

Alternatively, one may define a criterion by which it is determined whether, or not, two measurements belong to the same cell. One such criterion is the distance between the measurements, as given by a distance measure, e.g. the Euclidean metric or Mahalanobis distance. This is also known as hierarchical clustering (Hastie et al., 2009), and there are two types of hierarchical methods: agglomerative and divisive. Agglomerative is bottom-up, i.e. it starts with all measurements in one cell each, and builds larger cells (Hastie et al., 2009). Divisive is top-down, i.e. it starts with all measurements in the same cell and splits into smaller cells

(Hastie et al., 2009).

In terms of the distance used to determine cell membership, two alternatives are complete linkage and single linkage. Let W_1 and W_2 be two cells whose union is empty, i.e. the two cells do not have any measurements in common. For complete linkage, the distance between the cells is measured as

$$\max \left\{ d \left(\mathbf{z}_k^{(i)}, \mathbf{z}_k^{(j)} \right) : \mathbf{z}_k^{(i)} \in W_1, \mathbf{z}_k^{(j)} \in W_2 \right\}, \quad (7.21)$$

where $d(\cdot, \cdot)$ is the distance measure used. This means that the distance between the two cells is the maximum distance between a pair of measurements. For single linkage, the distance between the cells is measured as

$$\min \left\{ d \left(\mathbf{z}_k^{(i)}, \mathbf{z}_k^{(j)} \right) : \mathbf{z}_k^{(i)} \in W_1, \mathbf{z}_k^{(j)} \in W_2 \right\}. \quad (7.22)$$

In this case the distance between the two cells is the minimum distance between a pair of measurements.

In Paper B a type of agglomerative, single linkage, hierarchical partitioning method, called Distance Partitioning, is proposed for use in multiple extended target tracking. Distance Partitioning forms the basis for the measurement set partitioning that is used in this thesis, however additional methods are also proposed in Paper B and Paper D. Example 7.3 gives a small comparison of Distance Partitioning and K -means++ clustering.

Example 7.3: Measurement set partitioning

True target measurements were generated from three Gaussian distributions with the following means and covariances,

$$m^{(1)} = [0 \ 0]^T, \quad P^{(1)} = \text{diag}([1 \ 1]), \quad (7.23a)$$

$$m^{(2)} = [15 \ 0]^T, \quad P^{(2)} = \text{diag}([0.25 \ 1]), \quad (7.23b)$$

$$m^{(3)} = [0 \ 15]^T, \quad P^{(3)} = \text{diag}([1 \ 0.25]). \quad (7.23c)$$

In total, 10 measurements were sampled from each distribution, let \mathbf{Z}^T denote the set of 30 target measurements. Further, 10 clutter measurements were generated by uniform sampling in $[-5, 20] \times [-5, 20]$. Let \mathbf{Z}^C denote the set of 10 clutter measurements, and let \mathbf{Z}^{TC} denote the union of \mathbf{Z}^T and \mathbf{Z}^C .

The measurements are shown in Figure 7.4a. Figure 7.4b and Figure 7.4c show \mathbf{Z}^T and \mathbf{Z}^{TC} , respectively, after partitioning with Distance Partitioning with threshold 2, measurements with the same color belong to the same cell. We see that Distance Partitioning gives correct cells for the target generated measurements, and places the clutter measurements in individual cells in Figure 7.4c. Both these partitions are quite intuitive, and correspond well to our desire to have a partition in which the cells contain measurements that all stem from either a single extended target or a clutter source.

In Figure 7.4d \mathbf{Z}^T is shown after partitioning with K -means++ clustering, we see that the result is the same partition as when Distance Partitioning is used, see

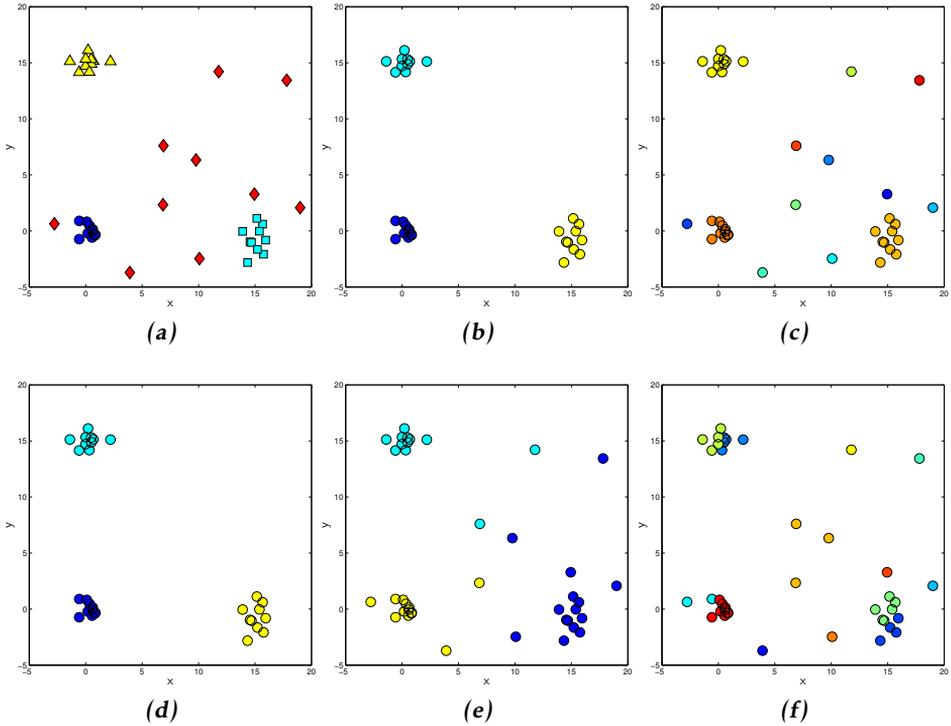


Figure 7.4: Comparison of clustering methods. Measurements with the same color belong to the same cell. (a) True target measurements shown as blue circles, cyan squares and yellow triangles, respectively, and clutter measurements shown as red diamonds. (b) Partition of \mathbf{Z}^T computed using Distance Partitioning, threshold 2. (c) Partition of \mathbf{Z}^{TC} computed using Distance Partitioning, threshold 2. (d) Partition of \mathbf{Z}^T computed using K-means++ clustering, $K = 3$. (e) Partition of \mathbf{Z}^{TC} computed using K-means++ clustering, $K = 3$. (f) Partition of \mathbf{Z}^{TC} computed using K-means++ clustering, $K = 13$.

Figure 7.4b. However, when K-means++ clustering is used for partitioning of \mathbf{Z}^{TC} , the results are much less intuitive. The partitions, for $K = 3$ and $K = 13$ are shown in Figure 7.4e and Figure 7.4f. For this measurement set, K-means++ does not return the correct partition for any value of K between 3 and 13.

7.4 Performance evaluation

In terms of multiple extended target tracking, there is no conceptual difference to multiple point target tracking that prevents the use of the OSPA metric (see Section 5.3). However, the OSPA requires a metric $d(\cdot, \cdot)$ for comparison of a true

extended target state and a state estimate. In this section we will discuss some alternatives for performance evaluation of a single extended target estimate.

In Section 4.5 the RMSE and NEES were introduced as performance metrics for state vector estimates, these metrics are no less valid for an extended target state vector. However, depending on the particular modeling framework used, the extension states could benefit from being treated differently.

For extended targets whose extension parameters are included in the state vector, cf. (7.6), the RMSE and NEES can naturally be used also for the states related to the extension. In case the extended target state is decomposed as in (7.9) or (7.12), the extension estimation error can be evaluated using a matrix norm, e.g. the Frobenius norm,

$$\|X_k - \hat{X}_{k|k}\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^d |X_k^{[ij]} - \hat{X}_{k|k}^{[ij]}|^2}, \quad (7.24)$$

where $d \times d$ is the dimension of the extension matrix, and the notation $A^{[ij]}$ is used to denote the i, j th element of the matrix A . The Frobenius norm for matrices is analogous to the Euclidean norm for vectors. It is used for performance evaluation in Paper G and Paper H.

Another performance metric, often used in computer vision, is a difference measure called Intersection-Over-Union (IOU). The IOU measures the volumes of the intersection and the union of the true extended target and the estimate, and then takes the ratio of the two volumes. Note that the IOU is not a metric, e.g. it does not satisfy the triangle inequality. Further, in comparison to the RMSE, NEES and Frobenius norm, who are all equal to zero when there is no error, the IOU is equal to one when there is no error. The IOU measure is used for performance evaluation in Paper C.

In case a measurement rate γ_k is estimated, a suitable performance metric is the absolute difference,

$$|\gamma_k - \hat{\gamma}_{k|k}|. \quad (7.25)$$

Example 7.4 compares performance metrics for GIW distributed extended targets.

Example 7.4: Extension estimation performance evaluation

Let the extended target state be GIW distributed, with true state

$$\xi_k = (\mathbf{x}_k, X_k), \quad (7.26a)$$

$$\mathbf{x}_k = [0, 0]^T, \quad (7.26b)$$

$$X_k = R \left(\frac{45}{180} \pi \right) \text{diag} \left([5^2, 2^2] \right) R^T \left(\frac{45}{180} \pi \right), \quad (7.26c)$$

where $R(\cdot)$ is a 2D rotation matrix. There are two state estimates,

$$\hat{\xi}_{k|k}^{(1)} = \left(\hat{\mathbf{x}}_{k|k}^{(1)}, \hat{X}_{k|k}^{(1)} \right), \quad (7.27a)$$

$$\hat{\mathbf{x}}_{k|k}^{(1)} = [-0.25, 0.15]^T, \quad (7.27b)$$

$$\hat{X}_{k|k}^{(1)} = R\left(\frac{35}{180}\pi\right) \text{diag}\left([6^2, 1.75^2]\right) R^T\left(\frac{35}{180}\pi\right), \quad (7.27c)$$

and

$$\hat{\xi}_{k|k}^{(2)} = \left(\hat{\mathbf{x}}_{k|k}^{(2)}, \hat{X}_{k|k}^{(2)} \right), \quad (7.28a)$$

$$\hat{\mathbf{x}}_{k|k}^{(2)} = [-1, 1]^T, \quad (7.28b)$$

$$\hat{X}_{k|k}^{(2)} = R\left(\frac{45}{180}\pi\right) \text{diag}\left([5.1^2, 2.1^2]\right) R^T\left(\frac{45}{180}\pi\right). \quad (7.28c)$$

The true state and the estimates are shown in Figure 7.5a, the intersections and the unions of the 2σ -ellipses are shown in Figure 7.5b and Figure 7.5c. The Euclidean norms for the kinematical state vector differences, the Frobenius norms for the extension matrix differences, and the IOU values are

$$\left\| \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}^{(1)} \right\|_2 = 0.29, \quad \left\| \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}^{(2)} \right\|_2 = 1.41, \quad (7.29a)$$

$$\left\| X_k - \hat{X}_{k|k}^{(1)} \right\|_F = 12.79, \quad \left\| X_k - \hat{X}_{k|k}^{(2)} \right\|_F = 1.09, \quad (7.29b)$$

$$\text{IOU} = 0.70, \quad \text{IOU} = 0.64. \quad (7.29c)$$

Determined by the IOU measure, $\hat{\xi}_{k|k}^{(1)}$ is the better estimate. To be able to determine by the Euclidean and Frobenius norms, the two norms would have to be weighed together. Depending on how this is performed, either estimate could be the better one.

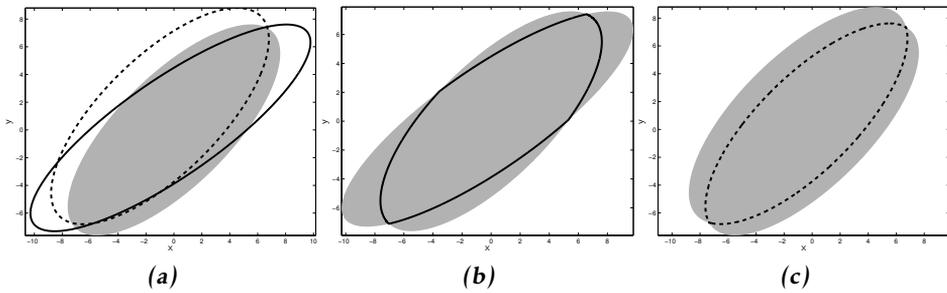


Figure 7.5: Illustration of the difference measure intersection over union. The 2σ -ellipses are plotted. (a) True extended target state ξ_k (gray area), state estimate $\hat{\xi}_{k|k}^{(1)}$ (black solid line), and state estimate $\hat{\xi}_{k|k}^{(2)}$ (black dashed line). (b) The intersection (black solid line) and the union (gray area) of ξ_k and $\hat{\xi}_{k|k}^{(1)}$. (c) The intersection (black dashed line) and the union (gray area) of ξ_k and $\hat{\xi}_{k|k}^{(2)}$.

8

Concluding remarks

This chapter summarizes the thesis, with conclusions in Section 8.1 and recommendations for future work in Section 8.2.

8.1 Conclusions

A method for loop closure detection in SLAM using data from laser sensors was presented. A compact and efficient feature description of each point cloud is given, and AdaBoost is used to construct a classifier that uses the features to classify a point cloud pair as being either from the same location, or not. The classifier is able to detect loop closure from arbitrary direction, and in experiments it is shown to produce detection rates that compare well to related work at low false alarm rates. The real world SLAM experiments showed that the classifier can be used within the context for which it was constructed.

Two different implementations of the extended target probability hypothesis density filter was presented, one Gaussian mixture implementation and one Gaussian inverse Wishart implementation. The Gaussian mixture implementation was also extended to non-linear motion and measurement models. The ideal filter requires consideration of the full set of partitions, which is computationally unfeasible in all but the very simplest cases. It was shown that the full set of partitions can be approximated with a subset of partitions, without having to sacrifice tracking performance. Four different partitioning methods were suggested, and it was shown that they reduce the number of partitions considered by several orders of magnitude, and also that they outperform the well known partitioning method K -means clustering.

It was shown in simulations and experiments that the case of spatially close targets can be difficult to handle. Suitable remedies to improve performance under these circumstances were suggested, and results showed that performance was improved. Further, it was shown that the filter is sensitive to the number of measurements generated by each extended target. A framework for estimating the number of measurements generated was presented, and shown to be capable of estimating an individual measurement rate for each target. The complexity of the implementations increases exponentially with time, and approximations are necessary to ensure computational feasibility. To this end, merging of distribution mixtures was presented. Finally, extended target prediction, spawning and combination was also addressed.

8.2 Future work

A noted drawback of the presented loop detection classifier is that it, compared to related work, is more sensitive to translation, i.e. required a higher degree of point cloud overlap. While this is not problematic in environments with well defined pathways, such as road networks or office hallways, it would present a challenge in environments with less restrictions on motion. A topic for future work is to increase the classifiers ability to handle translation.

The underlying ideas behind the presented loop detection classifier could have extensions to the environment labeling problem. Environment labeling is a problem in which the parts of the sensor data is labeled according to which class it originated from, e.g. ground, vegetation, building walls, cars, humans, etc. Features similar to the ones used to describe the point clouds as a whole could be used as local descriptors of each point in the point clouds. Using a multi-class classifier, the data point could then be labeled with the most likely class label.

Two different models for estimation of the size and shape of extended targets were used, however they were both limited to simpler shapes. A comparison of different models for the shape and size of the extended targets would be interesting. Given a partition of the current measurement set, and prior extended target estimates, the cell to target association problem is similar to classic point target tracking. It would be interesting to see how the partitioning methods could be used together with classic target tracking approaches, like Multiple Hypothesis Tracking, to construct multiple extended target tracking algorithms. If such a tracking algorithm can be devised, a comparison to the extended target PHD filter could show the advantages and disadvantages of using random set theory.

Further work on measurement set partitioning can be undertaken, to ensure that the filtering framework is capable of handling multiple extended targets that maneuver close to each other in heavily cluttered measurement data. When targets are spatially close, the suggested partitioning methods sometimes fail in producing correct partitions. A method capable of detecting that a cell contains measurement that actually belong to multiple sources could improve performance.

Regarding estimation of the measurement rates, better models for the relation between the measurement rate and the target kinematics and target extension is needed. Reduction of distribution mixtures is addressed using merging, and a simple way to construct a pairwise criterion is suggested. The related literature on Gaussian mixture reduction contains different approaches to the problem of finding which components in the mixture should be merged, and which should not be merged. A comparison of these approaches applied to other distribution mixtures would be interesting.

The presented approach to prediction of an extended target modeled with random matrices can be tested further in multiple target scenarios. Only one step prediction is considered in the paper, however the case of multi-step prediction could further show the merits of the prediction. It would also be interesting to include the prediction in an interacting multiple model framework. The spawning and combination of extended targets only handles the two target case, an extension to the case of an arbitrary number of targets would be interesting.

Somewhere along the intersection of mapping and target tracking lies the problem of separating the sensor data into segments that correspond to either stationary or moving objects. This thesis has not handled the segmentation problem, although it is shown that the loop detection classifier is not sensitive to moving objects. An extended target tracking framework could possibly be used to determine which parts of the surrounding environment are stationary, and which parts are moving.

Bibliography

- D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA, January 2007.
- T. Bailey and H. F. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, September 2006.
- T. Bailey, J. I. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 424–429, May 2006.
- Y. Bar-Shalom. *Multitarget-multisensor tracking: applications and advances*, volume II of *Multitarget-multisensor Tracking: Applications and Advances*. Artech House, 1992.
- Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- Y. Bar-Shalom and X. Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.
- Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
- Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and data fusion, a handbook of algorithms*. YBS, 2011.
- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- M. Baum and U. D. Hanebeck. Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs. In *Proceedings of the International Confer-*

- ence on *Information Fusion (FUSION)*, pages 338–345, Chicago, IL, USA, July 2011.
- M. Baum, M. Feldmann, D. Fränken, U. D. Hanebeck, and J. W. Koch. Extended object and group tracking: A comparison of random matrices and random hypersurface models. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Leipzig, Germany, October 2010a.
- M. Baum, B. Noack, and U. D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010b.
- M. Baum, B. Noack, and U. D. Hanebeck. Random hypersurface mixture models for tracking multiple extended objects. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Orlando, Florida, USA, December 2011.
- P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- P. Biber and W. Strasser. The normal distribution transform: A new approach to laser scan matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2743–2748, Las Vegas, USA, October 2003.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA, 1999.
- Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. A track before detect algorithm for tracking extended targets. *IEE Proceedings Radar, Sonar and Navigation*, 153(4):345–351, August 2006.
- M. C. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotics Research*, 27(6):667–691, June 2008.
- J. Callmer, K. Granström, J. I. Nieto, and F. T. Ramos. Tree of Words for Visual Loop Closure Detection in Urban SLAM. In *Proceedings of the Australian Conference on Robotics & Automation (ACRA)*, Canberra, Australia, December 2008.
- M. Campbell, E. Garcia, D. Huttenlocher, I. Miller, P. Moran, A. Nathan, B. Schimpf, N. Zych, J. Catlin, F. Chelarescu, H. Fujishima, F.-R. Kline, S. Lushin, M. Reitmann, A. Shapiro, and J. Wong. Team cornell: Technical review of the darpa urban challenge vehicle, 2007. URL http://www.darpa.mil/grandchallenge/TechPapers/Team_Cornell.pdf.

- Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.
- D. E. Clark and J. Bell. Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing*, 54(7):2652–2661, July 2006.
- D. E. Clark and B.-N. Vo. Convergence analysis of the Gaussian mixture PHD filter. *IEEE Transactions on Signal Processing*, 55(4):1204–1212, April 2007.
- J. Degerman, J. Wintenby, and D. Svensson. Extended target tracking using principal components. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Chicago, IL, USA, July 2011.
- J. C. Dezert. Tracking maneuvering and bending extended target in cluttered environment. In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, pages 283–294, Orlando, FL, USA, September 1998.
- M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.
- H. F. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006.
- O. Erdinc, P. Willett, and Y. Bar-Shalom. Probability hypothesis density filter for multitarget multisensor tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 146–153, Philadelphia, CA, USA, July 2005.
- R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, December 2006.
- M. Feldmann and D. Fränken. Tracking of Extended Objects and Group Targets using Random Matrices - A New Approach. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cologne, Germany, July 2008.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- Y. Freund and R. E. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of European Conference on Computational Learning Theory (EuroCOLT)*, pages 23–37, Barcelona, Spain, March 1995.
- J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

- B. Gates. A Robot in Every Home. *Scientific American*, 296(1):58–65, January 2007.
- A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2004.
- K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings Radar, Sonar and Navigation*, 152(5):364–371, October 2005.
- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2):107–113, April 1993.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.
- K. Granström and U. Orguner. A New Prediction Update for Extended Target Tracking with Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 2012b.
- K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012c.
- K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012d.
- K. Granström and U. Orguner. On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices. *IEEE Transactions on Signal Processing*, 2012e.
- K. Granström and T. B. Schön. Learning to Close the Loop from 3D Point Clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, Taipei, Taiwan, October 2010.
- K. Granström, J. Callmer, F. T. Ramos, and J. I. Nieto. Learning to Detect Loop Closure from Range Data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, Kobe, Japan, May 2009.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.

- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *The International Journal of Robotics Research*, 30(14):1728–1754, December 2011.
- K. Granström, C. Lundquist, F. Gustafsson, and U. Orguner. On extended target tracking using PHD filters. In *Workshop on Stochastic Geometry in SLAM at IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, Minnesota, USA, May 2012.
- M. Guerriero, L. Svensson, D. Svensson, and P. Willett. Shooting two birds with two bullets: how to find Minimum Mean OSPA estimates. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2010.
- T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning : Data mining, inference, and prediction*. Springer, New York, 2 edition, 2009.
- A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- A.M. Johansen, S.S. Singh, A. Doucet, and B.-N. Vo. Convergence of the smc implementation of the phd filter. *Methodology and Computing in Applied Probability*, 8(2):265–291, 2006.
- R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, March 1960.
- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- J. W. Koch and M. Feldmann. Cluster tracking under kinematical constraints using random matrices. *Robotics and Autonomous Systems*, 57(3):296–309, March 2009.
- J. Lan and X. Rong Li. Tracking of extended object or target group using random matrix – part I: New model and approach. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2177–2184, Singapore, July 2012.

- C. S. Lee, D.E. Clark, and J. Salvi. Slam with single cluster phd filters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2096–2101, May 2012.
- L. Ljung. *System Identification Theory for the User*. Prentice Hall, 2 edition, 1999.
- F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, October 1997.
- C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011a.
- C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011b.
- C. Lundquist, K. Granström, and U. Orguner. An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation. *Journal of Selected Topics in Signal Processing*, 2012a.
- C. Lundquist, M. A. Skoglund, K. Granström, and T. Glad. Insights from implementing a system for peer review. *IEEE Transactions on Education*, 2012b. doi: 10.1109/TE.2012.2211876.
- M. Magnusson, T. Duckett, and A. J. Lilienthal. Scan registration for autonomous mining vehicles using 3D-NDT. *jfr*, 24(10):803–827, October 2007.
- R. P. S. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007a.
- R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007b.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA, June 2008.
- J. Markoff. Virtual and artificial, but 58,000 want course, August 2011. URL <http://www.nytimes.com/2011/08/16/science/16stanford.html>.
- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, September 2002.

- J. Mullane, B.-N. Vo, M.D. Adams, and W.S. Wijesoma. A random set formulation for Bayesian SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1043–1049, September 2008.
- J. Mullane, B.-N. Vo, and M.D. Adams. Rao-Blackwellised PHD SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5410–5416, May 2010.
- J. Mullane, B.-N. Vo, M.D. Adams, and B.-T. Vo. A random-finite-set approach to bayesian slam. *IEEE Transactions on Robotics*, 27(2):268–282, April 2011.
- A. Nüchter. *3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*. Springer Publishing Company, Incorporated, 2009.
- U. Orguner, C. Lundquist, and K. Granström. Extended Target Tracking with a Cardinalized Probability Hypothesis Density Filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 65–72, Chicago, IL, USA, July 2011.
- R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 165–174, Berkeley, CA, USA, October 2006.
- N. Petrov, L. Mihaylova, A. Gning, and D. Angelova. A sequential Monte Carlo approach for extended object tracking in the presence of clutter. In *Proceedings of the Informatik 2011: 6th Workshop on Sensor Data Fusion: Trends, Solutions, Applications*, 2011.
- F. T. Ramos, D. Fox, and H. F. Durrant-Whyte. CRF-matching: Conditional random fields for feature-based scan matching. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, USA, June 2007.
- X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking: Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364, October 2003.
- D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):389–395, December 2003.
- T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.
- D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, August 2008.
- D. Simon. *Optimal State Estimation: Kalman, H-infinity and Nonlinear Approaches*. John Wiley & Sons, 2006.

- R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, 1990.
- A. Swain and D. Clark. Extended object filtering using spatial independent cluster processes. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- S. Thrun and J. J. Leonard. Simultaneous Localization and Mapping. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 871–889. Springer, 2008.
- S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- B.-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multi-target filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, October 2005.
- B.-T. Vo, B.-N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, July 2007.
- W. Wieneke and J. W. Koch. Probabilistic tracking of multiple extended targets using random matrices. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, Orlando, FL, USA, April 2010.
- Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.
- H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 73–78, Chicago, IL, USA, July 2011.

Part II

Publications

Paper A

Learning to close loops from range data

Authors: Karl Granström, Thomas B. Schön, Fabio T. Ramos and Juan I. Nieto

Edited version of the paper:

K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *The International Journal of Robotics Research*, 30(14):1728–1754, December 2011.

Early versions of this work was presented in:

K. Granström, J. Callmer, F. T. Ramos, and J. I. Nieto. Learning to Detect Loop Closure from Range Data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, Kobe, Japan, May 2009.

K. Granström and T. B. Schön. Learning to Close the Loop from 3D Point Clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, Taipei, Taiwan, October 2010.

Learning to close loops from range data

Karl Granström*, Thomas B. Schön*, Fabio T. Ramos** and Juan I. Nieto**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
{karl,schon}@isy.liu.se

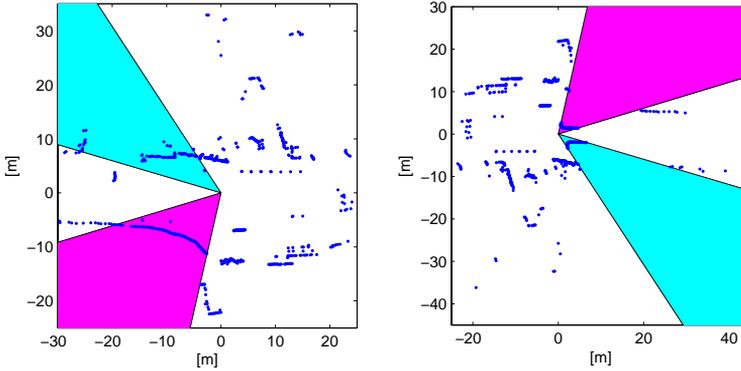
**Australian Centre for Field Robotics
University of Sydney
Sydney, Australia
{f.ramos,
j.nieto}@acfr.usyd.edu.au

Abstract

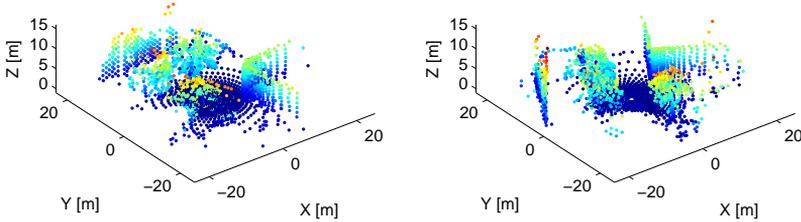
This paper addresses the loop closure detection problem in SLAM, and presents a method for solving the problem using pairwise comparison of point clouds in both 2D and 3D. The point clouds are mathematically described using features that capture important geometric and statistical properties. The features are used as input to the machine learning algorithm AdaBoost, which is used to build a non-linear classifier capable of detecting loop closure from pairs of point clouds. Vantage point dependency in the detection process is eliminated by only using rotation invariant features, thus loop closure can be detected from arbitrary direction. The classifier is evaluated using publicly available data, and is shown to generalize well between environments. Detection rates of 66%, 63% and 53% for 0% false alarm rate are achieved for 2D outdoor data, 3D outdoor data and 3D indoor data, respectively. In both 2D and 3D, experiments are performed using publicly available data, showing that the proposed algorithm compares favorably to related work.

1 Introduction

Loop closure detection is defined as the problem of detecting when the robot has returned to a previously visited location. Being an integral part of the Simultaneous Localization and Mapping (SLAM) problem, loop closure detection has received considerable attention in recent years. In particular, methods using vision sensors have broadly been presented, see e.g. (Goedeme et al., 2006; Tapus and Siegwart, 2006; Fraundorfer et al., 2007; Ho and Newman, 2007; Cummins and Newman, 2008; Angeli et al., 2008; Callmer et al., 2008; Eade and Drummond, 2008; Milford and Wyeth, 2008; Cummins and Newman, 2009; Konolige et al., 2010; Paul and Newman, 2010). Range sensors on the other hand, have not been so widely considered for loop detection, in particular with 2D sensors. This paper addresses the problem of loop closure detection using range sensor measurements and, similarly to many vision solutions, the problem is solved via



(a) 2D example.



(b) 3D example.

Figure 1: Examples of the loop closure detection problem. In (a) two planar point clouds from the same location are shown. Parts of the scene are occluded by vehicles parked along the side of the road in the right point cloud, these parts are highlighted in color. In (b) two 3D point clouds are shown, color is used to accentuate the height. These point clouds are from the same location, which can be identified by the building corner and the wall opposite the corner.

pairwise data comparison. The proposed method applies equally well for both 2D and 3D data (see Figure 1 for examples of the problem in 2D and 3D).

In this paper the loop closure detection problem is cast as a classification task, in which a data pair is classified as either being from the same location, or not. The range sensor measurements, represented as point clouds, are mathematically described using features which capture important statistical and geometrical properties. The features provide an efficient means for dimensionality reduction, and also facilitate easy comparison of the point clouds. Furthermore, the features are fully invariant to rotation, thus enabling loop closure detection from arbitrary direction. Following the feature extraction process, a machine learning algorithm called AdaBoost is used to train a classifier. AdaBoost builds a classifier by combining simple binary classifiers, resulting in a decision boundary which is non-

linear. AdaBoost renders a classifier with good generalization properties which is able to robustly detect loop closure.

Similar point cloud features have been used in (Mozos et al., 2005; Arras et al., 2007; Brunskill et al., 2007). In this work, the set is extended with novel features to better address the problem of loop closure detection. The major contribution of this paper is the formulation of the loop closure detection algorithm with extensive experimental evaluation in urban environments, comparisons to related work using publicly available data sets, and a detailed implementation description.

Early versions of this work have been presented previously (Granström et al., 2009; Granström and Schön, 2010). This paper presents extensions of the previously presented results. In particular, a more detailed and thorough evaluation is presented by using different publicly available datasets. The paper is organized as follows: the next section contains an overview of related work. Section 3 presents the suggested loop closure detection method. A general framework for SLAM using the loop closure methodology is presented in Section 4. Extensive experimental results and comparisons are presented in Section 5, and Conclusions and Future work are presented in Section 6.

2 Related work

In this section we give an overview of related work on large-scale SLAM and loop closure detection, using 2D and 3D range sensors, as well as cameras. A detailed comparison between the work in this paper and the related work using similar sensor setups is given in Section 5.

SLAM algorithms based on raw laser scans have been shown to present a more general solution than classic feature-based (Gutmann and Konolige, 1999). For example, in (Hahnel et al., 2003; Bosse and Zlot, 2008; Newman et al., 2006), raw laser scans were used for relative pose estimation. The mapping approach presented in (Gutmann and Konolige, 1999) joins sequences of laser scans to form local maps. The local maps are then correlated with a global laser map to detect loop closures. Laser range scans are used in conjunction with EKF-SLAM in (Nieto et al., 2007). The authors introduced an algorithm where landmarks are defined by templates composed of raw sensed data. The main advantage claimed is that the algorithm does not need to rely on geometric landmarks as traditional EKF-SLAM. When a landmark is re-observed, the raw template could be augmented with new sensor measurements, thus improving the landmark's representation. The authors also introduced a shape validation measure as a mechanism to enhance data association. In summary, the main advantage in all these works is the ability of the algorithms to work in different environments thanks to the general representation obtained from raw sensor data.

Mapping algorithms based on laser scans and vision have shown to be robust. The work presented in (Ho and Newman, 2005) performs loop closure detection

using visual cues and laser data. Shape descriptors such as angle histograms and entropy are used to describe and match the laser scans. A loop closure is only accepted if both visual and spatial appearance comparisons credited the match. In (Ramos et al., 2007b), laser range scans are fused with images to form descriptors of the objects used as landmarks. The laser scans are used to detect regions of interest in the images through polynomial fitting of laser scan segments while the landmarks are represented using visual features.

An approach to loop closure detection in 2D which is similar to the one taken in this paper is presented in (Brunskill et al., 2007). The authors construct submaps from multiple consecutive point clouds. Before initializing a new submap, it is checked whether the current point cloud is similar to any of the previous submaps. Each submap is described using a set of rotation invariant features, several of which are similar to the features used in this work. Next, AdaBoost is used to train one classifier for each submap, where each classifier test whether or not a point cloud belongs to the corresponding submap. The learning is unsupervised and performed online during SLAM, in contrast to learning in this work which is supervised and performed offline, prior to SLAM. The authors present results from two small scale indoor data sets, and show detection rates of 91% and 83% at precision rates of 92% and 84% for the two data sets, respectively. Comparing to the work presented in this paper, the main difference is that (Brunskill et al., 2007) learn representations for each submap, and require one classifier for each submap, while a general similarity metric between two arbitrary point clouds is learned in this paper. Thus the loop closure detection problem can be solved using just one classifier.

Another example of loop closure detection for 2D point clouds is the work in (Bosse and Zlot, 2008). They use consecutive laser scans to build submaps, typically a submap contains laser scans from tens of meters of travel. The submaps are then compressed using orientation and projection histograms as a compact description of submap characteristics. Entropy metrics and quality metrics are used to compare point clouds to each other. A 51% detection rate for 1% false alarm rate is reported for suburban data. Extending the work on 2D data, keypoints are designed which provide a global description of the point clouds (Bosse and Zlot, 2009a), thus making it possible to avoid pairwise comparison of all local submaps which can prove to be very time consuming for large data sets.

Work on object recognition and shape matching in 2D using point based descriptions includes the work on shape context (Belongie et al., 2002). The shape context is a global descriptor of each point which allows the point correspondence problem to be solved as an optimal assignment problem. For loop closing in 2D, a method which relies on the extraction of linear landmarks is proposed in (Rencken et al., 1999). Loops are detected by matching landmarks from partial maps in structured indoor environments. For global robot localization using 2D laser in unstructured environments, the gestalt features have also been proposed (Walthelm, 2002).

For the similar problem of object recognition using 3D points, regional shape descriptors have been used (Johnson and Hebert, 1999; Frome et al., 2004). Object recognition must handle occlusion from other objects, similarly to how loop closure detection must handle occlusion from moving objects. However object recognition often relies on an existing database of object models. Regional shape descriptors have also been used for place recognition for 3D point clouds (Bosse and Zlot, 2009b). Here, place recognition is defined as the problem of detecting the return to the same place and finding the corresponding relative pose (Bosse and Zlot, 2009a,b), i.e. it includes both relative pose estimation, and what we here define as loop closure detection.

In (Magnusson et al., 2009) results for loop closure detection are presented for outdoor, indoor and underground mine data. The method presented is based on the Normal Distribution Transform (NDT) (Biber and Strasser, 2003), which acts as a local descriptor of the point cloud. After discretizing space into bins, or cubes, the points in each bin are described as either linear, planar or spherical by comparing the size of the covariance matrix eigenvalues. The NDT is exploited to create feature histograms based on surface orientation and smoothness. Invariance to rotation is achieved after scans have been aligned according to the dominant planar surface orientation. The authors show detection rates of 47.0%, 69.6% and 28.6% for 0% false alarm, for outdoor, indoor and mine data, respectively.

In more recent work another method for loop detection for 3D point clouds was presented (Steder et al., 2010). The point cloud is transformed into a range image, from which features are extracted by computing the second derivative of the image gradient. The extracted features are compared to features from previous scans using the Euclidean distance. Using feature correspondence, a relative rotation and translation can be computed, and the operation is evaluated by computing a score for how well the two scans are aligned. Rotation invariance is achieved by orienting image patches along the world z-axis. According to the authors this does not restrict the performance of the method as long as the robot moves on a flat surface. This assumption is however not valid in all environments, e.g. underground mines (Magnusson et al., 2009).

Work on vision based loop closure detection have been presented in (Cummins and Newman, 2008, 2009), with detection rates of up to 37% and 48% at 0% false alarm for the City Centre and the New College datasets (Smith et al., 2009), respectively. The authors show results for very large data sets (1000km), and also present interesting methods to handle occlusion, a problem that is often present in dynamic environments. The work is extended via inclusion of a laser range sensor in (Paul and Newman, 2010), and the detection rate for the New College data set is increased to 74%. Another vision based loop closure detection method is suggested in (Callmer et al., 2008). SURF features are extracted from images, and classified as words using Tree-of-Words. A spatial constraint is imposed by checking nearest neighbors for each word in the images. A similar approach using visual words for monocular SLAM is taken in (Eade and Drummond, 2008),

however the vocabulary is built online in contrast to offline as in (Cummins and Newman, 2008, 2009; Callmer et al., 2008). In a Graph-SLAM system, loops are closed when new edges are created. A SLAM system inspired by rodents is presented in (Milford and Wyeth, 2008). The authors use a monocular camera to collect data over a 66km trajectory with multiple nested loops. More than 51 loops are closed, with no false loops, however there is no explicit loop closure detection. A topological mapping method where loop closure is detected via strong geometrical constraints for stereo images is presented in (Konolige et al., 2010). Another topological method using vision is the work by (Tapus and Siegwart, 2006). It should be noted that it is difficult to compare results from different types of sensors.

A classification approach based on point cloud features and AdaBoost has been used for people detection in cluttered office environments (Arras et al., 2007) and indoor place recognition (Mozos et al., 2005). For people detection the point clouds were segmented and each segment classified as either belonging to a pair of legs or not. Detection rates of over 90% were achieved. For place recognition multiple classes (> 2) are generally used. For this reason the results do not easily compare to the present loop closure detection problem, which has two classes (either same place or not).

3 Loop closure detection

Loop closure detection can be seen as a place recognition problem - it consists of detecting that the robot has previously visited the current location. The problem is central to SLAM, as it allows the estimated map and robot location to be refined. This section presents the suggested loop closure detection algorithm. Here, we pose the loop closure problem as being the problem of determining whether or not two point clouds are from the same location or not. A mobile robot equipped with a range sensor moves through unknown territory and acquires point clouds \mathbf{p}_k at times t_k along the trajectory. A point cloud \mathbf{p}_k is defined as

$$\mathbf{p}_k = \{p_i^k\}_{i=1}^N, \quad p_i^k \in \mathbb{R}^D, \quad (1)$$

where N is the number of points in the cloud and D is the dimensionality of the data, here $D = 2$ or $D = 3$. The points are given in Cartesian coordinates

$${}^c p_i^k = \begin{cases} \begin{bmatrix} x_i^k & y_i^k \end{bmatrix}^T, & \text{if } D = 2 \\ \begin{bmatrix} x_i^k & y_i^k & z_i^k \end{bmatrix}^T, & \text{if } D = 3 \end{cases} \quad (2)$$

but can of course be converted into polar/spherical coordinates

$${}^p p_i^k = \begin{cases} \begin{bmatrix} r_i^k & \varphi_i^k \end{bmatrix}^T, & \text{if } D = 2 \\ \begin{bmatrix} r_i^k & \varphi_i^k & \psi_i^k \end{bmatrix}^T, & \text{if } D = 3 \end{cases} \quad (3)$$

using the appropriate Cartesian to polar/spherical transformation. Here r , φ and ψ is range, horizontal angle and vertical angle, respectively. For simplicity,

time index k and the differentiation between coordinate systems, i.e. c and p in Equations (2) and (3), is dropped in the remainder of the paper. In Appendix A, where the features are defined, it will be clear from context which coordinate system that is intended.

After moving in a loop the robot arrives at a previously visited location, and the two point clouds, acquired at different times, should resemble each other. A comparison of the point clouds is performed in order to determine if a loop closure has occurred or not. To facilitate this comparison, two types of features are first introduced. From the features a classifier is then learned using AdaBoost. The learned classifier is used to detect loop closure in the experiments.

3.1 Algorithm overview

Our loop detection algorithm uses the same principle as in other loop detection approaches, i.e. pairwise comparison of data, see e.g. (Bosse and Zlot, 2008; Callmer et al., 2008; Cummins and Newman, 2008; Magnusson et al., 2009; Steder et al., 2010). Each point cloud is described using a large set of rotation invariant features. These features are combined in a non-linear manner using a boosting classifier which outputs the likelihood of the two point clouds being from the same location.

There are two main parts to the algorithm, the first is the learning phase where a classifier is learned from training data. The second part is the classification phase, where the learned classifier is used to classify pairs of point clouds in SLAM experiments. A diagram of the algorithm is given in Figure 2. In the learning phase (left), pairs of point clouds with corresponding class labels y are found in point cloud data sets. From each point cloud features are computed. Examples of the features employed are mean range, area (in 2D) or volume (in 3D), distance etc. A detailed description of the features is presented in subsection 3.2 and Appendix A. The difference between the features from point clouds k and l is called the set of extracted features, and is denoted $\mathbf{F}_{k,l}$. The set of extracted features with corresponding class labels are input to AdaBoost (Freund and Shapire, 1995), a machine learning procedure which greedily builds a strong classifier $\mathbf{c}(\mathbf{F}_{k,l})$ by a linear combination of simple, so called weak, classifiers. When the weak classifiers are combined into a strong classifier, the resulting decision boundary is non-linear. The same strategy has been employed for face detection in (Viola and Jones, 2004).

In the classification phase of our algorithm, Figure 2 (right), the learned classifier is used to detect loop closure in SLAM experiments. The SLAM framework used here is trajectory based, meaning that the state vector contains a history of previous poses. The particular SLAM framework is described in Section 4.

3.2 Features

The main reason for working with features is the ability to compress the information in point clouds by defining meaningful statistics describing shape and other properties – working with n_f features is easier (e.g. requires less memory

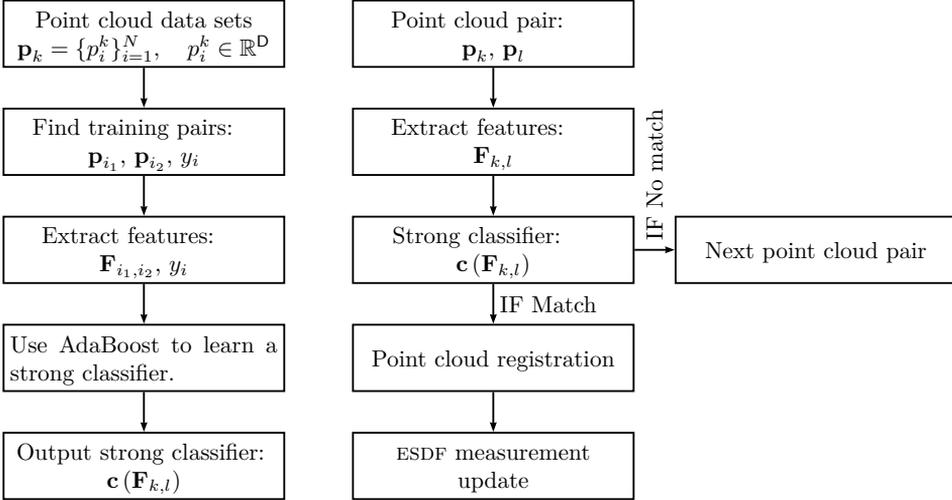


Figure 2: Algorithm overview. Left diagram, learning phase. Right diagram, classification phase. In the learning phase, one or more point cloud data sets are used to learn a strong classifier. The learning phase is run first in our algorithm. In the classification phase the strong classifier is used to detect loop closure in SLAM experiments. In our SLAM implementation, the classification phase is run each time a new pose has been added to the state vector.

and is less computationally expensive) than working with the full point clouds since $n_f \ll N$. In this work, two types of features f_j are used. The first type is a function that takes a point cloud as input and returns a real number. Typically, features that represent geometrical or statistical properties of the point cloud are used, e.g. volume of point cloud or average range. The features of the first type are collected in a vector $\mathbf{f}_k \in \mathbb{R}^{n_f^1}$, where k again refers to the time t_k when the point cloud was acquired. Here, n_f^1 is the number of features of the first type. The second type of feature used is a range histogram with bin size b_j . In total, n_f^2 histograms are computed, giving a total of $n_f^1 + n_f^2 = n_f$ features.

In order to facilitate comparison of two point clouds from times t_k and t_l , the features of both types are considered. For the first type, elementwise absolute value of the feature vector difference is computed,

$$\mathbf{F}_{k,l}^1 = |\mathbf{f}_k - \mathbf{f}_l|. \quad (4)$$

The underlying idea here is that point clouds acquired at the same location will have similar feature values \mathbf{f}_k and \mathbf{f}_l , and hence each element of $\mathbf{F}_{k,l}$ should be small. For the second type of feature, for each bin size b_j the correlation coefficient for the two corresponding range histograms is computed. Here, the underlying idea is that point clouds acquired at the same location should have similar range histograms, and thus the correlation coefficient should be close to 1. The

correlation coefficients are collected in a vector $\mathbf{F}_{k,l}^2$, and the comparisons of both types of features are concatenated in a vector as

$$\mathbf{F}_{k,l} = [\mathbf{F}_{k,l}^1, \mathbf{F}_{k,l}^2]. \quad (5)$$

$\mathbf{F}_{k,l}$ is referred to as the set of extracted features for two point clouds indexed k and l .

In (Granström et al., 2009) 20 features are used in 2D, in (Granström and Schön, 2010) those features are extended to 3D and augmented with new features. Some of the features used here are the same regardless of dimension, e.g. mean range, while other features are generalized, e.g. from area in 2D to volume in 3D. Similar 2D features can be found in (Mozos et al., 2005; Arras et al., 2007; Brunskill et al., 2007). In total, $n_f = 44$ features are used in 2D and $n_f = 41$ features are used in 3D. For formal definitions, see Appendix A.1.

3.3 Classification using AdaBoost

Boosting is a machine learning method for finding combinations of simple base classifiers in order to produce a form of committee whose performance can be significantly better than any one of the base classifiers used alone. The simple base classifiers need to be just slightly better than a random guess, thus they are called weak classifiers, see e.g. (Bishop, 2006). The resulting combination is better than the best individual weak classifier, and analogously the resulting classifier is thus called strong. Each weak classifier is learned using a weighted form of the data set, where the weighting of each data point depends on the performance of the previous weak classifiers.

A widely used form of boosting is AdaBoost, which constructs a strong classifier by a linear combination of weak classifiers (Freund and Shapire, 1995). When the weak classifiers are combined into a strong classifier, the resulting decision boundary is non-linear. As more weak classifiers are added, the classification error on the training data converges towards zero, and eventually becomes zero. Although this might be interpreted as overfitting, AdaBoost has been shown to generalize well on testing data (Freund and Shapire, 1995).

Although later generalized to multiple classes, AdaBoost was originally designed for problems with two classes. Here, the two classes are called positive and negative, or p and n, respectively. The positive class consists of point cloud pairs from the same location, the negative class consists of point cloud pairs from different locations. As input to the AdaBoost learning algorithm, n hand-labeled training data pairs are provided,

$$(\mathbf{F}_{1,1_2}, y_1), \dots, (\mathbf{F}_{i_1, i_2}, y_i), \dots, (\mathbf{F}_{n_1, n_2}, y_n), \quad (6)$$

where each data point \mathbf{F}_{i_1, i_2} has a corresponding class label y_i . Let \mathbf{F}_i be a compact way of writing \mathbf{F}_{i_1, i_2} . To learn a classifier using AdaBoost, data points from each class are needed. Let N_p and N_n be the number of training data points belonging to p and n, respectively, i.e. $n = N_n + N_p$.

The data labels in the two class problem are defined as

$$y_i = \begin{cases} 1 & \text{if } \mathbf{f}i \in \text{p} \\ 0 & \text{if } \mathbf{f}i \in \text{n} \end{cases} \quad (7)$$

In the AdaBoost algorithm, each data pair $(\mathbf{f}i, y_i)$ is given a weight w_t^i , where t denotes the specific iteration of the algorithm. The weights are initialized as $w_1^i = \frac{1}{2N_n}$ if $y_i = 0$, or $w_1^i = \frac{1}{2N_p}$ if $y_i = 1$. This initialization ensures that each class is given half the weight of the data, and all data pairs within a class are given an equal weight.

After initialization, AdaBoost iteratively adds weak classifiers to a set of previously added weak classifiers. The weak classifiers used here are decision stumps, i.e. one node decision trees, defined as

$$c(\mathbf{f}i, \theta) = \begin{cases} 1 & \text{if } p\mathbf{f}i^{(f)} < p\lambda \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

with parameters $\theta = \{f, p, \lambda\}$, where $\mathbf{f}i^{(f)}$ is the selected component f of $\mathbf{f}i$, p is the polarity ($p = \pm 1$), and $\lambda \in \mathbb{R}$ is a threshold. The result of a weak classifier (8) is that the input space is partitioned into two half spaces, separated by an affine decision boundary which is parallel to one of the input axes.

In each iteration, the weak classifier that minimizes the weighted classification error with respect to θ is chosen. Given the parameters of the best weak classifier, the training data is classified and the weights of the mis-classified data are increased (or, conversely, the weights of the correctly classified data are decreased). Further, using the classification error a weight α_t is computed for the best weak classifier. Details on how the weights are computed are given below.

This procedure is repeated until T weak classifiers have been computed. Weak classifiers can be added several times in each dimension of \mathbb{R}^{n_f} , each time with a new polarity and threshold, i.e. same f and new p and λ . The weighted combination of T weak classifier together create the strong classifier. A detailed presentation of AdaBoost is given in Algorithm 3.

In this work, to find the best weak classifier, we employ a similar technique as is used in (Viola and Jones, 2004). The search for the best weak classifier is summarized in Algorithm 4. In our implementation, we search over all features each time we find the best weak classifier. With just over 40 features, doing so does not pose a significant complexity issue. However, if the number of features were in the order of thousands, as in (Viola and Jones, 2004), searching over all features could prove to be very time consuming.

4 Simultaneous localization and mapping

In this section we briefly outline the SLAM framework used for testing the new method in detecting loop closure. The algorithm is well known and not part of

Algorithm 3 AdaBoost**Input:** $(\mathbf{F}_1, y_1), \dots, (\mathbf{F}_n, y_n)$ **Initialize weights:** $W_1^i = \frac{1}{2N_n}$ if $y_i = 0$, $W_1^i = \frac{1}{2N_p}$ if $y_i = 1$ 1: **for** $t = 1, \dots, T$ **do**

2: Normalize the weights:

$$\widetilde{W}_t^i = \frac{W_t^i}{\sum_{j=1}^{N_n+N_p} W_t^j}, \quad i = 1, \dots, N_n + N_p \quad (9)$$

3: Select the best weak classifier, i.e. the one that minimizes the weighted error,

$$\theta_t = \arg \min_{\theta} \sum_{i=1}^n \widetilde{W}_t^i |c(\mathbf{F}_i, \theta) - y| \quad (10)$$

where $\theta = \{f, p, \lambda\}$.4: Define $c_t(\mathbf{F}_i) = c(\mathbf{F}_i, \theta_t)$, and let ε_t be the corresponding weighted error.

5: Update the weights:

$$W_{t+1}^i = \widetilde{W}_t^i \beta_t^{1-e_i}, \quad (11)$$

where $e_i = 0$ if \mathbf{F}_i is classified correctly and $e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$.6: **end for**

The strong classifier is:

$$\mathbf{c}(\mathbf{F}_{k,l}) = \frac{\sum_{t=1}^T \alpha_t c_t(\mathbf{F}_{k,l})}{\sum_{t=1}^T \alpha_t} \in [0, 1] \quad (12)$$

where $\alpha_t = \log \frac{1}{\beta_t}$. The closer $\mathbf{c}(\mathbf{F}_{k,l})$ is to one, the higher the loop likelihood is. To obtain a binary decision, i.e. loop or no loop, the classification likelihood can be thresholded using a threshold

$$K \in [0, 1]. \quad (13)$$

Output: $\mathbf{c}(\mathbf{F}_{k,l})$

our main contribution, hence we only provide the specific design choices made and refer to the relevant references for the exact implementation details.

4.1 Exactly Sparse Delayed-state Filters

The Exactly Sparse Delayed-state Filter (ESDF), a delayed state extended information filter, maintains a delayed state vector containing the poses where point clouds were acquired. The state vector is augmented with a new pose when a new point cloud is acquired. The state information matrix is sparse without approximation, which results in an estimation comparable to the full covariance matrix solution. Using sparse solutions, like SEIF (Thrun et al., 2004), has been shown to be inconsistent (Eustice et al., 2005). Using the ESDF, prediction and update can be performed in constant time regardless of the information matrix size. Refer to (Eustice et al., 2006) for details on the implementation.

Algorithm 4 Find the best weak classifier

Input: $(\mathbf{F}_1, y_1), \dots, (\mathbf{F}_n, y_n)$ with corresponding weights W_t^1, \dots, W_t^n .

Initialize: $T_- = \sum_{i:y_i=0} W_t^i$ and $T_+ = \sum_{i:y_i=1} W_t^i$.

1: **for** $d = 1, \dots, n_f$ **do**

2: Sort the data in the current feature dimension d in ascending order, and let i^1, \dots, i^n be the corresponding index, i.e. $\mathbf{f}i^{1(d)} \leq \mathbf{f}i^{2(d)} \leq \dots \leq \mathbf{f}i^{n(d)}$.

3: Compute the cumulative sum of weights for each class,

$$S_-^j = \sum_{k=1}^j (1 - y_{i^k}) W_t^{i^k} \quad j = 1, \dots, n \quad (14a)$$

$$S_+^j = \sum_{k=1}^j y_{i^k} W_t^{i^k} \quad j = 1, \dots, n \quad (14b)$$

4: Compute errors $\varepsilon_1^j = S_+^j + T_- - S_-^j$ and $\varepsilon_2^j = S_-^j + T_+ - S_+^j$.

5: Find the minimum error, $\varepsilon_d = \min_{j,k} \varepsilon_k^j$. Let ℓ, γ be the minimizing arguments, i.e. $\varepsilon_d = \varepsilon_\gamma^\ell$.

6: Compute the threshold,

$$\lambda_d = \frac{\mathbf{F}_{i^\ell}^{(d)} + \mathbf{F}_{i^{\ell+1}}^{(d)}}{2} \quad (15)$$

7: Compute polarity,

$$p_d = \begin{cases} -1 & \text{if } \gamma = 1 \\ 1 & \text{if } \gamma = 2 \end{cases} \quad (16)$$

8: **end for**

Find the feature dimension with lowest error,

$$f = \arg \min_d \varepsilon_d \quad (17)$$

and set $p = p_f$ and $\lambda = \lambda_f$. The optimal parameters, and corresponding error, are $\theta_t = \{f, p, \lambda\}$ and $\varepsilon_t = \varepsilon_f$.

Output: θ_t and ε_t

4.2 Robot pose, process and measurement models

In this paper we use the coordinate frame notation introduced by Smith et al. (1990) to handle the robot pose, process model and measurement model. Let $\mathbf{x}_{i,j}$ denote the location of coordinate frame j with respect to coordinate frame i . In 2D, $\mathbf{x}_{i,j}$ is a 3-DOF pose consisting of $(x_{i,j}, y_{i,j})$ -position and heading angle $\psi_{i,j}$. In 3D, $\mathbf{x}_{i,j}$ is a 6-DOF pose consisting of $(x_{i,j}, y_{i,j}, z_{i,j})$ -position and Euler angles $(\phi_{i,j}, \theta_{i,j}, \psi_{i,j})$ representing roll, pitch and heading angles. Here, the roll, pitch and heading definitions from (Eustice, 2005) are used. This Euler representation is singular at pitch $\theta = \pm 90^\circ$, however ground robots rarely operate at such configurations and the singularity has not been any problem in our SLAM experiments. It can be noted that an alternative angle representation that does not suffer from singularities could have been used, e.g. axis angle or quaternions. Using $\mathbf{x}_{k,l}$ and

$\mathbf{x}_{l,m}$, the location of coordinate frame m with respect to coordinate frame k can be expressed using the compounding operator \oplus introduced in (Smith et al., 1990) as

$$\mathbf{x}_{k,m} = \mathbf{x}_{k,l} \oplus \mathbf{x}_{l,m}. \quad (18)$$

Using the inverse compounding operator \ominus from (Smith et al., 1990), the location of coordinate frame k with respect to coordinate frame l is expressed as

$$\mathbf{x}_{l,k} = \ominus \mathbf{x}_{k,l}. \quad (19)$$

Formal mathematical definitions of the compounding operators \oplus and \ominus in 2D and 3D can be found in Appendix A.2. Subsequently, if the locations of coordinate frames l and m with respect to coordinate frame k are known, the location of m with respect to l is expressed as

$$\mathbf{x}_{l,m} = \ominus \mathbf{x}_{k,l} \oplus \mathbf{x}_{k,m}. \quad (20)$$

Note, that since each \mathbf{x} consists of a position and a heading, the compounding operator is just a short hand representation for combinations of rigid body transformations. In our SLAM experiments the pose

$$\mathbf{x}_{0,k} = \begin{cases} [x_{0,k} \ y_{0,k} \ \psi_{0,k}]^T & \text{in 2D} \\ [x_{0,k} \ y_{0,k} \ z_{0,k} \ \phi_{0,k} \ \theta_{0,k} \ \psi_{0,k}]^T & \text{in 3D} \end{cases} \quad (21)$$

is the location of point cloud k 's local coordinate frame in the global coordinate frame 0. Both process and measurement model are defined as coordinate frame operations using the compounding operator. The process, or motion, model is

$$\mathbf{x}_{0,k+1} = f(\mathbf{x}_{0,k}, \mathbf{x}_{k,k+1}) + \mathbf{w}_{k+1} \quad (22a)$$

$$= \mathbf{x}_{0,k} \oplus \mathbf{x}_{k,k+1} + \mathbf{w}_{k+1}, \quad (22b)$$

where $\mathbf{x}_{k,k+1}$ is computed using point cloud registration, Section 4.3, and \mathbf{w}_{k+1} is a white Gaussian process noise. After a loop closure has been detected between point clouds m and n , the corresponding relative pose $\mathbf{x}_{m,n}$ is computed using the measurement model, defined as

$$\mathbf{x}_{m,n} = h(\mathbf{x}_{0,m}, \mathbf{x}_{0,n}) + \mathbf{e}_{m,n} \quad (23a)$$

$$= \ominus \mathbf{x}_{0,m} \oplus \mathbf{x}_{0,n} + \mathbf{e}_{m,n} \quad (23b)$$

$$= \mathbf{x}_{m,0} \oplus \mathbf{x}_{0,n} + \mathbf{e}_{m,n}, \quad (23c)$$

where $\mathbf{e}_{m,n}$ is white Gaussian measurement noise.

4.3 Point cloud registration

Point cloud registration, also referred to as scan matching, is the process of finding a rigid body transformation (rotation and translation) that aligns two point clouds to each other. Typically, this is performed by minimizing a cost function, e.g. the sum of distances to nearest neighbor points. There are a number of different methods proposed in the literature, in this work we have used four different methods; the well known Iterative Closest Point (ICP) (Besl and McKay, 1992; Chen and Medioni, 1992; Zhang, 1994), 3D NDT (Magnusson et al., 2007),

CRF-Match (Ramos et al., 2007a) and an implementation of the histogram based method by (Bosse and Zlot, 2008).

In 2D, we use ICP to compute the vehicle motion, i.e. to compute $\mathbf{x}_{k,k+1}$ in (22b). After loop closure has been detected, we use either the histogram method or CRF-Match to find an initial point cloud registration, which is then refined using ICP.

In 3D we use 3D NDT, initialized by odometry to compute vehicle motion. We have performed a SLAM experiment on a publicly available indoor data set, and for this data the consecutive relative poses are available together with the point clouds. After loop closure detection, we use ICP to compute the relative pose. Here, ICP is initialized with the relative pose estimate obtained from the ESDF state vector. While this method works well for the particular SLAM experiment presented here, in a general SLAM solution a point cloud registration method that does not rely on a good initial guess would be needed.

5 Experimental results

This section presents the results from the experiments performed. We examine the proposed method by evaluating the strong classifiers properties, and by doing SLAM experiments in both 2D and 3D. The classifier is evaluated in terms of detection rate (D), missed detection rate (MD) and false alarm rate (FA). The rates are defined as

$$D = \frac{\# \text{ positive data pairs classified as positive}}{\# \text{ positive data pairs}},$$

$$MD = \frac{\# \text{ positive data pairs classified as negative}}{\# \text{ positive data pairs}},$$

$$FA = \frac{\# \text{ negative data pairs classified as positive}}{\# \text{ negative data pairs}}.$$

These rates are important characteristics for any classification or detection problem, and typically it is difficult to achieve low MD and low FA simultaneously. Instead, a choice has to be made as to which error is more important to minimize. For the loop closing problem, we argue that the main concern is minimizing FA , while keeping MD as low as possible. A relevant question is then how low FA should be, since lowering FA further comes at the price of higher MD .

In previous work, D has been reported at 1% FA (Bosse and Zlot, 2008), in other work D has been reported at 0% FA (or equivalently at 100% precision) (Magnusson et al., 2009; Cummins and Newman, 2008), yet others report D at 0.01% FA (Callmer et al., 2008). While it is very important to keep FA low, it is possible to find and reject false alarms in subsequent stages, e.g. when the relative pose is found via point cloud registration (Bosse and Zlot, 2008), or using a cascade of several methods (Bosse and Zlot, 2009a). However, even if a combination of methods is used, the false alarms have to be rejected at some stage since closing a false loop could prove disastrous for the localization and/or mapping process. Further, finding a subset of loop closures is typically sufficient to produce good

results (Magnusson et al., 2009; Bosse and Zlot, 2008; Cummins and Newman, 2008, 2009). Therefore, the detection rate at 0% false alarm is more robust. However, for completeness and ease of comparison, results at both 0% and 1% false alarm are presented.

The experiments in Section 5.2 were conducted using k -fold cross validation on the data sets. Note that in each experiment the validation portion of the data was fully disjoint from the training portion. The partitioning into folds was performed by randomly permuting the order of the data. Since different permutations give slightly different results, k -fold cross validation was performed multiple times, each time with a different permutation of the data. The results presented are the mean of the cross validations. The data used in experiments is presented in Section 5.1. After evaluating the 2D and 3D classifiers, Section 5.2, the classifiers are tested in SLAM experiments which are presented in Section 5.3. The experiments are compared to the estimated SLAM trajectories with the dead reckoning sensors, and with GPS when it is available. The resulting SLAM maps are also shown, overlaid on aerial photographs in the outdoor cases. The results are summarized, and a comparison to related work is given, in Section 5.4.

5.1 Data

In this subsection we describe the data used in the 2D and 3D experiments. Three of the six data sets are publicly available, references to the data repositories are provided. The datasets used for training are divided into two classes, positive and negative. Five of the data sets contain a large quantity of point clouds, thus making it possible to find tens of thousands of training pairs. However, to keep the computational cost tractable, the amount of training pairs were limited.

2D data

For the 2D experiments, four different datasets were used. The first data set, called *kenmore_pradoroof* (*ken*), is publicly available (Howard and Roy, 2003). Thanks to Michael Bosse, Autonomous Systems Laboratory, CSIRO ICT Centre, Australia, for providing the data set. It has maximum measurable range $r_{\max} = 50\text{m}$ and horizontal angular resolution $\delta_{\varphi} = 1^{\circ}$. The data set is approximately 18km long. The last three data sets all have maximum measurable range $r_{\max} = 50\text{m}$ and horizontal angular resolution $\delta_{\varphi} = 0.5^{\circ}$. Two of them, *Sydney 1* (*syd1*) and *Sydney 2* (*syd2*), were acquired in a residential and business area close to the University of Sydney, Sydney, Australia. These two datasets are approximately 0.65 and 2 km long. Using an initial set of 50 data pairs for each class, a classifier was learned and used for SLAM experiments using the *ken*, *syd1* and *syd2* data sets. The resulting trajectories are shown in Figure 3.

Using the estimated trajectories, positive and negative data pairs were extracted based on the translational distance between the poses at which the point clouds were acquired. For each of the three datasets, positive pairs were taken as all pairs where the translational distance was less than or equal to 1m, 2m and 3m. Negative pairs were obtained by taking a random subset of remaining data pairs, such that for each translational distance the number of positive N_p and negative

Table 1: Number of 2D point cloud pairs, at various translational distances. The point cloud pairs were used for training and evaluation of the classifier. N_p and N_n are the number of positive and negative point cloud pairs, respectively.

Data set	Dist. [m]	N_p, N_n
<i>ken</i>	1	1321
	2	4242
	3	7151
<i>syd1</i>	1	31
	2	568
	3	956
<i>syd2</i>	1	286
	2	2039
	3	3570

N_n data pairs are equal. The number of data pairs for each dataset and each translational distance is shown in Table 1.

Careful visual examination of the *ken* trajectory in Figure 3c shows parts for which the trajectory estimation was of lower quality. The main reason for this was that our scan registration, an implementation of the histogram method from (Bosse and Zlot, 2008), failed to find the correct rotation and translation when true loops were detected from the opposite direction. Thus several detected loop closures from the opposite direction could not be included in the estimation. Since we want to use the SLAM results for finding training pairs at certain translational distances, we also want to be certain that the translational distance computed from the SLAM results is close to the true translational distance. For this reason, only the first half of the *ken* data set, for which we could estimate the trajectory with higher accuracy (only loops from the same direction), was used to find positive and negative training data. This trajectory is shown in Figure 3d. In addition to being used for finding training data, the *ken* data set is also used to evaluate the classifier’s dependence to translation.

The fourth dataset, Sydney 3 (*syd3*), was also collected around the University of Sydney and is approximately 2 km long. This dataset contains point clouds with just a 180° field of view, and was therefore not used for learning the classifier. Instead it was integrated in a SLAM experiment, where GPS was used to collect ground truth data. All four data sets were collected using planar SICK laser range sensors. Placing two such sensors “back-to-back” gives a full 360 degree view of the environment. The sensors sweep from right to left, thus introducing an order for the range measurement. Some of the features defined in Appendix A use this ordering of the points when the feature value is computed.

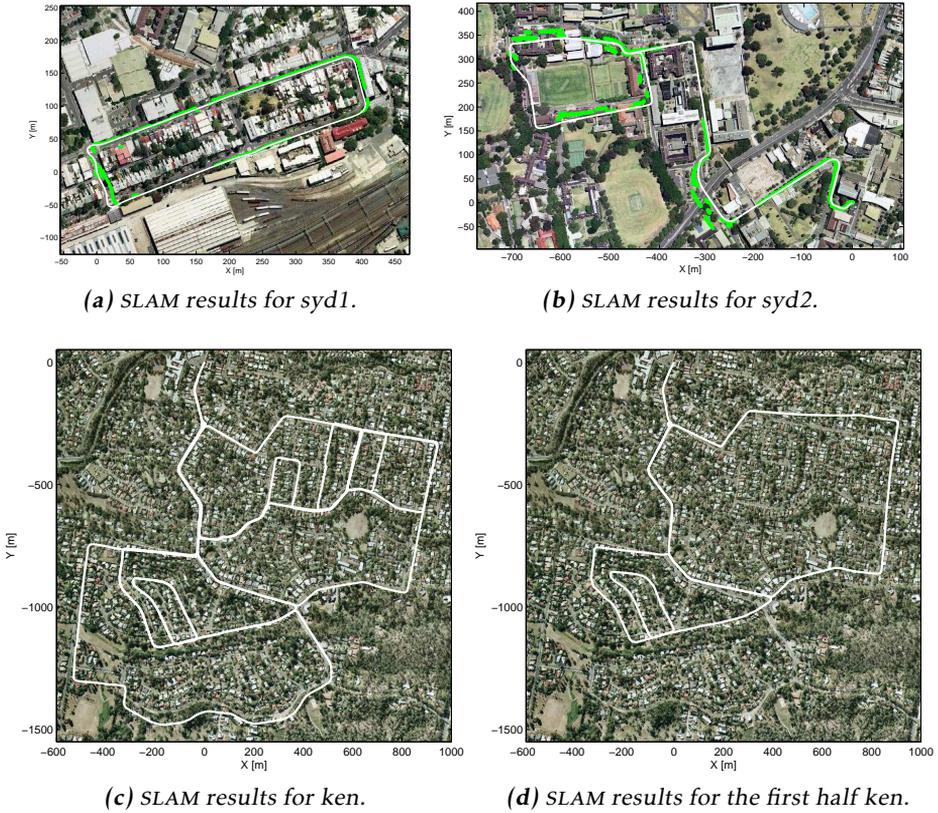


Figure 3: Estimated SLAM trajectories (white) overlaid on Google maps images, and compared to GPS (green) when available. Note that the GPS signal becomes increasingly unreliable under trees and tall buildings. The SLAM trajectories were used to find point cloud pairs for training and evaluation of the classifier.

3D data

In the 3D experiments, two datasets were used, both are publicly available (Wulf, 2009). Thanks to Oliver Wulf, Leibniz University, Germany and Martin Magnusson, AASS, Örebro University, Sweden for providing the data sets. The first one, Hannover 2 (*hann2*), contains 924 outdoor 3D scans from a campus area, covering a trajectory of approximately 1.24 km. Each 3D point cloud contains approximately 16600 points with a maximum measurable range of 30m. From this dataset 3130 positive data pairs (point clouds from the same location) and 7190 negative data pairs (point clouds from different locations) were selected. The positive data pairs were chosen as the scan pairs taken less than 3m apart (Magnusson et al., 2009). The negative data were chosen as a random subset of the remaining data pairs, i.e. those more than 3m apart.

The second dataset, AASS-loop (AASS), contains 60 indoor 3D scans from an office environment, covering a trajectory of 111 m. Each 3D point cloud contains approximately 112000 points with a maximum measurable range of 15m. From this dataset 16 positive and 324 negative data pairs are taken. The positive data pairs are those taken less than 1m apart (Magnusson et al., 2009), the negative data pairs are a random subset of the remaining data pairs. Due to the limited number of positive data pairs, we chose to not use all negative data. The impact of few data pairs of one class, called unbalanced data, is discussed further in the paper.

Both 3D data sets were acquired using 2D planar laser range finders, and 3D point clouds were obtained using pan/tilt units. Each 3D point cloud thus consists of a collection of 2D planar range scans. The points in each 3D point cloud can be ordered according to the order in which the points are measured by the sensor setup. Some of the features defined in Appendix A use this ordering of the points when the feature value is computed.

5.2 Classifier evaluation

In this section we evaluate the classifiers learned previously. An important aspect of any robotics application is computational complexity. If a method is to be implemented on a robot, it is important that it can be computed in real time in order to not significantly delay the robot’s functionality. The computational times for different parts of the suggested method for loop closure detection are presented here. When learning a classifier, an initial important step is to determine an appropriate number of training rounds T (cf. Algorithm 3) for the classifier. Training should proceed as long as the validation error decreases, but not longer to avoid overfitting and to keep computational complexity low. Another important aspect is which features are the most beneficial to the final strong classifier. This is verified in two ways: 1) by considering which features are selected in early training iterations, and, 2) by removing features from the training data and checking how they affect the classifier’s performance. The strong classifier’s receiver operating characteristics (ROC) are evaluated, and a comparison is made between 2D and 3D performance by downsampling 3D data to 2D. The classifiers dependence to translation is also evaluated, as well as how it handles dynamic objects. Finally, the difficulty posed by repetitive structures in the environment is addressed.

Computational complexity

The code used in this work was implemented in Matlab and run on a 2.83GHz Intel Core2 Quad CPU with 3.48 GB of RAM running Windows. It should be noted that the implementation is not optimized for speed. The timing results are presented in Table 2. The times to compute the features are averages over all point clouds in the data sets *syd2*, *hann2* and AASS. As expected the time increases with the number of points in each cloud. Computing the features only needs to be performed once per point cloud in a SLAM experiment. Since comparing the features and computing $\mathbf{c}(\mathbf{F}_{k,l})$ are the same operations in both 2D and 3D, the

Table 2: Execution time of loop closure detection classifier, all times in ms. The times to compute the features are shown separately for 2D and 3D data, respectively. Comparing the features is a procedure that is equal in 2D and 3D, thus the time to compare the features is the same in 2D and 3D.

Compute features	<i>syd2</i>	<i>hann2</i>	AASS
Type 1	14.35	15.96	206.11
Type 2	0.22	3.38	18.99
Total	14.57	19.34	225.10
Total per point	20.2×10^{-3}	1.17×10^{-3}	2.00×10^{-3}

Compare features	Time
Type 1	6.58×10^{-3}
Type 2	0.824
Total	0.831
Compute $\mathbf{c}(\mathbf{F}_{k,l})$	0.78

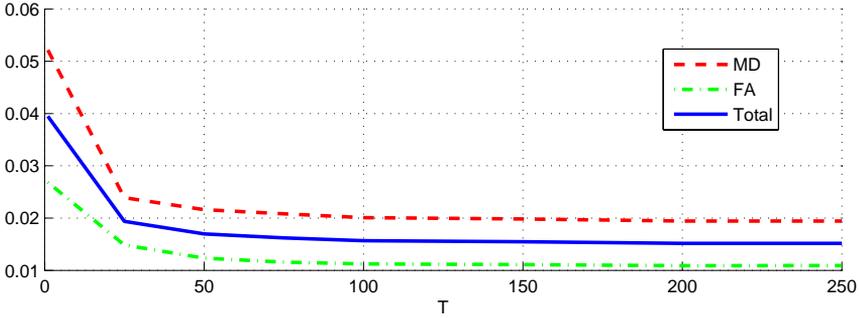
presented times are averages over the training pairs for all 2D and 3D data sets. Comparing features and classifying a set of extracted features takes about 1.6ms when $T = 50$ weak classifiers are used. Training a strong classifier for $T = 50$ iterations takes 15s when about 10000 data pairs are used.

Number of training rounds T

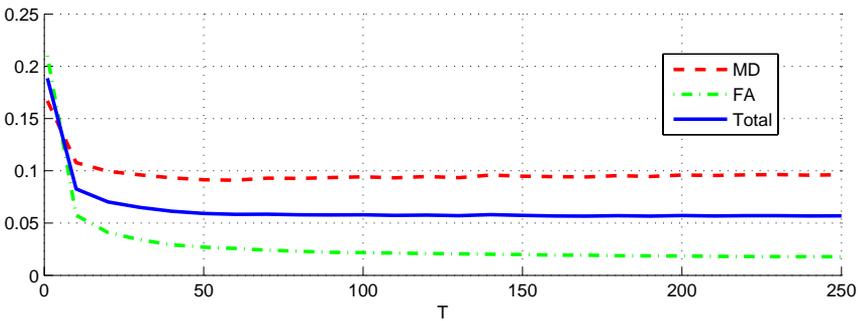
Strong classifiers were trained for different values of T , the resulting error rates are shown in Figure 4. The total error rate is the ratio between the number of misclassified data pairs and the total number of data pairs. As can be seen in Figure 4, the validation error levels decrease as the learning algorithm iterates up until about 50 training iterations, when the validation error levels stop decreasing. Hence, $T = 50$ was chosen for all subsequent experiments in both 2D and 3D.

Most informative features

An interesting aspect of the suggested solution to the loop closure detection problem is which features are the most informative for classification. In each training iteration of the learning phase, the weak classifier that best improves performance is selected. Each feature can be chosen multiple times, each time with a new polarity and threshold. The features that are chosen in early training iterations will have a larger weight than features chosen in later training iterations. Therefore, we considered the features chosen in the first five training rounds in this analysis. To further examine the importance of the features, strong classifiers were learned from the training data after removing the features individually. The best features are those that negatively affected the validation error rate the most when removed. Results for the 2D data are presented in Table 3, 3D results are presented in Table 4. In the tables, Test 1 shows which features were chosen in the first five training rounds, Test 2 indicates the features whose removal resulted in the highest validation error rates.



(a) 2D results.



(b) 3D results.

Figure 4: Error rates during AdaBoost training plotted against the number of training rounds T . For both 2D and 3D data the validation error levels stop decreasing after about 50 training rounds, hence $T = 50$ is an appropriate choice for the subsequent experiments.

Results for the 2D data are in Table 3. Both tests suggest that for *ken*, features 4 (average of all ranges), 44 (range histogram with bin size 3m) and 22 (range kurtosis for all ranges) are most informative. For *syd1*, feature 23 (mean relative range) appears in both tests as an important feature, and for *syd2* feature 34 (mean group size) appears in both tests as important for loop closure detection. For both Sydney datasets, Test 1 suggests that feature 38 is best for loop closure detection. This feature corresponds to range histograms with bin size 0.5m. For *ken* and *syd1*, Test 2 suggests that feature 21 (range kurtosis excluding maximum ranges) is most informative. The difference in total error is quite small for the five best features (Table 3, Test 2), however the results do suggest that the best five features are different for the three data sets. As all three data sets were acquired in rather similar suburban environments, see Figure 3, this raises the important question of whether the presented loop closure detection method generalizes between data sets. This question is addressed further in Section 5.3.

Table 3: Most informative features for loop closure detection using 2D data. The feature numbers correspond to the list numbers in Appendix A. Test 1 shows which features were added in the first five training rounds. Test 2 shows the resulting validation error when the features were removed from the training data before learning the classifier.

Test 1					
Training Round	1	2	3	4	5
Added Feature, <i>ken</i>	4	44	22	12	19
Added Feature, <i>syd1</i>	38	23	17	40	4
Added Feature, <i>syd2</i>	38	22	34	7	35
Test 2					
Feature removed, <i>ken</i>	21	22	4	44	35
Total error [%]	2.98	2.94	2.92	2.83	2.82
Feature removed, <i>syd1</i>	21	10	17	23	27
Total error [%]	0.36	0.31	0.30	0.30	0.30
Feature removed, <i>syd2</i>	34	6	43	42	29
Total error [%]	0.38	0.36	0.36	0.35	0.34

Results for the 3D data are presented in Table 4. As can be seen, for AASS, the features that are added in early training rounds also have the largest negative effect when removed. Those features, numbers 33, 40, 32 and 41, correspond to range histograms with bin sizes 0.1, 2.5 and 3 m, respectively, and standard deviation of range difference for ranges shorter than or equal to $g_{r_3} = 0.5r_{\max}$. For *hann2*, the results are less consistent, however feature 35, corresponding to range histogram with bin size 0.5 m, appears to be most effective at separating the two classes of data pairs. As with the 2D data, the results from Test 2 suggest that two different sets of five features are best at detecting loop closure. Comparing to the 2D results in Table 3, the difference in total error is larger for the 3D data.

Furthermore, considering all results in Tables 3 and 4 together shows that the most important features for loop closure detection are not the same for either the 2D data sets or the 3D data sets. As mentioned above, an important and immediate question raised by this is whether or not the method is able to generalize between environments (i.e. between data sets). In e.g. 3D, *hann2* is an outdoor data set and AASS is an indoor data set, suggesting that the learned classifier might not generalize from outdoor to indoor. Again, this issue is addressed further in Section 5.3, where it is shown that the classifier does in fact generalize well between different environments and different sensor setups.

Classifier performance

In this section we present the performance of the classifiers in terms of D and FA , as defined above. Figure 5 shows ROC curves for the classifier in 2D and 3D.

Table 4: Most informative features for loop closure detection using 3D data. The feature numbers correspond to the list numbers in Appendix A. Test 1 shows which features were added in the first five training rounds. Test 2 shows the resulting validation error when the features were removed from the training data before learning the classifier.

Test 1					
Training Round	1	2	3	4	5
Added Feature, <i>hann2</i>	35	1	7	27	20
Added Feature, AASS	33	40	32	36	41
Test 2					
Feature removed, <i>hann2</i>	21	8	10	28	35
Total error [%]	1.29	1.15	1.14	1.13	1.13
Feature removed, AASS	41	22	33	32	40
Total error [%]	2.27	2.24	2.16	2.08	2.04

Good levels of detection are achieved for all levels of false alarm for the data used here. Table 5 shows a summary of the results achieved compared to related work. For the 3D results and the 2D *ken* results, the same data sets were used in the experiments. The two Sydney data sets have not been used in any previous work.

For the 2D classifier, the performance for different translational distances can be compared for all three data sets. In general, performance degrades with increasing translational distance, however even at 3m distance detection rates are sufficient for reliable SLAM.

As seen in Table 5 and Figure 5, for *syd1* the classifier characteristics are worse for 1m distance than for 2m distance. Further, 0% was the lowest D for 0% FA for AASS. This occurred in 5 out of 10000 cross validations. Furthermore, the mean D is lower than (Magnusson et al., 2009) and the standard deviation of D is higher than for *hann2*. For these two data sets, the number of training data pairs is small and unbalanced (31+31 and 16+324, respectively), which is an intuitive reason for the worse performance. The training data is crucial to the AdaBoost learning, and it is possible that there is not enough data pairs to be able to achieve a high degree of class separation.

To test this hypothesis, 16 positive and 300 negative data pairs were randomly selected from the large set of *hann2* data pairs, and a classifier was learned and evaluated from the subset of data. Out of 1000 such random subsets, 30 resulted in classifiers with 0% D for 0% FA (mean D was $72\% \pm 19\%$ for 0% FA). While this result is not sufficient to conclude that the small number of positive data pairs is the sole reason for the worse results for AASS, compared to related work and *hann2*, it does support the hypothesis that the relatively low number of positive training data has a strong negative effect on the learned classifiers ability to achieve a good degree of class separation. The ROC-curve corresponding to this

Table 5: Classification performance, all numbers in %. Results for the presented algorithm are shown in the middle column, results for related work are shown in the right column. The 3D results, and the 2D ken results, are for the same data sets as were used in related work. However, the 2D syd1 and syd2 data sets have not been used in any related work. Note that (Bosse and Zlot, 2008) detected loops between submaps consisting of laser range scans from tens of meters of travel, while our results are for loop detection between individual laser range scans.

Data set	FA	D	Min/Max	D	Source
hann2	0	63 ± 6	28/76	47	(Magnusson et al., 2009)
	1	99 ± 0.1	98/99	81	(Magnusson et al., 2009)
AASS	0	53 ± 14	0/88	70	"
	1	78 ± 6	56/88	63	"
ken, 1m	0	59 ± 11	18/84	N/A	(Bosse and Zlot, 2008)
	1	93 ± 1	90/99	51	"
ken, 2m	0	45 ± 13	3/72.00	N/A	"
	1	90 ± 1	86/92	51	"
ken, 3m	0	30 ± 10	3/60	N/A	"
	1	84 ± 1	80/87	51	"
syd1, 1m	0	57 ± 23	0/93		N/A
	1	57 ± 23	0/93		"
syd1, 2m	0	33 ± 9	4/62		"
	1	62 ± 4	45/75		"
syd1, 3m	0	19 ± 7	2/35		"
	1	63 ± 2	61/68		"
syd2, 1m	0	66 ± 9	27/84		"
	1	81 ± 5	61/91		"
syd2, 2m	0	35 ± 9	8/57		"
	1	81 ± 2	75/85		"
syd2, 3m	0	19 ± 8	2/38		"
	1	63 ± 2	56/68		"

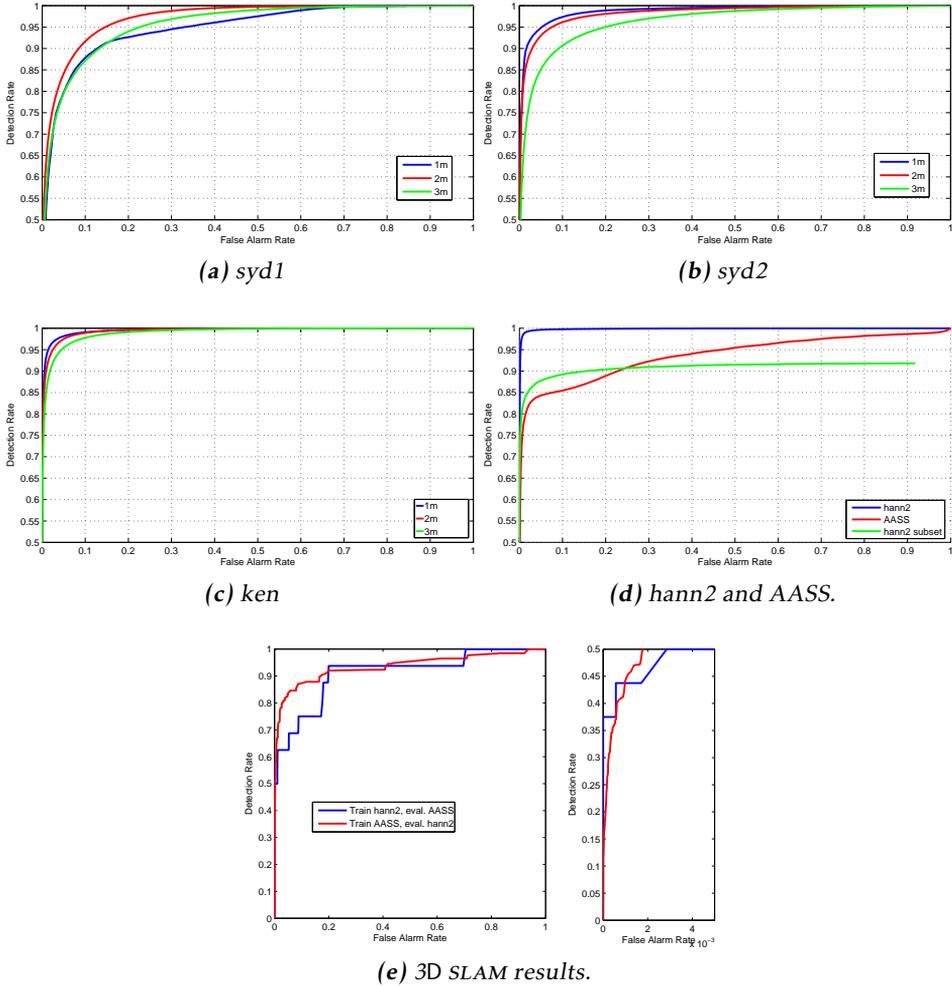


Figure 5: Receiver operating characteristic (ROC) curves. For each level of false alarm, the higher the detection is, the better the classifier is.

test is the green curve in Figure 5d. If compared to the curve for the full *hann2* data set, it shows a clear negative effect.

Comparison of 2D and 3D performance

This section presents a quantitative comparison of the performance of the classifier in 2D and 3D. Intuitively, performance in the 3D case should be considerably better than in 2D, since the added dimension and larger quantity of points significantly increases the information content of the point cloud. To obtain 2D data which is comparable to 3D data, 2D point clouds were extracted from the 3D data set *hann2* by taking all points which were located $1\text{m} \pm 15\text{cm}$ above the ground

Table 6: Comparison of 2D and 3D performance, all numbers in % unless otherwise stated.

Data set	FA	D	Min/Max
<i>hann2</i> , 2D	0	2.12 ± 1.40	0/8.34
	1	43.96 ± 2.81	36.36/55.08
<i>hann2</i> , 3D	0	63 ± 6	28/76
	1	99 ± 0.1	98/99
<i>syd2</i> , 3m, $\delta_\varphi = 1^\circ$	0	12.33 ± 6.21	N/A
<i>syd2</i> , 3m, $\delta_\varphi = 0.5^\circ$	0	18.57 ± 7.56	N/A

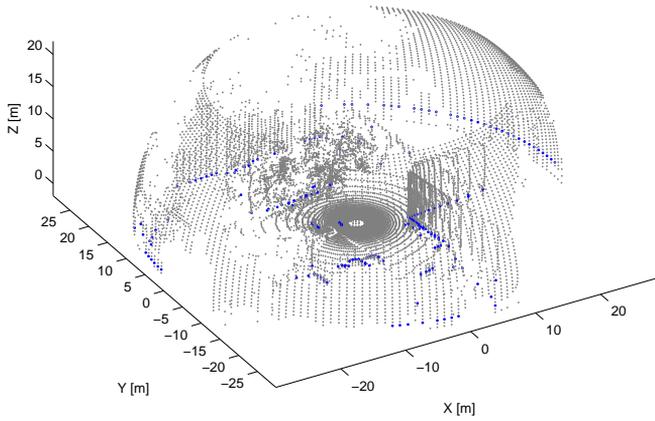
plane. The z-components were removed, i.e. the points were projected into the 1m-above-ground plane. The process is illustrated in Figure 6a and Figure 6b.

After extracting the 2D point clouds, a classifier was learned and evaluated using the same data pairs as when the 3D data was used. The results are shown in Figure 6c and Table 6. As expected the performance is worsened due to the lower information content in the point clouds. It is difficult to elaborate why the performance is so much worse, in comparison to the 2D results presented in Table 5 and Figure 5. A possible explanation is that for the *hann2* 2D point clouds the horizontal angular resolution is $\delta_\varphi = 1.0^\circ \pm 0.33^\circ$ (mean \pm one standard deviation), compared to the *syd1* and *syd2* data sets which have a horizontal angular resolution of $\delta_\varphi = 0.5^\circ$. Thus, the *hann2* 2D data contains less information, about 400 ± 163 points (mean \pm one standard deviation), compared to *syd1* and *syd2* which both contain 722 points per cloud.

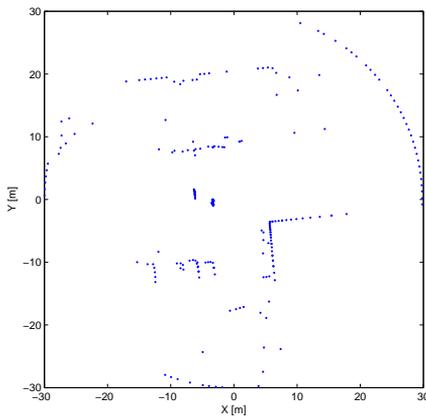
To test the hypothesis that the lower angular resolution is detrimental to classifier performance, the 3m data pairs from *syd2* were converted to $\delta_\varphi = 1^\circ$ data by taking every second range measurement. Cross validation experiments similar to the ones presented in Table 5 were performed, and the results presented in Table 6 show that decreasing the angular resolution has a negative effect on performance. This experiment does not rule out other explanations, e.g. different degrees of structure or self-similarity in the two environments, however it does support the hypothesis that the lower angular resolution contributes to some degree to the poor performance. Similar experiments were performed evaluating the different maximum measurable ranges, $r_{\max} = 30\text{m}$ for *hann2* and $r_{\max} = 50\text{m}$ for *syd2*, however the results showed no statistically significant difference in performance.

Dependence to translation

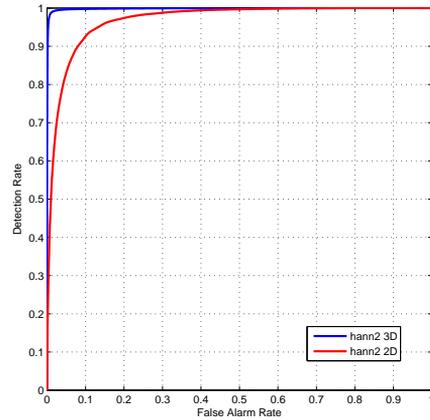
This section presents results from experiments testing how the learned classifier handles translation between the point clouds. While invariance to rotation is explicitly built into the features, and thus also into the learned classifier, there is no translational counterpart to the rotation invariance. For this experiment, data from the densely sampled *ken* data set was used. From the data set, a 78m trajectory which traverses a roundabout was chosen. Along the trajectory 1000 point clouds were acquired, with a mean translation of just 8cm between consecutive



(a) Original 3D point cloud in gray, with the extracted 2D point cloud in blue.



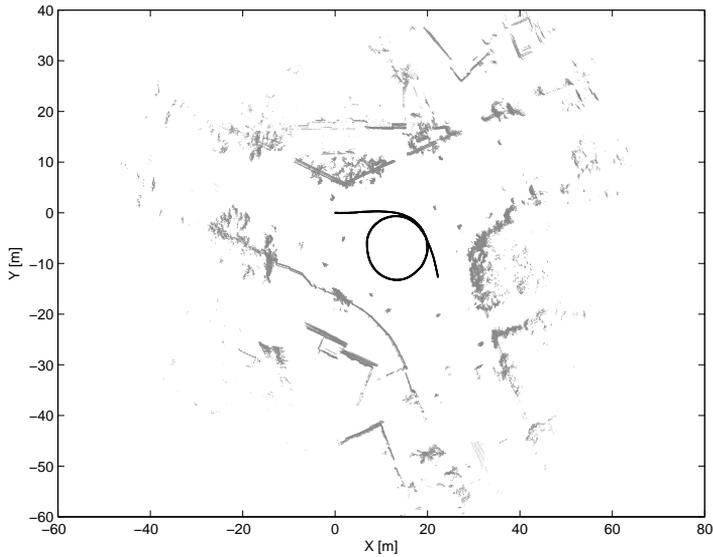
(b) Extracted 2D point cloud.



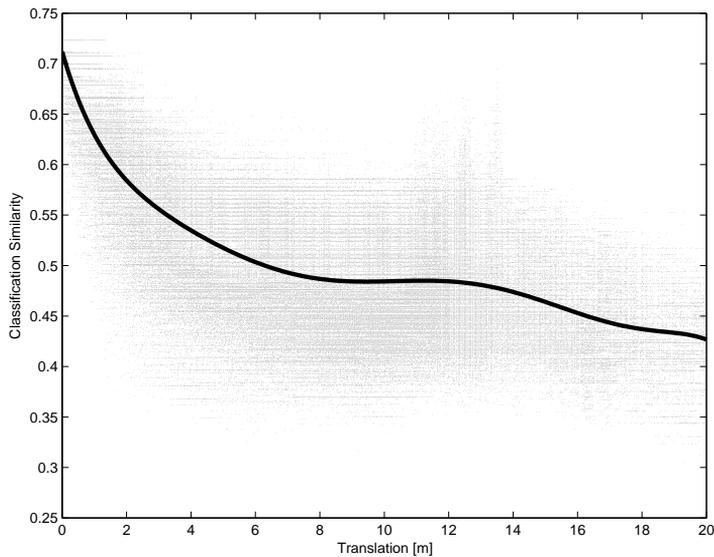
(c) Resulting receiver operating characteristic.

Figure 6: Illustration of the extraction of 2D point clouds from 3D point clouds.

point clouds. The resulting trajectory and area map are shown in Figure 7a. Each of the 1000 point clouds was compared to the remaining 999 point clouds, and the resulting classification similarity is plotted against translation in Figure 7b. The figure also features a polynomial fitted to the results in the least squares sense. As can be seen, there is a rather rapid decay in classification similarity as the translation increases, suggesting that the classifier is highly sensitive to translation. The explanation is, however, not so simple. As shown in Figure 8 the point



(a) Map of the area for which translation dependence were experimented on. The poses are printed in black, the point clouds are printed in gray.



(b) Plot of translation against classification similarity. The black line shows a seventh order polynomial fitted to the points in the least squares sense.

Figure 7: Classifier dependence to translation.

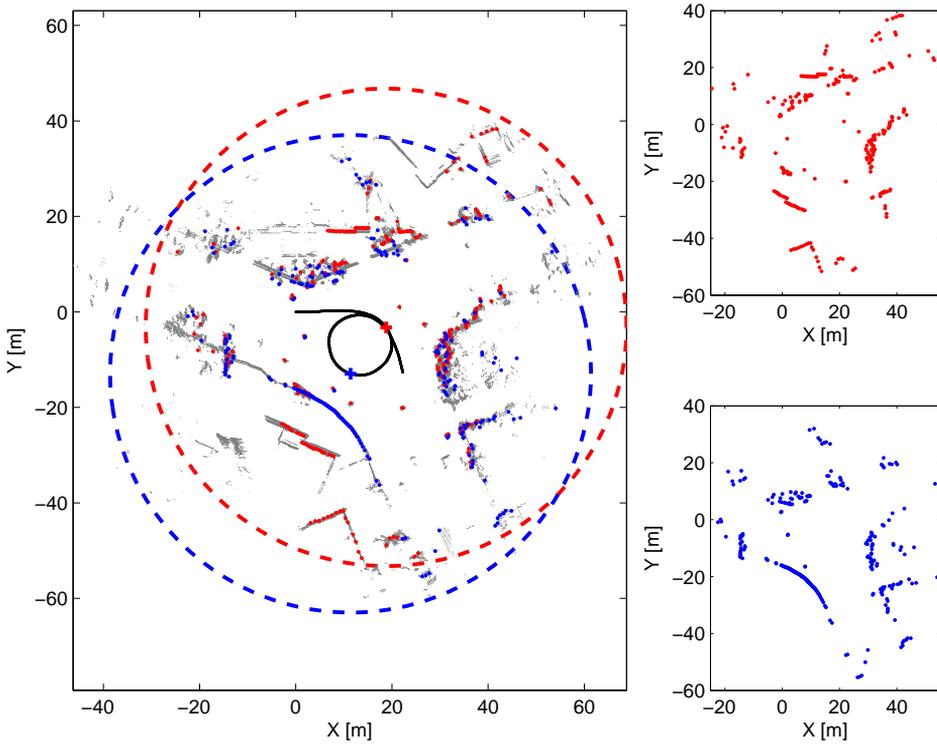


Figure 8: Illustration of the translation problem, for 2D data in outdoor environments. The plot on the left shows a trajectory, black line, along which 1000 point clouds were acquired, the constructed area map is shown in gray. Two poses are marked by crosses in blue and red, respectively, and the corresponding sensor surveillance boundary is shown by the dashed circles. Despite the fact that the relative pose translation, 12m, is well below the sensor’s maximum range, 50m, the point clouds corresponding to the poses are significantly different.

clouds change significantly in appearance after a small translation, thus making it difficult to determine that the point clouds are acquired in close vicinity of each other. Rather than showing the learned classifier’s dependence to translation, this experiment shows the inherent difficulty of sensing a 3D environment with a 2D sensor.

However, further results which show the dependence to translation are found in Table 5 and Figure 5. While the detection rate decreases with increasing translational distance, it is still possible to achieve good loop closure detection results for up to 3m distance, in both 2D and 3D. The decreasing detection rates do imply an inversely proportional dependence between detection and translational distance. This suggests that the presented method is more suitable for environments where

the vehicle is expected to travel along defined paths, e.g. office hallways, urban or rural roads, etc. In an environment where the vehicle is expected to be less restrained to defined paths, the presented method would possibly perform worse. To summarize, the presented method can handle translation, however considerable overlap of sensor field of view appears to be needed.

Dynamic objects

A challenge in loop closure detection is the presence of dynamic objects in the environment. Dynamic objects change the appearance of a scene, making it more difficult to recognize that two point clouds are indeed from the same location. An example of the challenge that dynamic objects present was given in Figure 1a, where the robot returns to a place where two vehicles have been parked along the side of the street.

In this section, we present results from experiments where the classifiers sensitivity to dynamic objects are tested. From the 2D *syd2* data set we were able to obtain 287 pairs of point clouds from the same location where dynamic objects have changed the appearance of the scene. In order to isolate the challenge of dynamic objects from other factors which may also affect the loop closure classification, the pairs of point clouds that are tested in this experiment are acquired at very low translational distance. Thus, the differences between the point clouds can be said to fully be an effect of the dynamic objects.

In order to assess the point cloud difference, which can be compared to the classification likelihood, we have computed the root mean square of the normalized extracted features for each pair of point clouds,

$$\sqrt{\frac{1}{n_f^1 + n_f^2} \sum_j \left(\frac{\mathbf{F}_{k,l}^{(j)}}{\mathbf{F}_\mu^{(j)}} \right)^2}. \quad (24)$$

The extracted features $\mathbf{F}_{k,l}$ were normalized with the average extracted feature \mathbf{F}_μ , since the extracted features are quite different in magnitude. The normalization was thus performed to give each component of the extracted feature vector an approximately equal weight. To compute \mathbf{F}_μ , the positive class data pairs from the same data set were used. We chose to use the pairs from the same data set to assess the average similarity for the particular environment. A simple relation that can be used to better understand (24) is that if $\mathbf{F}_{k,l} = k\mathbf{F}_\mu$, the point cloud difference is k .

Results from the experiment are shown in Figure 9. The plot of classification likelihood against feature difference, Figure 9a, does not show any clear trend, in contrast to, e.g., Figure 7b which shows a downward trend in classification likelihood as translational distance increases. While the likelihood for some point cloud pairs is rather low, around 0.5, at the highest feature difference computed, around 5.25, the likelihood of loop closure is high for several of the point cloud pairs. It appears that the classifier can handle dynamic objects in many cases, which is in accordance with empirical impressions from using the classifier in

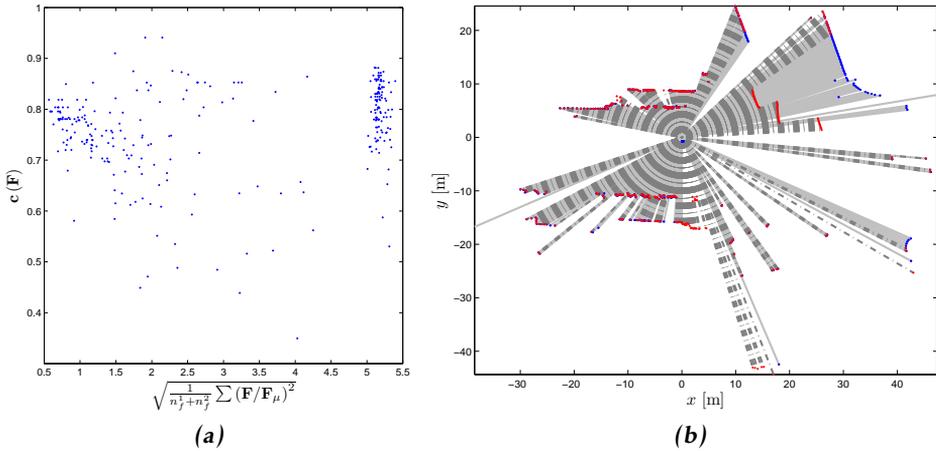
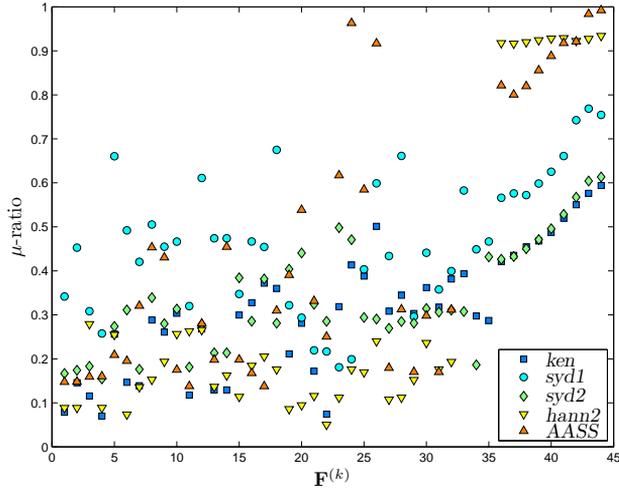


Figure 9: Results from experiment with dynamic objects. (a): Comparison of classifier likelihood and feature difference, computed as (24). Despite large feature differences, reasonable classification likelihood can still be achieved in many cases. (b): Example of point cloud pair, where the appearance of the scene is changed by dynamic objects. The first point cloud is shown as blue dots, with the measurement rays shown in light gray. The other point cloud is shown as red dots, with the measurement rays shown in dash-dotted dark gray. The feature difference is 5.31, the classification likelihood, 0.53 is quite low. Note that range measurements at maximum range are not plotted for increased clarity.

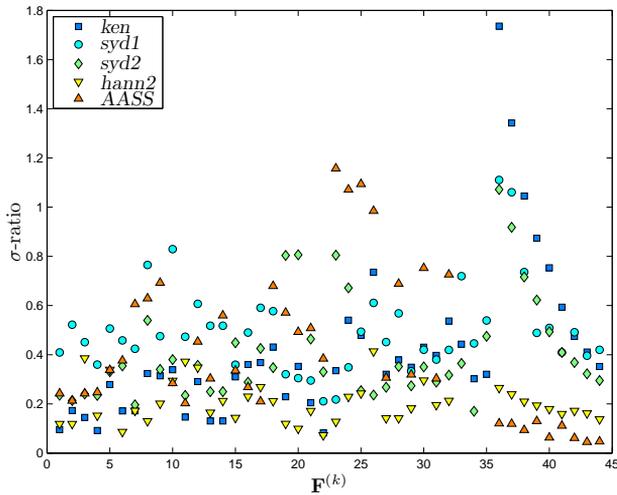
SLAM experiments. If the positive training data includes point cloud pairs with dynamic objects, e.g. cars and humans, then the learned classifier can handle dynamic objects in the test data to some extent.

Repetitive structures

Another difficulty faced by loop closure detection methods is the presence of repetitive structures in the environment. In e.g. an office environment, many hallways look similar, and many doorways also look similar. A high degree of repetitiveness in the environment is thus difficult, since the appearance of many places will be similar and consequently the computed feature values will be similar. While performing the experiments presented previously, repetitive structures in the data sets did not appear to pose a major difficulty to the presented loop closure detection method. To test our empirical observation that repetitive structures were not an issue in the data sets used, we considered the sample mean and standard deviation of the extracted features. If a feature is repetitive, many point clouds will measure the same feature value, and the feature difference will thus be similar for both point cloud pairs that are from the same location, and for point cloud pairs that are not from the same location.



(a)



(b)

Figure 10: Results from repetitive structures experiment. The plots show the ratio for the positive and negative data pairs of mean, in (a), and standard deviation, in (b), of the extracted features. Ratios close to, or larger than one, suggest a high degree of repetitive structures.

Let $\mathbf{F}_i^{(k)}$ be component k of the extracted features for data pair i , \mathbf{F}_{i_1, i_2} . Further, let μ_k^p and μ_k^n denote the mean and let σ_k^p and σ_k^n denote the standard deviation of \mathbf{F}_i^k for the positive and negative data pairs, respectively. For features of the first type, the mean should be small for the positive class, and larger for the negative

class. Features of the second type should be close to one for the positive class, and smaller for the negative class. If a feature is repetitive, μ_k^p and μ_k^n will be of similar size. Thus, the ratio

$$\begin{cases} \mu_k^p/\mu_k^n & \text{for type 1 features} \\ \mu_k^n/\mu_k^p & \text{for type 2 features} \end{cases} \quad (25)$$

can be used as a measure of repetitiveness. A value closer to one means that the environment is repetitive with respect to that feature.

For each of the positive data pairs, $F_i^{(k)}$ should be low for the first feature type, or close to one for the second feature type, and σ_k^p should thus be small. The negative pairs are random samples of point clouds, thus σ_k^n will be small if the feature is repetitive, and otherwise larger. Thus, the ratio

$$\frac{\sigma_k^p}{\sigma_k^n} \quad (26)$$

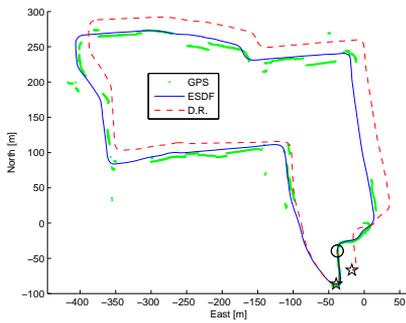
can also be used as a measure of repetitiveness. Analogously to (25), a value closer to one means that the environment is repetitive with respect to that feature. Using the 1m 2D data pairs and both sets 3D data pairs, the ratios (25) and (26) were computed. The results are shown in Figure 10.

For AASS, features 24 and 26 both have μ and σ ratios that suggest a high degree of repetitiveness. Considering the small size of this data set, it is difficult to draw any definite conclusions though. Regarding remaining data sets, for the mean ratio (25), in general none of the features appear to suffer from repetitiveness. The range histograms for the 3D data are quite close though. For the standard deviation ratio (26), the features corresponding to range histograms with smaller bins appear to be somewhat sensitive to repetitiveness in 2D. Both the μ and σ ratios are below 0.5 in about 80% of the cases. To summarize, the results from the experiment largely support our observation that repetitive structures in the environment is not a major problem in the data sets used in this paper.

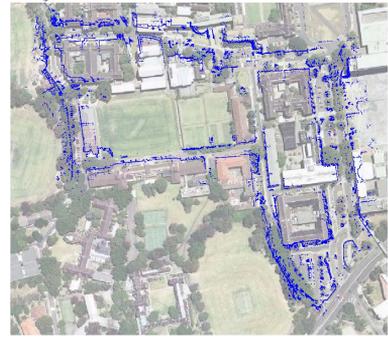
However, in a very large scale data set from a highly repetitive environment, repetitiveness could possibly become a problem. One way to handle such a difficulty is to modify the main SLAM filter to handle multiple hypotheses, similarly to the multiple hypothesis filter for target tracking, see e.g. (Bar-Shalom and Rong Li, 1995). This way, ambiguous loop detections could be kept as separate hypotheses until one or more hypotheses could be rejected.

5.3 SLAM experiments

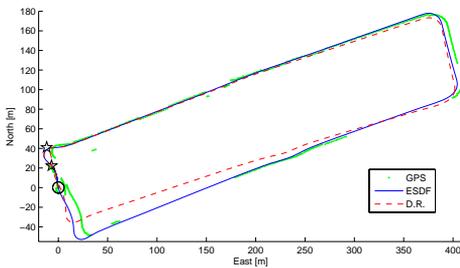
In this section we present SLAM experiments in both 2D and 3D, using the framework presented in Section 4. These experiments were conducted for two reasons, the first is to verify how the classifier would perform in a SLAM setting, the other is to verify how the classifier performs when it is trained on data from one environment and then tested on data from another. Thus, in each experiment, the classifier was trained on one dataset and then used to detect loop closure on an-



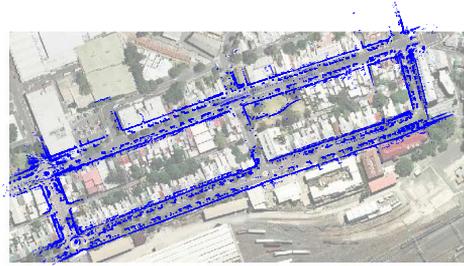
(a) Estimated trajectory compared to dead reckoning (D.R.) and GPS.



(b) Resulting SLAM map overlaid onto an aerial photograph.



(c) Estimated trajectory compared to dead reckoning (D.R.) and GPS.



(d) Resulting SLAM map overlaid onto an aerial photograph.

Figure 11: 2D SLAM results. In (a) and (c), the rings mark the starting points and the stars mark the respective end points of the estimated trajectory and dead reckoning.

other dataset. For the 2D results, both training and testing were performed with outdoor data. For the 3D experiments, both outdoor and indoor data were used, and thus we are able to demonstrate how the classifier generalizes from one environment to another.

2D SLAM

The data pairs from *syd2* were used to train a classifier, which was then used to detect loop closure in experiments with the datasets *syd1* and *syd3*. Figure 11 shows the estimated ESDF trajectories compared to dead reckoning and GPS, and also the resulting point cloud maps overlaid onto aerial photographs. The results show a clear improvement in trajectory estimation when the suggested loop closure detection classifier was used.

3D SLAM

In the first experiment, the positive and negative data pairs from *hann2* were used to train a classifier. The classifier was then used to classify data pairs from the AASS dataset. Each point cloud \mathbf{p}_k was compared to all previous point clouds, $\{\mathbf{p}_i\}_{i=1}^{k-1}$. The result from the experiment is shown as a classification matrix in Figure 12a. The k, l :th element of the classification matrix is the classification likelihood $\mathbf{c}(\mathbf{F}_{k,l})$, (12) in Algorithm 3. For completeness, the classification matrix in Figure 12a contains the classification of the (k, k) point cloud pairs. In a SLAM experiment however, such tests are obviously redundant. Figure 12b shows the classification matrix after thresholding each element, (13) in Algorithm 3. Black squares correspond to pairs of point clouds classified as being from the same location. Figure 12c shows the corresponding ground truth: black squares correspond to pairs of point clouds acquired less than 1m apart (Magnusson et al., 2009).

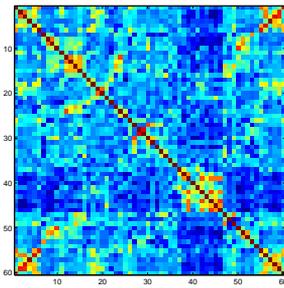
There is a high similarity between Figures 12b and 12c, showing the generalization properties of the features and the classifier. The classifier used in the experiment was trained on outdoor data containing 17000 points per cloud, $r_{\max} = 30$, and then tested on indoor data containing 112000 points per cloud, $r_{\max} = 15$. Figure 12e shows a 2D projection of the resulting map from the SLAM experiment, with the robot trajectory overlaid. The robot trajectory is compared to dead reckoning in Figure 12d. For this part of the experiment, a minimum loop size of five poses was introduced, explaining why the detected loop closure between poses 28 and 29 in Figure 12b is not present in Figure 12e.

In the second experiment, the AASS data was used to train a classifier, which was then used to classify the *hann2* data. The classification results from this experiment are presented in Figure 13. For this experiment the detection rate is just 3% for 0% false alarm, an intuitive explanation for the poor performance is again the small number of training data, $N_p = 16$ and $N_n = 324$. It could be noted though that even such low detection rates can be enough to produce good SLAM maps (Cummins and Newman, 2009). The ROC-curves for both the 3D SLAM experiments are shown in Figure 5e. Both ROC-curves show good detection rates for false alarm rates $\geq 1\%$. The SLAM experiment where the AASS data was used for training does not handle very low levels of false alarm though, which the right plot in Figure 5e shows.

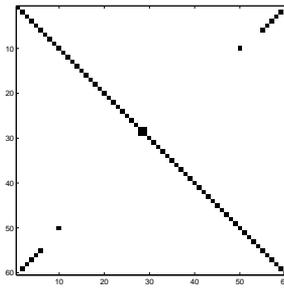
5.4 Summary and comparison

In this last subsection, we summarize the results from the experiments, and discuss how the presented loop closure detection method compares to related work. In experiments, we have presented results that

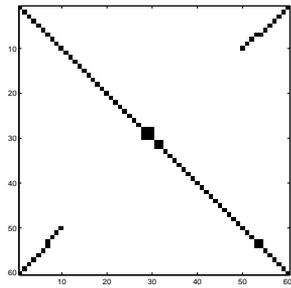
1. show the classifiers' execution time,
2. show the number of weak classifiers needed to construct a strong classifier,
3. test which features are most informative,



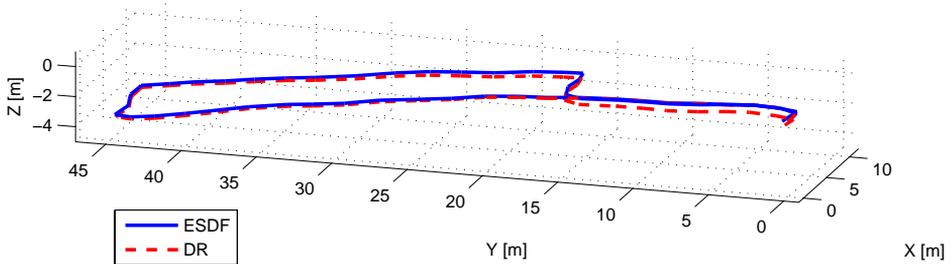
(a) Classification matrix, warmer color means higher similarity.



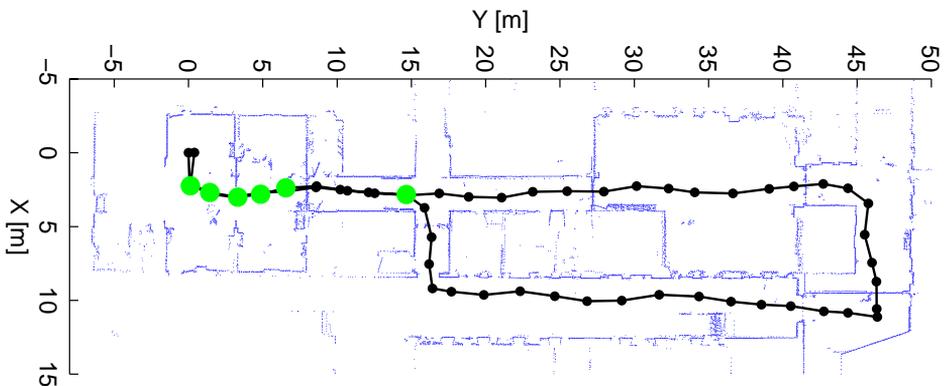
(b) Thresholded classification matrix, $K = 0.717$ (cf. (13) in Algorithm 3).



(c) Ground truth distance matrix, showing data pairs less than 1m apart (Magnusson et al., 2009).

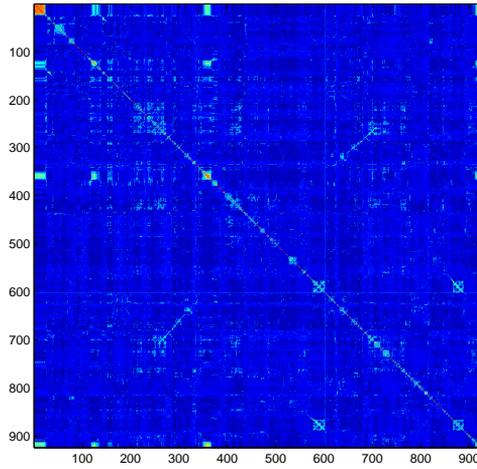


(d) Estimated trajectory (ESDF) vs. dead reckoning (DR).

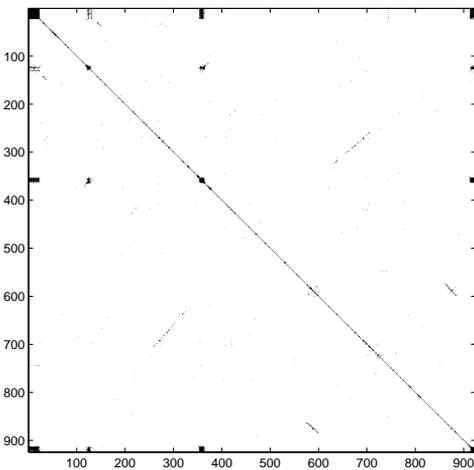


(e) Resulting map with robot trajectory overlaid. Detected loop closures are marked with thick green circles.

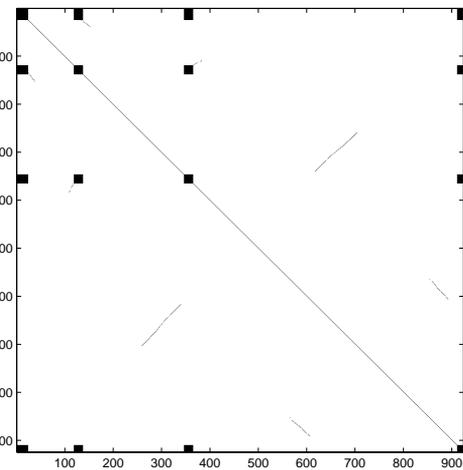
Figure 12: Results from SLAM experiment on AASS data.



(a) Classification matrix, warmer color means higher similarity.



(b) Thresholded classification matrix, $K = 0.35$ (cf. (13) in Algorithm 3).



(c) Point cloud pairs from same location used for training.

Figure 13: Results from SLAM experiment on hann2 data.

4. evaluate the classifier's receiver operating characteristic,
5. compares performance in 2D and 3D,
6. evaluate the dependence to translation,
7. show how the classifier handles dynamic objects,
8. test how the classifier handles repetitive structures in the environment, and

9. show that the classifier can be used to detect loops in SLAM experiments, both in 2D and 3D.

The presented loop detection method was evaluated on the publicly available *ken* data set, used in previous work on the same problem (Bosse and Zlot, 2008). It should be noted however that the work by (Bosse and Zlot, 2008) is for submaps containing laser scans from tens of meters of travel, while our results are for single laser scans. Further, as mentioned in Section 5.1, our results are for the first half of the data set. Thus, while the presented results are for the same data set, a *ceteris paribus* comparison is not possible. Even so, 84% detection at 1% false alarm is significantly higher than 51% detection at 1% false alarm, thus the presented work can be claimed to outperform related work on 2D data by (Bosse and Zlot, 2008) in terms of achieving high detection rates at low false alarm rates. In the work by (Brunskill et al., 2007), detection rates are not reported at low false alarm rates (or conversely, at high precision rates), and the data sets used are smaller in scale. A thorough comparison of quantitative results is thus unfortunately not possible.

For *hann2* 3D data, detection rates are higher than the NDT work by (Magnusson et al., 2009), while the work by (Steder et al., 2010) outperforms the presented method. It could be noted here that the method presented by (Steder et al., 2010) includes registering the two point clouds, which is not included in the detection here. Registering the two point clouds allows the robot to evaluate the registration result, and weed out possible false alarms as point clouds that are poorly aligned. It is possible that the presented method's detection rates could be improved further if the method was coupled with a registration process.

The presented method is faster than (Steder et al., 2010) however and, which is noted above and in related work, detecting only a subset of loops is typically sufficient to produce good SLAM results. Low execution times are of high importance, especially for larger data sets, since each point cloud must be compared to all previous point clouds. Furthermore, the presented method is fully invariant to rotation, while the work by (Steder et al., 2010) relies on the assumption that the robot is traveling over a flat surface, and the work by (Magnusson et al., 2009) relies on finding dominant planar surfaces.

The experiments that showed the most informative features showed that the results differed between different data sets, suggesting that the classifier might not generalize well between different data sets or environments. It was shown in Section 5.3 however, that the classifier does in fact generalize between different environments and sensor setups. This fact is important – since the classifier relies on being learned from manually labeled data, it must generalize well in order to function in an environment which is different from that on which it was learned. Experiments with dynamic objects showed that the type of dynamic objects that typically appear in suburban environments can be handled in most cases. Repetitive structures in the environment was shown to not pose a considerable challenge in the data sets used here.

The presented method is, compared to related work in both 2D and 3D, at a disadvantage in terms of the ability to handle translation. When the environment contains well defined pathways, such as office hallways or urban or rural roads, and the data is sampled without much translation in between point clouds, the sensitivity to translation is not a problem, which is shown by the SLAM experiments in 2D and 3D. Obtaining densely sampled data in 2D is easy using standard sensors, i.e. the SICK LMS200-sensors. In 3D, densely sampled data can be obtained using state-of-the-art sensors, i.e. the Velodyne HDL-64E. Thus, the need for densely sampled data does not pose a significant limitation. For data which is from environments without well defined pathways the dependence to translation could possibly prove to be problematic.

To summarize, the presented method performs well in environments with defined pathways, the execution times are favorable and detections rates at low false alarm rates compare well to related work and are sufficient to produce good SLAM results.

6 Conclusions and future work

This paper presented a method for loop closure detection, using pairwise comparison of point clouds. The presented method uses rotation invariant features, which provide a way to compress the sensed information into meaningful statistics. This reduces the dimension of the data by up to a factor of 2000 (point clouds with $> 100k$ points), thus the features also present a way to store the data in an efficient manner. The features are input to AdaBoost, which builds a classifier with good generalization properties. Inheriting the rotation invariance from the features, the learned classifier is fully invariant to rotation, without the need to discretize the metric space, assume that the robot is traveling over a flat surface, or be limited by predefined geometric primitives. Thus, it is possible to detect loop closure from arbitrary directions. Experiments in both 2D and 3D showed the algorithms ability to achieve levels of detection at 0% false alarm, at detection levels comparable to related work. The SLAM experiments presented showed that the method can perform reliable localization and mapping in very challenging environments. Experiments using both indoor and outdoor data showed the generalization properties of the framework proposed. The method is shown to be suitable for real-time performance: computing the set of features takes at most 0.2s (for a point cloud with 112000 points), and comparing the set of features for two point clouds takes less than 2ms.

In the experiments, the dependence between number of training data and classifier performance was noted. In future work, we intend to investigate this dependence further, and also address how training data can be selected in order to achieve the best performance at the lowest computational cost. Experiments showed that in addition to being fully invariant to rotation, the classifier can also handle up to 3m translation when detection loop closure between pairs of point clouds. In future work, we wish to evaluate whether this distance can be ex-

tended, such that the classifier can handle loop closure detection with less partial overlap between the point clouds. It would also be interesting to test the classifier on a very large scale data set from a highly repetitive environment, to see how it would perform in such a situation. Further, the presented SLAM framework relies on pairwise comparison between the current point cloud and all previous point clouds, resulting in a time complexity which grows linearly with the robot trajectory. The computed set of features can possibly be used in an initial nearest neighbor search, candidates from which are then used as input to the classifier. A similar approach has previously been taken for 2D point cloud submaps using keypoints and kd - and Bkd -trees (Zlot and Bosse, 2009).

A Appendix

A.1 Feature definitions

In this appendix we define the features that were used to learn classifiers for loop closure detection. The first subsection defines the features used in 2D, the second subsection presents the features used in 3D.

Given a point cloud \mathbf{p}_k , 15 or 14 parameters need to be specified for computing the features in 2D or 3D, respectively. The parameters are given in Table 7. Except for r_{\max} , all parameters are set manually. In order to find appropriate values, we have used empirical results. For the range histograms, instead of choosing just one bin size, we use 9 different bin sizes and leave it to the algorithm to find which corresponding features are informative.

Table 7: Parameters used when the features are computed. All parameters except for r_{\max} are set manually.

Parameter	Numerical value	Comment
r_{\max}	15m (AASS), 30m (hann2), 50m (all 2D data)	Maximum measurable range. Determined by sensor used.
g_{dist}	2.5m	
$g_{\text{min size}}$	3	Only used in 2D.
$g_{r_1}, g_{r_2}, g_{r_3}$	$r_{\max}, 0.75r_{\max}, 0.50r_{\max}$	
b_1, \dots, b_9	0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5 and 3 meters	Bin sizes for the range histograms.

2D features

The following features were used for loop closure detection in 2D. Features 1 to 35 are of type 1, features 36 to 44 are of type 2.

1. -2. **Area** Measures the area covered by a point cloud. Points whose range is greater than r_{\max} have their range set to r_{\max} . Each point is seen as the

center point of the base of an isosceles triangle. The height of the triangle is $h_i = r_i$, and the width of the base is $w_i = 2r_i \tan\left(\frac{\delta_\varphi}{2}\right)$, where δ_φ is the horizontal angular resolution of the sensor. The area of the triangle is $a_i = \frac{h_i w_i}{2} = r_i^2 \tan\left(\frac{\delta_\varphi}{2}\right)$. The area feature is computed as

$$a_{\max} = r_{\max}^2 \tan\left(\frac{\delta_\varphi}{2}\right), \quad (27a)$$

$$f_1 = \frac{1}{N a_{\max}} \sum_{i=1}^N r_i^2 \tan\left(\frac{\delta_\varphi}{2}\right) = \frac{1}{N} \sum_{i=1}^N \left(\frac{r_i}{r_{\max}}\right)^2. \quad (27b)$$

The area is normalized by dividing by the maximum measurable area. Note that the specific numerical value of δ_φ is not needed to compute the feature. f_2 is the area computed for all ranges $r_i < r_{\max}$.

3. - 4. **Average Range** Let the normalized range be $r_i^n = r_i/r_{\max}$. f_3 is the average r_i^n for ranges $r_i < r_{\max}$ and f_4 is the average r_i^n for all ranges.
5. - 6. **Standard Deviation of Range** f_5 is the standard deviation of r_i^n for ranges $r_i < r_{\max}$ and f_6 is the standard deviation of r_i^n for all ranges.
7. - 9. **Circularity** A circle is fitted to all points in the cloud in a least squares sense, which returns the center of the fitted circle p_c and the radius of the fitted circle r_c . f_7 is r_c/r_{\max} , f_8 is the residual sum of squares divided by $N r_c$,

$$f_8 = \frac{1}{N r_c} \sum_{i=1}^N (r_c - \|p_c - p_i\|)^2, \quad (28)$$

where $\|\cdot\|$ is the Euclidean norm. f_9 is $\frac{\|p_c\|}{r_{\max}}$.

10. - 12. **Centroid** Let \bar{p} be the mean position of the point cloud, computed for all points $r_i < r_{\max}$. $f_{10} = \|\bar{p}\|$, f_{11} is the mean distance from \bar{p} for points $r_i < r_{\max}$ and f_{12} is the standard deviation of the distances from \bar{p} for points $r_i < r_{\max}$.
13. - 14. **Maximum Range** f_{13} is the number of ranges $r_i = r_{\max}$ and f_{14} is the number of ranges $r_i < r_{\max}$.
15. - 17. **Distance** Let the distance between consecutive points be

$$\delta p_i = \|p_i - p_{i+1}\|. \quad (29)$$

f_{15} is the sum of δp_i for all points. f_{16} is the sum of δp_i , for consecutive points with $r_i, r_{i+1} < r_{\max}$. f_{17} is the sum of all $\delta p_i < g_{\text{dist}}$, for consecutive points with $r_i, r_{i+1} < r_{\max}$.

18. **Regularity** f_{18} is the standard deviation of δp_i , for consecutive points with $r_i, r_{i+1} < r_{\max}$.

19. - 20. **Curvature** Let A be the area covered by the triangle with corners in p_{i-1} , p_i and p_{i+1} , and let d_{i-1} , d_i and d_{i+1} be the pairwise point to point distances. The curvature at p_i is computed as $k_i = \frac{4A}{d_{i-1}d_i d_{i+1}}$. Curvature is computed for $p_i \in \mathbf{I}$, where $\mathbf{I} = \{p_i : r_{i-1}, r_i, r_{i+1} < r_{\max}, d_{i-1}, d_i, d_{i+1} < g_{\text{dist}}\}$. f_{19} is the mean curvature and f_{20} is the standard deviation of the curvatures.

21. - 22. **Range Kurtosis** Range kurtosis is a measure of the peakedness of the histogram of ranges. Sample kurtosis is computed for all points $r_i < r_{\max}$ as follows

$$m_k = \frac{1}{N_{r_i < r_{\max}}} \sum_{i : r_i < r_{\max}} (r_i - \bar{r})^k, \quad (30a)$$

$$f_{21} = \frac{m_4}{(m_2)^2} - 3, \quad (30b)$$

where \bar{r} is mean range, and $N_{r_i < r_{\max}}$ is the number of ranges $r_i < r_{\max}$. f_{22} is range kurtosis computed for all points in the cloud.

23. - 26. **Relative Range** Let the relative range be $r_i^r = r_i/r_{i+1}$. f_{23} is the mean of r_i^r and f_{24} is the standard deviation of r_i^r for all ranges. f_{25} and f_{26} are the mean and the standard deviation of r_i^r , respectively, computed for $r_i, r_{i+1} < r_{\max}$.

27. - 32. **Range Difference** Mean and standard deviation of range difference $r_i^d = |r_i - r_{i+1}|$. The features are calculated for all ranges less than or equal to a varying range gate g_r . g_{r_1} gives f_{27} (mean) and f_{28} (standard deviation), and g_{r_2} and g_{r_3} gives f_{29} to f_{32} . The features are normalized by division by the respective g_{r_i} .

33. - 34. **Group** A group is defined as a cluster of points in which the distance between consecutive points is less than a maximum distance gate g_{dist} . To be considered a group, the cluster has to contain more than a certain number of points specified by the minimum group size gate $g_{\text{min size}}$. f_{33} is the total number of groups found, f_{34} is the average number of points in each group.

35. **Mean Angular Difference** Measures the sum of the angles between consecutive point to point vectors. Given two consecutive points p_i and p_{i+1} , a vector that connects the points is given as $\bar{p}_{i,i+1} = [x_{i+1} - x_i, y_{i+1} - y_i]^T$. The feature is calculated as

$$f_{15} = \sum_{i: r_{[i,i+1,i+2]} < r_{\max}} \arccos \left(\frac{\bar{p}_{i,i+1}^T \bar{p}_{i+1,i+2}}{\|\bar{p}_{i,i+1}\| \|\bar{p}_{i+1,i+2}\|} \right). \quad (31)$$

36. - 44. **Range histogram** f_{33} to f_{41} are range histograms. Bins of sizes b_j , see Table 7, are used to tabulate the ranges.

3D features

The following features were used for loop closure detection in 3D. Features 1 to 32 are of type 1, features 33 to 41 are of type 2. Note that some of the 3D features are defined analogously to some of the 2D features, hence the definitions are not repeated.

1. - 2. **Volume** Measures the volume of the point cloud by adding the volumes of the individual laser measurements. Each point is seen as the center point of the base of a pyramid with its peak in the origin. Let δ_φ and δ_ψ be the laser range sensors horizontal and vertical angular resolution, and let $l_i = 2r_i \tan\left(\frac{\delta_\varphi}{2}\right)$ and $w_i = 2r_i \tan\left(\frac{\delta_\psi}{2}\right)$ be length and width of the pyramid base, and $h_i = r_i$ the height at point i . The volume of the pyramid is $v_i = \frac{l_i w_i h_i}{3}$. The volume is computed as

$$v_{\max} = \frac{4}{3} \tan\left(\frac{\delta_\varphi}{2}\right) \tan\left(\frac{\delta_\psi}{2}\right) r_{\max}^3 \quad (32a)$$

$$f_1 = \frac{1}{N v_{\max}} \sum_{i=1}^N v_i = \frac{1}{N} \sum_{i=1}^N \left(\frac{r_i}{r_{\max}}\right)^3 \quad (32b)$$

The volume is normalized by dividing by the maximum measurable volume $N v_{\max}$, i.e. the volume when all ranges equal r_{\max} . Notice that the explicit values of δ_φ and δ_ψ do not matter. f_2 is the volume computed using points with $r_i < r_{\max}$.

3. - 6. Defined analogously to features 3 to 6 in 2D.
7. - 9. **Sphere** A sphere is fitted to all points in the cloud in a least squares sense, which returns the center of the fitted sphere p_c and the radius of the fitted sphere r_c . f_7 is r_c/r_{\max} , f_8 is the residual sum of squares divided by $N r_c$,

$$f_8 = \frac{1}{N r_c} \sum_{i=1}^N (r_c - \|p_c - p_i\|)^2, \quad (33)$$

where $\|\cdot\|$ is the Euclidean norm. f_9 is $\frac{\|p_c\|}{r_{\max}}$.

10. - 32. Defined analogously to features 10 to 32 in 2D.
33. - 41. **Range histogram** Defined analogously to features 36 to 44 in 2D.

A.2 Compounding operations

This appendix contains definitions of the compounding operations \oplus and \ominus and their corresponding Jacobians. Let $\mathbf{x}_{i,j}$ denote the location of coordinate frame j with respect to coordinate frame i . The definitions are taken from (Eustice, 2005; Smith et al., 1990).

Compounding in 2D

Let the 2D 3-DOF pose be given by

$$\mathbf{x}_{i,j} = \begin{bmatrix} x_{i,j} & y_{i,j} & \psi_{i,j} \end{bmatrix}^T. \quad (34)$$

The compounding operation $\mathbf{x}_{i,k} = \mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k}$ is defined as

$$\mathbf{x}_{i,k} = \mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k} \quad (35a)$$

$$= \begin{bmatrix} x_{i,j} + x_{j,k} \cos(\psi_{i,j}) - y_{j,k} \sin(\psi_{i,j}) \\ y_{i,j} + x_{j,k} \sin(\psi_{i,j}) + y_{j,k} \cos(\psi_{i,j}) \\ \psi_{i,j} + \psi_{j,k} \end{bmatrix}. \quad (35b)$$

The Jacobian of the compounding operator \mathbf{J}_{\oplus} is given by

$$\mathbf{J}_{\oplus} = \frac{d(\mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k})}{d(\mathbf{x}_{i,j}, \mathbf{x}_{j,k})} = \frac{d(\mathbf{x}_{i,k})}{d(\mathbf{x}_{i,j}, \mathbf{x}_{j,k})} = \begin{bmatrix} \mathbf{J}_{1\oplus} & \mathbf{J}_{2\oplus} \end{bmatrix} \quad (36a)$$

$$= \begin{bmatrix} 1 & 0 & -(x_{j,k} \sin(\phi_{i,j}) + y_{j,k} \cos(\phi_{i,j})) & \cos(\phi_{i,j}) & -\sin(\phi_{i,j}) & 0 \\ 0 & 1 & x_{j,k} \cos(\phi_{i,j}) - y_{j,k} \sin(\phi_{i,j}) & \sin(\phi_{i,j}) & \cos(\phi_{i,j}) & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (36b)$$

where $\mathbf{J}_{1\oplus}$ and $\mathbf{J}_{2\oplus}$ correspond to the left and right 3×3 -matrix half partitioning of \mathbf{J}_{\oplus} . The inverse relationship \ominus , explaining $\mathbf{x}_{j,i}$ as a function of the coordinates in $\mathbf{x}_{i,j}$, is given by

$$\mathbf{x}_{j,i} = \ominus \mathbf{x}_{i,j} = \begin{bmatrix} -x_{i,j} \cos(\phi_{i,j}) - y_{i,j} \sin(\phi_{i,j}) \\ x_{i,j} \sin(\phi_{i,j}) - y_{i,j} \cos(\phi_{i,j}) \\ -\phi_{i,j} \end{bmatrix}, \quad (37)$$

with Jacobian \mathbf{J}_{\ominus}

$$\mathbf{J}_{\ominus} = \frac{d(\mathbf{x}_{j,i})}{d\mathbf{x}_{i,j}} = \frac{d(\ominus \mathbf{x}_{i,j})}{d\mathbf{x}_{i,j}} \quad (38a)$$

$$= \begin{bmatrix} -\cos(\phi_{i,j}) & -\sin(\phi_{i,j}) & x_{i,j} \sin(\phi_{i,j}) - y_{i,j} \cos(\phi_{i,j}) \\ \sin(\phi_{i,j}) & -\cos(\phi_{i,j}) & x_{i,j} \cos(\phi_{i,j}) + y_{i,j} \sin(\phi_{i,j}) \\ 0 & 0 & -1 \end{bmatrix}. \quad (38b)$$

Compounding in 3D

Let the 3D 6-DOF pose be given by

$$\mathbf{x}_{i,j} = \begin{bmatrix} x_{i,j} & y_{i,j} & z_{i,j} & \phi_{i,j} & \theta_{i,j} & \psi_{i,j} \end{bmatrix}^T. \quad (39)$$

The compounding operation $\mathbf{x}_{i,k} = \mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k}$ is defined as

$$\mathbf{x}_{i,k} = \mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k} \quad (40a)$$

$$= \begin{bmatrix} R_{i,j} \begin{bmatrix} x_{j,k} & y_{j,k} & z_{j,k} \end{bmatrix}^T + \begin{bmatrix} x_{i,j} & y_{i,j} & z_{i,j} \end{bmatrix}^T \\ \text{atan2} \left(R_{i,k}^{(1,3)} \sin \psi_{i,k} - R_{i,k}^{(2,3)} \cos \psi_{i,k}, -R_{i,k}^{(1,2)} \sin \psi_{i,k} + R_{i,k}^{(2,2)} \cos \psi_{i,k} \right) \\ \text{atan2} \left(-R_{i,k}^{(3,1)}, R_{i,k}^{(1,1)} \cos \psi_{i,k} + R_{i,k}^{(2,1)} \sin \psi_{i,k} \right) \\ \text{atan2} \left(R_{i,k}^{(2,1)}, R_{i,k}^{(1,1)} \right) \end{bmatrix}, \quad (40b)$$

where the rotation matrix $R_{i,j}$ is defined as

$$R_{i,j} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (41)$$

where the subscripts i,j are omitted for brevity. Further, $R_{i,k} = R_{i,j}R_{j,k}$ and $R_{i,k}^{(m,n)}$ is the (m,n) :th element of the rotation matrix $R_{i,k}$. The Jacobian of the compounding operator \mathbf{J}_{\oplus} is given by

$$\mathbf{J}_{\oplus} = \frac{d(\mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k})}{d(\mathbf{x}_{i,j}, \mathbf{x}_{j,k})} = \frac{d(\mathbf{x}_{i,k})}{d(\mathbf{x}_{i,j}, \mathbf{x}_{j,k})} = [\mathbf{J}_{1\oplus} \quad \mathbf{J}_{2\oplus}] \quad (42a)$$

$$= \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{M} & R_{i,j} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{K}_1 & \mathbf{0}_{3 \times 3} & \mathbf{K}_2 \end{bmatrix}. \quad (42b)$$

where $\mathbf{J}_{1\oplus}$ and $\mathbf{J}_{2\oplus}$ correspond to the left and right 6×6 -matrix half partitioning of \mathbf{J}_{\oplus} , and

$$\mathbf{M} = \begin{bmatrix} R_{i,j}^{(1,3)} y_{j,k} - R_{i,j}^{(1,2)} z_{j,k} & (z_{i,k} - z_{i,j}) \cos \psi_{i,j} & -(y_{i,k} - y_{i,j}) \\ R_{i,j}^{(2,3)} y_{j,k} - R_{i,j}^{(2,2)} z_{j,k} & (z_{i,k} - z_{i,j}) \sin \psi_{i,j} & (x_{i,k} - x_{i,j}) \\ R_{i,j}^{(3,3)} y_{j,k} - R_{i,j}^{(3,2)} z_{j,k} & -x_{j,k} \cos \theta_{i,j} - (y_{j,k} \sin \phi_{i,j} + z_{j,k} \cos \phi_{i,j}) \sin \theta_{i,j} & 0 \end{bmatrix}$$

$$\mathbf{K}_1 = \begin{bmatrix} \cos \theta_{i,j} \cos(\psi_{i,k} - \psi_{i,j}) \sec \theta_{i,k} & \sin(\psi_{i,k} - \psi_{i,j}) \sec \theta_{i,k} & 0 \\ -\cos \theta_{i,j} \sin(\psi_{i,k} - \psi_{i,j}) & \cos(\psi_{i,k} - \psi_{i,j}) & 0 \\ R_{j,k}^{(1,2)} \sin \phi_{i,k} + R_{j,k}^{(1,3)} \cos \phi_{i,k} \sec \theta_{i,k} & \sin(\psi_{i,k} - \psi_{i,j}) \tan \theta_{i,k} & 1 \end{bmatrix}$$

$$\mathbf{K}_2 = \begin{bmatrix} 1 & \sin(\phi_{i,k} - \phi_{j,k}) \tan \theta_{i,k} & (R_{i,j}^{(1,3)} \cos \psi_{i,k} + R_{i,j}^{(2,3)} \sin \psi_{i,k}) \sec \theta_{i,k} \\ 0 & \cos(\phi_{i,k} - \phi_{j,k}) & -\cos \theta_{j,k} \sin(\phi_{i,k} - \phi_{j,k}) \\ 0 & \sin(\phi_{i,k} - \phi_{j,k}) \sec \theta_{i,k} & \cos \theta_{j,k} \cos(\phi_{i,k} - \phi_{j,k}) \sec \theta_{i,k} \end{bmatrix} \quad (43)$$

The inverse relationship \ominus , explaining $\mathbf{x}_{j,i}$ as a function of the coordinates in $\mathbf{x}_{i,j}$, is given by

$$\mathbf{x}_{j,i} = \ominus \mathbf{x}_{i,j} = \begin{bmatrix} -R_{i,j}^T \begin{bmatrix} x_{i,j} & y_{i,j} & z_{i,j} \end{bmatrix}^T \\ \text{atan2} \left(R_{i,j}^{(3,1)} \sin \psi_{j,i} - R_{i,j}^{(3,2)} \cos \psi_{j,i}, -R_{i,j}^{(2,1)} \sin \psi_{j,i} + R_{i,j}^{(2,2)} \cos \psi_{j,i} \right) \\ \text{atan2} \left(-R_{i,j}^{(1,3)}, R_{i,j}^{(1,1)} \cos \psi_{j,i} + R_{i,j}^{(1,2)} \sin \psi_{j,i} \right) \\ \text{atan2} \left(R_{i,j}^{(1,2)}, R_{i,j}^{(1,1)} \right) \end{bmatrix} \quad (44)$$

with Jacobian \mathbf{J}_\ominus

$$\mathbf{J}_\ominus = \frac{d(\mathbf{x}_{j,i})}{d\mathbf{x}_{i,j}} = \frac{d(\ominus\mathbf{x}_{i,j})}{d\mathbf{x}_{i,j}} = \begin{bmatrix} -\mathbf{R}_{i,j}^T & \mathbf{N} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q} \end{bmatrix} \quad (45)$$

where

$$\mathbf{N} = \begin{bmatrix} 0 & -R_{i,j}^{(3,1)}(x_{i,j} \cos \psi_{i,j} + y_{i,j} \sin \psi_{i,j}) + z_{i,j} \cos \theta_{i,j} & R_{i,j}^{(2,1)}x_{i,j} - R_{i,j}^{(1,1)}y_{i,j} \\ z_{j,i} & -R_{i,j}^{(3,2)}(x_{i,j} \cos \psi_{i,j} + y_{i,j} \sin \psi_{i,j}) + z_{i,j} \sin \theta_{i,j} \sin \phi_{i,j} & R_{i,j}^{(2,2)}x_{i,j} - R_{i,j}^{(1,2)}y_{i,j} \\ -y_{j,i} & -R_{i,j}^{(3,3)}(x_{i,j} \cos \psi_{i,j} + y_{i,j} \sin \psi_{i,j}) + z_{i,j} \sin \theta_{i,j} \cos \phi_{i,j} & R_{i,j}^{(2,3)}x_{i,j} - R_{i,j}^{(1,3)}y_{i,j} \end{bmatrix}$$

$$\mathbf{Q} = \frac{1}{1 - (R_{i,j}^{(1,3)})^2} \begin{bmatrix} -R_{i,j}^{(1,1)} & -R_{i,j}^{(1,2)} \cos \phi_{i,j} & R_{i,j}^{(1,3)} R_{i,j}^{(3,3)} \\ R_{i,j}^{(1,2)} \sqrt{1 - (R_{i,j}^{(1,3)})^2} & -R_{i,j}^{(3,3)} \cos \phi_{i,j} \sqrt{1 - (R_{i,j}^{(1,3)})^2} & R_{i,j}^{(2,3)} \sqrt{1 - (R_{i,j}^{(1,3)})^2} \\ R_{i,j}^{(1,3)} R_{i,j}^{(1,1)} & -R_{i,j}^{(2,3)} \cos \psi_{i,j} & -R_{i,j}^{(3,3)} \end{bmatrix} \quad (46)$$

Composite relationships

Using the two operations defined above, operations for more than two spatial relationships can be performed. The following composite relationships hold in both 2D and 3D:

$$\mathbf{x}_{i,l} = \mathbf{x}_{i,j} \oplus \mathbf{x}_{j,l} = \mathbf{x}_{i,j} \oplus (\mathbf{x}_{j,k} \oplus \mathbf{x}_{k,l}) \quad (47a)$$

$$= \mathbf{x}_{i,k} \oplus \mathbf{x}_{k,l} = (\mathbf{x}_{i,j} \oplus \mathbf{x}_{j,k}) \oplus \mathbf{x}_{k,l} \quad (47b)$$

$$\mathbf{x}_{i,j} \ominus \mathbf{x}_{k,j} = \mathbf{x}_{i,j} \oplus (\ominus\mathbf{x}_{i,k}) \quad (47c)$$

$$\mathbf{x}_{j,k} = \ominus\mathbf{x}_{i,j} \oplus \mathbf{x}_{i,k} \quad (47d)$$

The Jacobian of (47d), $\ominus\mathbf{J}_\oplus$, is given by

$$\ominus\mathbf{J}_\oplus = \frac{d\mathbf{x}_{j,k}}{d(\mathbf{x}_{i,j}, \mathbf{x}_{i,k})} = \frac{d\mathbf{x}_{j,k}}{d(\mathbf{x}_{j,i}, \mathbf{x}_{i,k})} \times \frac{d(\mathbf{x}_{j,i}, \mathbf{x}_{i,k})}{d(\mathbf{x}_{i,j}, \mathbf{x}_{i,k})} = \mathbf{J}_\oplus \times \begin{bmatrix} \mathbf{J}_\ominus & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = [\mathbf{J}_{1\oplus} \mathbf{J}_\ominus \quad \mathbf{J}_{2\oplus}]. \quad (48)$$

Bibliography

- A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, October 2008.
- K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3402–3407, Roma, Italy, April 2007.
- Y. Bar-Shalom and X. Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.
- S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- P. Biber and W. Strasser. The normal distribution transform: A new approach to laser scan matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2743–2748, Las Vegas, USA, October 2003.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- M. C. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotics Research*, 27(6):667–691, June 2008.
- M. C. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2d lidar maps. *Robotics and Autonomous Systems*, 57(12):1211–1224, July 2009a.
- M. C. Bosse and R. Zlot. Place recognition using regional point descriptors for 3d mapping. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Cambridge, Massachusetts, USA, July 2009b.
- E. Brunskill, T. Kollar, and N. Roy. Topological mapping using spectral clustering and classification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3491–3496, San Diego, California, USA, October 2007.
- J. Callmer, K. Granström, J. I. Nieto, and F. T. Ramos. Tree of Words for Visual Loop Closure Detection in Urban SLAM. In *Proceedings of the Australian Conference on Robotics & Automation (ACRA)*, Canberra, Australia, December 2008.

- Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.
- M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *ijrr*, 27(6):647–665, June 2008.
- M. Cummins and P. Newman. Highly Scalable Appearance-Only SLAM – FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems (RSS)*, Seattle, WA, USA, June 2009.
- E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *Proceedings of the British Machine Vision Conference*, Leeds, United Kingdom, September 2008.
- R. M. Eustice. *Large-area visually augmented navigation for autonomous underwater vehicles*. PhD thesis, Massachusetts Institute of Technology / Woods Hole Oceanographic Institution Joint Program, 2005.
- R. M. Eustice, M. Walter, and J. J. Leonard. Sparse extended information filters: Insights into sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3281–3288, Alberta, Canada, August 2005.
- R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, December 2006.
- F. Fraundorfer, C. Engels, and D. Nister. Topological mapping, localization and navigation using image collections. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3872–3877, October 2007.
- Y. Freund and R. E. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of European Conference on Computational Learning Theory (EuroCOLT)*, pages 23–37, Barcelona, Spain, March 1995.
- A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Prague, Czech Republic, May 2004.
- T. Goedeme, T. Tuytelaars, and L. Van Gool. Visual topological map building in self-similar environments. In *Proceedings of International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Setubal, Portugal, August 2006.
- K. Granström and T. B. Schön. Learning to Close the Loop from 3D Point Clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, Taipei, Taiwan, October 2010.
- K. Granström, J. Callmer, F. T. Ramos, and J. I. Nieto. Learning to Detect Loop

- Closure from Range Data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, Kobe, Japan, May 2009.
- K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *The International Journal of Robotics Research*, 30(14):1728–1754, December 2011.
- J. S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, Monterey, CA, USA, November 1999.
- D. Hahnel, W. Burgard, D. Fox, and S. Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 206–211, Las Vegas, Nevada, USA, October 2003.
- K. Ho and P. Newman. Combining visual and spatial appearance for loop closure detection in slam. In *Proceedings of European Conference on Mobile Robots*, Ancona, Italy, September 2005.
- K. L. Ho and P. Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286, 2007.
- A. Howard and N. Roy. The robotics data set repository (radish), 2003. URL <http://radish.sourceforge.net/>.
- A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):433–449, May 1999.
- K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *ijrr*, 29(8):941–957, 2010.
- M. Magnusson, T. Duckett, and A. J. Lilienthal. Scan registration for autonomous mining vehicles using 3D-NDT. *jfr*, 24(10):803–827, October 2007.
- M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal. Automatic appearance-based loop detection from 3D laser data using the normal distributions transform. *jfr*, 26(11–12):892–914, November 2009.
- M. J. Milford and G. F. Wyeth. Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008.
- O. M. Mozos, C. Stachniss, and W. Burgard. Supervised Learning of Places from Range Data using AdaBoost. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1730–1735, Barcelona, Spain, April 2005.

- P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1050–4729, Orlando, FL, USA, May 2006.
- J. I. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching slam. *Robotics and Autonomous Systems*, 55(1):39–49, 2007.
- R. Paul and P. Newman. FAB-MAP 3D: Topological mapping with spatial and visual appearance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2649–2656, Anchorage, Alaska, USA, 2010.
- F. T. Ramos, D. Fox, and H. F. Durrant-Whyte. CRF-matching: Conditional random fields for feature-based scan matching. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, USA, June 2007a.
- F. T. Ramos, J. I. Nieto, and H. F. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1050–4729, Rome, Italy, April 2007b.
- W. D. Rencken, W. Feiten, and R. Zöllner. Relocalisation by Partial Map Matching. In *Sensor Based Intelligent Robots*, pages 21–35, 1999.
- M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *ijrr*, 28(5):595–599, May 2009.
- R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, 1990.
- B. Steder, G. Grisetti, and W. Burgard. Robust place recognition for 3D range data based on point features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1400–1405, Anchorage, Alaska, USA, May 2010.
- A. Tapus and R. Siegwart. A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1188–1193, Orlando, Florida, USA, 2006.
- S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. F. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *ijrr*, 23(7/8), 2004.
- P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- A. Walthelm. A new approach to global self-localization with laser range scans in unstructured environments. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 202–208, Versailles, France, 2002.

- O. Wulf. Robotic 3d scan repository, 2009. URL <http://kos.informatik.uni-osnabrueck.de/3Dscans/>. [Online; accessed 10-August-2009].
- Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.
- R. Zlot and M. C. Bosse. Place recognition using keypoint similarities in 2D lidar maps. In O. Khatib, V. Kumar, and G. Pappas, editors, *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 363–372. Springer Berlin / Heidelberg, 2009.

Paper B

Extended target tracking using a Gaussian mixture PHD filter

Authors: Karl Granström, Christian Lundquist, and Umut Orguner

Edited version of the paper:

K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.

An early version of this work was presented in:

K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.

Extended target tracking using a Gaussian mixture PHD filter

Karl Granström*, Christian Lundquist*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
{karl,lundquist}@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

This paper presents a Gaussian-mixture implementation of the PHD filter for tracking extended targets. The exact filter requires processing of all possible measurement set partitions, which is generally infeasible to implement. A method is proposed for limiting the number of considered partitions and possible alternatives are discussed. The implementation is used on simulated data and in experiments with real laser data, and the advantage of the filter is illustrated. Suitable remedies are given to handle spatially close targets and target occlusion.

1 Introduction

In most multi-target tracking applications it is assumed that each target produces at most one measurement per time step. This is true for the cases when the distance between the target and the sensor is large in comparison to the target's size. In other cases however, the target size may be such that multiple resolution cells of the sensor are occupied by the target. Targets that potentially give rise to more than one measurement per time step are categorized as extended. Examples include the cases when vehicles use radar sensors to track other road-users, when ground radar stations track airplanes which are sufficiently close to the sensor, or in mobile robotics when pedestrians are tracked using laser range sensors.

Gilholm and Salmond (2005) have presented an approach for tracking extended targets under the assumption that the number of received target measurements in each time step is Poisson distributed. Their algorithm was illustrated with two examples where point targets which may generate more than one measurement and objects that have a 1-D extension (infinitely thin stick of length l) are tracked. In (Gilholm et al., 2005) a measurement model was suggested which is an inhomogeneous Poisson point process. At each time step, a Poisson distributed random number of measurements are generated, distributed around the target. This

measurement model can be understood to imply that the extended target is sufficiently far away from the sensor for its measurements to resemble a cluster of points, rather than a geometrically structured ensemble. A similar approach is taken in (Boers et al., 2006) where track-before-detect theory is used to track a point target with a 1-D extent.

Baum and Hanebeck (2009) have presented the random hypersurface model, an extended target model which has been used to track elliptic targets (Baum et al., 2010b), as well as more general shapes (Baum and Hanebeck, 2011). A different approach to elliptic target modeling is the random matrix framework by Koch (Koch, 2008). The target kinematical states are modeled using a Gaussian distribution, while the ellipsoidal target extension is modeled using an inverse Wishart distribution. Using random matrices to track group targets under kinematical constraints is discussed in (Koch and Feldmann, 2009). Modifications and improvements to the Gaussian-inverse Wishart model of (Koch, 2008) have been suggested in (Feldmann et al., 2011), and the model (Koch, 2008) has been integrated into a Probabilistic Multi-Hypothesis Tracking (PMHT) framework in (Wieneke and Koch, 2010). A comparison of random matrices and the random hypersurface model under single target assumption is given in (Baum et al., 2010a). Measurements of target down-range extent are used to aid track retention in (Salmond and Parr, 2003). Other approaches to estimating the target extensions are given in (Granström et al., 2011; Lundquist et al., 2011; Zhu et al., 2011).

Using the rigorous finite set statistics (FISST), Mahler has pioneered the recent advances in the field of multiple target tracking with a set theoretic approach where the targets and measurements are treated as random finite sets (RFS). This type of approach allows the problem of estimating multiple targets in clutter and uncertain associations to be cast in a Bayesian filtering framework (Mahler, 2007b), which in turn results in an optimal multi-target Bayesian filter. As is the case in many nonlinear Bayesian estimation problems, the optimal multi-target Bayesian filter is infeasible to implement except for simple examples and an important contribution of FISST is to provide structured tools in the form of the statistical moments of a RFS. The first order moment of a RFS is called *probability hypothesis density* (PHD), and it is an intensity function defined over the state space of the targets. The so called PHD filter (Mahler, 2003, 2007b) propagates in time PHDs corresponding to the set of target states as an approximation of the optimal multi-target Bayesian filter. A practical implementation of the PHD filter is provided by approximating the PHDs with Gaussian-mixtures (GM) (Vo and Ma, 2006) which results in the Gaussian-mixture PHD (GM-PHD) filter. In the recent work (Mahler, 2009), Mahler presented an extension of the PHD filter to also handle extended targets of the type presented in (Gilholm et al., 2005).

In this paper, we present a Gaussian-mixture implementation of the PHD-filter for extended targets (Mahler, 2009), which we call the extended target GM-PHD-filter (ETGM-PHD). In this way, we, to some extent, give a practical extension of the series of work in (Gilholm et al., 2005; Vo and Ma, 2006; Mahler, 2009). An earlier version of this work was presented in (Granström et al., 2010) and the

current, significantly improved, version includes not only much more details and extensive investigations of the methodology but also practical examples with real data. For space considerations, we do not repeat the derivation of the PHD-filter equations for extended targets and instead refer the reader to (Mahler, 2009).

The document is outlined as follows. The multiple extended target tracking problem is defined in Section 2. The details of the Gaussian-mixture implementation are given in Section 3. For the measurement update step of the ETGM-PHD-filter, different partitions of the set of measurements have to be considered. A measurement clustering algorithm used to reduce the combinatorially exploding number of possible measurement partitions is described in Section 4. The proposed approaches are evaluated using both simulations and experiments. The target tracking setups for these evaluations are described in Section 5, the simulation results are presented in Section 6 and results using data from a laser sensor are presented in Section 7. Finally, Sections 8 and 9 contain conclusions and thoughts on future work.

2 Target Tracking Problem Formulation

In previous work, extended targets have often been modeled as targets having a spatial extension or shape that would lead to multiple measurements, as opposed to at most a single measurement. On the other hand, the extended target tracking problem can be simplified by the assumption that the measurements originating from a target are distributed approximately around a target reference point (Gilholm and Salmond, 2005) which can be e.g. the centroid or any other point depending on the extent (or the shape) of the target. Though all targets obviously have a spatial extension and shape, in the latter type of modeling, only the target reference point is important and the target extent does not need to be estimated.

The relevant target characteristics that are to be estimated form the target's state vector \mathbf{x} . Generally, beside the kinematic variables as position, velocity and orientation, the state vector may also contain information about the target's spatial extension. As mentioned above, when the target's state does not contain any variables related to the target extent, though the estimation is done as if the target was a point (i.e. the target reference point), the algorithms should still take care of the multiple measurements that originate from a target. Hence, in this study, we use a generalized definition of an extended target, given below, which does not depend on whether the target extent is estimated or not.

Definition B.1 (Extended Target). A target that potentially gives rise to more than one measurement per time step. _____

In this work, to simplify the presentation, no information about the size and shape of the target is kept in the state vector \mathbf{x} , i.e. the target extent is not explicitly estimated. Nevertheless, it must be emphasized that this causes no loss of generality as shown by the recent work (Granström et al., 2011) where the re-

sulting ETGMPHD filter is used to handle the joint estimation of size, shape and kinematic variables for rectangular and elliptical extended targets. We model both the target states to be estimated, and the measurements collected, as RFSSs. The motivation behind this selection is two-fold. First, in many practical systems, although the sensor reports come with a specific measurement order, the results of the target tracking algorithms are invariant under permutations of this order. Hence, modeling the measurements as elements of a set in which the order of the elements is irrelevant makes sense. Second, this work unavoidably depends on the previous line of work (Mahler, 2009), which is based on such a selection.

The initial GM-PHD work (Vo and Ma, 2006) does not provide tools for ensuring track continuity, for which some remedies are described in the literature, see e.g. (Panta et al., 2009). However it has been shown that labels for the Gaussian components can be included into the filter in order to obtain individual target tracks, see e.g. (Clark et al., 2006). In this work, for the sake of simplicity, labels are not used, however they can be incorporated as in (Clark et al., 2006) to provide track continuity.

We denote the unknown number of targets $N_{x,k}$, and the set of target states to be estimated at time k is $\mathbf{X}_k = \{\mathbf{x}_k^{(i)}\}_{i=1}^{N_{x,k}}$. The measurement set obtained at time k is $\mathbf{Z}_k = \{\mathbf{z}_k^{(i)}\}_{i=1}^{N_{z,k}}$ where $N_{z,k}$ is the number of measurements.

The dynamic evolution of each target state $\mathbf{x}_k^{(i)}$ in the RFS \mathbf{X}_k is modeled using a linear Gaussian dynamical model,

$$\mathbf{x}_{k+1}^{(i)} = F_k \mathbf{x}_k^{(i)} + G_k \mathbf{w}_k^{(i)}, \quad (1)$$

for $i = 1, \dots, N_{x,k}$, where $\mathbf{w}_k^{(i)}$ is Gaussian white noise with covariance $Q_k^{(i)}$. Note that each target state evolves according to the same dynamic model independent of the other targets.

The number of measurements generated by the i th target at each time step is a Poisson distributed random variable with rate $\gamma(\mathbf{x}_k^{(i)})$ measurements per scan, where $\gamma(\cdot)$ is a known non-negative function defined over the target state space. The probability of the i th target generating at least one measurement is then given as

$$1 - e^{-\gamma(\mathbf{x}_k^{(i)})}. \quad (2)$$

The i th target is detected with probability $p_D(\mathbf{x}_k^{(i)})$ where $p_D(\cdot)$ is a known non-negative function defined over the target state space. This gives the effective probability of detection

$$\left(1 - e^{-\gamma(\mathbf{x}_k^{(i)})}\right) p_D(\mathbf{x}_k^{(i)}). \quad (3)$$

The measurements originating from the i th target are assumed to be related to

the target state according to a linear Gaussian model

$$\mathbf{z}_k^{(j)} = H_k \mathbf{x}_k^{(i)} + \mathbf{e}_k^{(j)}, \quad (4)$$

where $\mathbf{e}_k^{(j)}$ is white Gaussian noise with covariance R_k . Each target is assumed to give rise to measurements independently of the other targets. We emphasize here, that in an RFS framework both the set of measurements \mathbf{Z}_k and the set of target states \mathbf{X}_k are unlabeled, and hence no assumptions are made regarding which target gives rise to which measurement.

The number of clutter measurements generated at each time step is a Poisson distributed random variable with rate $\beta_{FA,k}$ clutter measurements per surveillance volume per scan. Thus, if the surveillance volume is V_s , the mean number of clutter measurements is $\beta_{FA,k} V_s$ clutter measurements per scan. The spatial distribution of the clutter measurements is assumed uniform over the surveillance volume.

The aim is now to obtain an estimate of the sets of the target states $\mathbf{X}^K = \{\mathbf{X}_k\}_{k=1}^K$ given the sets of measurements $\mathbf{Z}^K = \{\mathbf{Z}_k\}_{k=1}^K$. We achieve this by propagating the predicted and updated PHDSs of the set of target states \mathbf{X}_k , denoted $D_{k|k-1}(\cdot)$ and $D_{k|k}(\cdot)$, respectively, using the PHD filter presented in (Mahler, 2009).

3 Gaussian-Mixture Implementation

In this section, following the derivation of the GM-PHD-filter for standard single measurement targets in (Vo and Ma, 2006), a PHD recursion for the extended target case is described. Since the prediction update equations of the extended target PHD filter are the same as those of the standard PHD filter (Mahler, 2009), the Gaussian mixture prediction update equations of the ETGM-PHD filter are the same as those of the standard GM-PHD filter in (Vo and Ma, 2006). For this reason, here we only consider the measurement update formulas for the ETGM-PHD filter.

The predicted PHD has the following Gaussian-mixture representation

$$D_{k|k-1}(\mathbf{x}) = \sum_{j=1}^{J_{k|k-1}} w_{k|k-1}^{(j)} \mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \quad (5)$$

where

- $J_{k|k-1}$ is the predicted number of components;
- $w_{k|k-1}^{(j)}$ is the weight of the j th component;
- $m_{k|k-1}^{(j)}$ and $P_{k|k-1}^{(j)}$ are the predicted mean and covariance of the j th component;
- $\mathcal{N}(x; m, P)$ denotes a Gaussian distribution defined over the variable x with mean m and covariance P .

The PHD measurement update equation for the extended target Poisson model of (Gilholm et al., 2005) was derived in (Mahler, 2009). The corrected PHD-intensity is given by the multiplication of the predicted PHD and a measurement pseudo-likelihood function (Mahler, 2009) $L_{\mathbf{Z}_k}$,

$$D_{k|k}(\mathbf{x}|\mathbf{Z}) = L_{\mathbf{Z}_k}(\mathbf{x}) D_{k|k-1}(\mathbf{x}|\mathbf{Z}). \quad (6)$$

The measurement pseudo-likelihood function $L_{\mathbf{Z}_k}$ in (6) is defined as

$$\begin{aligned} L_{\mathbf{Z}_k}(\mathbf{x}) \triangleq & 1 - \left(1 - e^{-\gamma(\mathbf{x})}\right) p_D(\mathbf{x}) + e^{-\gamma(\mathbf{x})} p_D(\mathbf{x}) \\ & \times \sum_{\mathfrak{p} \angle \mathbf{Z}_k} \omega_{\mathfrak{p}} \sum_{W \in \mathfrak{p}} \frac{\gamma(\mathbf{x})^{|W|}}{d_W} \cdot \prod_{\mathbf{z}_k \in W} \frac{\phi_{\mathbf{z}_k}(\mathbf{x})}{\lambda_k c_k(\mathbf{z}_k)}. \end{aligned} \quad (7)$$

where

- $\lambda_k \triangleq \beta_{FA,k} V_s$ is the mean number of clutter measurements;
- $c_k(\mathbf{z}_k) = 1/V_s$ is the spatial distribution of the clutter over the surveillance volume;
- the notation $\mathfrak{p} \angle \mathbf{Z}_k$ means that \mathfrak{p} partitions the measurement set \mathbf{Z}_k into non-empty cells W ;
- the quantities $\omega_{\mathfrak{p}}$ and d_W are non-negative coefficients defined for each partition \mathfrak{p} and cell W respectively.
- $\phi_{\mathbf{z}_k}(\mathbf{x}) = p(\mathbf{z}_k|\mathbf{x})$ is the likelihood function for a single target generated measurement, which would be a Gaussian density in this work.

The first summation on the right hand side of (7) is taken over all partitions \mathfrak{p} of the measurement set \mathbf{Z}_k . The second summation is taken over all cells W in the current partition \mathfrak{p} .

In order to derive the measurement update of the GM-PHD-filter, six assumptions were made in (Vo and Ma, 2006), which are repeated here for the sake of completeness.

Assumption B.1. All of the targets evolve and generate observations independently of one another. _____

Assumption B.2. Clutter is Poisson and independent of target-originated measurements. _____

Assumption B.3. The predicted multi-target RFS is Poisson. _____

Assumption B.4. Each target follows a linear Gaussian dynamical model, cf. (1), and the sensor has a linear Gaussian measurement model, cf. (4). _____

Assumption B.5. The survival and detection probabilities are state independent, i.e. $p_S(\mathbf{x}) = p_S$ and $p_D(\mathbf{x}) = p_D$. _____

Assumption B.6. The intensities of the birth and spawn RFS are Gaussian mixtures.

In this paper we adopt all of the above assumptions except that we relax the assumption on detection probability as follows.

Assumption B.7. The following approximation about the probability of detection function $p_D(\cdot)$ holds for all \mathbf{x} and for $j = 1, \dots, J_{k|k-1}$,

$$\begin{aligned} p_D(\mathbf{x}) &\mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \\ &\approx p_D\left(m_{k|k-1}^{(j)}\right)\mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right). \end{aligned} \quad (8)$$

Assumption B.7 is weaker than Assumption B.5 in that (8) is trivially satisfied when $p_D(\cdot) = p_D$, i.e. when $p_D(\cdot)$ is constant. In general, Assumption B.7 holds approximately when the function $p_D(\cdot)$ does not vary much in the uncertainty zone of a target determined by the covariance $P_{k|k-1}^{(j)}$. This is true either when $p_D(\cdot)$ is a sufficiently smooth function or when the signal to noise ratio (SNR) is high enough such that $P_{k|k-1}^{(j)}$ is sufficiently small. We still note here that Assumption B.7 is only for the sake of simplification rather than approximation, since $p_D(\mathbf{x})$ can always be approximated as a mixture of exponentials of quadratic functions (or equivalently as Gaussians) without losing the Gaussian-mixture structure of the corrected PHD, see (Vo and Ma, 2006). This, however, would cause a multiplicative increase in the number of components in the corrected PHD, which would in turn make the algorithm need more aggressive pruning and merging operations. A similar approach to variable probability of detection has been taken in order to model the *clutter notch* in ground moving target indicator target tracking (Ulmke et al., 2007).

For the expected number of measurements from the targets, represented by $\gamma(\cdot)$, similar remarks apply and we use the following assumption.

Assumption B.8. The following approximation about $\gamma(\cdot)$ holds for all \mathbf{x} , $n = 1, 2, \dots$ and $j = 1, \dots, J_{k|k-1}$,

$$\begin{aligned} e^{-\gamma(\mathbf{x})}\gamma^n(\mathbf{x})\mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \\ \approx e^{-\gamma\left(m_{k|k-1}^{(j)}\right)}\gamma^n\left(m_{k|k-1}^{(j)}\right)\mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right). \end{aligned} \quad (9)$$

The trivial situation $\gamma(\cdot) = \gamma$, i.e. when $\gamma(\cdot)$ is constant, is again a special case where Assumption B.8 is satisfied. In general, satisfying Assumption B.8 is more difficult than Assumption B.7 and a Gaussian mixture assumption for $\gamma(\cdot)$ would not work due to the exponential function. Nevertheless Assumption B.8 is expected to hold approximately either when $\gamma(\cdot)$ is a sufficiently smooth function

or when the signal to noise ratio (SNR) is high enough such that $P_{k|k-1}^{(j)}$ is sufficiently small.

With the assumptions presented above, the posterior intensity at time k is a Gaussian-mixture given by

$$D_{k|k}(\mathbf{x}) = D_{k|k}^{\text{ND}}(\mathbf{x}) + \sum_{p \in \mathcal{Z}_k} \sum_{W \in \mathcal{P}} D_{k|k}^{\text{D}}(\mathbf{x}, W). \quad (10)$$

The Gaussian-mixture $D_{k|k}^{\text{ND}}(\cdot)$, handling the no detection cases, is given by

$$D_{k|k}^{\text{ND}}(\mathbf{x}) = \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}), \quad (11a)$$

$$w_{k|k}^{(j)} = \left(1 - \left(1 - e^{-\gamma^{(j)}}\right) p_{\text{D}}^{(j)}\right) w_{k|k-1}^{(j)}, \quad (11b)$$

$$m_{k|k}^{(j)} = m_{k|k-1}^{(j)}, \quad P_{k|k}^{(j)} = P_{k|k-1}^{(j)}. \quad (11c)$$

where we used the short hand notations $\gamma^{(j)}$ and $p_{\text{D}}^{(j)}$ for $\gamma(m_{k|k-1}^{(j)})$ and $p_{\text{D}}(m_{k|k-1}^{(j)})$ respectively.

The Gaussian-mixture $D_{k|k}^{\text{D}}(\mathbf{x}, W)$, handling the detected target cases, is given by

$$D_{k|k}^{\text{D}}(\mathbf{x}, W) = \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}), \quad (12a)$$

$$w_{k|k}^{(j)} = \omega_{\text{p}} \frac{\Gamma^{(j)} p_{\text{D}}^{(j)}}{d_W} \Phi_W^{(j)} w_{k|k-1}^{(j)}, \quad (12b)$$

$$\Gamma^{(j)} = e^{-\gamma^{(j)}} \left(\gamma^{(j)}\right)^{|W|}, \quad (12c)$$

$$\Phi_W^{(j)} = \phi_W^{(j)} \prod_{\mathbf{z}_k \in W} \frac{1}{\lambda_k c_k(\mathbf{z}_k)}, \quad (12d)$$

where the product is over all measurements \mathbf{z}_k in the cell W and $|W|$ is the number of elements in W . The coefficient $\phi_W^{(j)}$ is given by

$$\phi_W^{(j)} = \mathcal{N}(\mathbf{z}_W; \mathbf{H}_W m_{k|k-1}^{(j)}, \mathbf{H}_W P_{k|k-1}^{(j)} \mathbf{H}_W^{\text{T}} + \mathbf{R}_W) \quad (12e)$$

and is calculated using

$$\mathbf{z}_W \triangleq \bigoplus_{\mathbf{z}_k \in W} \mathbf{z}_k, \quad (12f)$$

$$\mathbf{H}_W = \underbrace{[H_k^{\text{T}}, H_k^{\text{T}}, \dots, H_k^{\text{T}}]^{\text{T}}}_{|W| \text{ times}}, \quad (12g)$$

$$\mathbf{R}_W = \text{blkdiag}(\underbrace{R_k, R_k, \dots, R_k}_{|W| \text{ times}}). \quad (12h)$$

The operation \oplus denotes vertical vectorial concatenation. The partition weights ω_p can be interpreted as the probability of the partition p being true and are calculated as

$$\omega_p = \frac{\prod_{W \in p} d_W}{\sum_{p' \neq p} \prod_{W \in p'} d_{W'}}, \quad (12i)$$

$$d_W = \delta_{|W|,1} + \sum_{\ell=1}^{J_{k|k-1}} \Gamma^{(\ell)} p_D^{(\ell)} \Phi_W^{(\ell)} w_{k|k-1}^{(\ell)}, \quad (12j)$$

where $\delta_{i,j}$ is the Kronecker delta. The mean and covariance of the Gaussian components are updated using the standard Kalman measurement update,

$$m_{k|k}^{(j)} = m_{k|k-1}^{(j)} + \mathbf{K}_k^{(j)} \left(\mathbf{z}_W - \mathbf{H}_W m_{k|k-1}^{(j)} \right), \quad (13a)$$

$$P_{k|k}^{(j)} = \left(I - \mathbf{K}_k^{(j)} \mathbf{H}_W \right) P_{k|k-1}^{(j)}, \quad (13b)$$

$$\mathbf{K}_k^{(j)} = P_{k|k-1}^{(j)} \mathbf{H}_W^T \left(\mathbf{H}_W P_{k|k-1}^{(j)} \mathbf{H}_W^T + \mathbf{R}_W \right)^{-1}. \quad (13c)$$

In order to keep the number of Gaussian components at a computationally tractable level, pruning and merging is performed as in (Vo and Ma, 2006).

4 Partitioning the Measurement Set

As observed in the previous section, an integral part of extended target tracking with the PHD filter is the partitioning of the set of measurements (Mahler, 2009). The partitioning is important, since more than one measurement can stem from the same target. Let us exemplify the process of partitioning with a measurement set containing three individual measurements, $\mathbf{Z}_k = \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\}$. This set can be partitioned in the following different ways;

$$\begin{aligned} p_1 : W_1^1 &= \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\}, \\ p_2 : W_1^2 &= \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}\}, \quad W_2^2 = \{\mathbf{z}_k^{(3)}\}, \\ p_3 : W_1^3 &= \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(3)}\}, \quad W_2^3 = \{\mathbf{z}_k^{(2)}\}, \\ p_4 : W_1^4 &= \{\mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\}, \quad W_2^4 = \{\mathbf{z}_k^{(1)}\}, \\ p_5 : W_1^5 &= \{\mathbf{z}_k^{(1)}\}, \quad W_2^5 = \{\mathbf{z}_k^{(2)}\}, \quad W_3^5 = \{\mathbf{z}_k^{(3)}\}. \end{aligned}$$

Here, p_i is the i th partition, and W_j^i is the j th cell of partition i . Let $|p_i|$ denote the number of cells in the partition, and let $|W_j^i|$ denote the number of measurements in the cell. It is quickly realized that as the size of the measurement set increases,

the number of possible partitions grows very large. In order to have a computationally tractable target tracking method, only a subset of all possible partitions can be considered. In order to achieve good extended target tracking results, this subset of partitions must represent the most likely ones of all possible partitions.

In Section 4.1, we propose a simple heuristic for finding this subset of partitions, which is based on the distances between the measurements. Note that our proposed method is only one instance of a vast number of other clustering algorithms found in the literature, and that other methods could have been used. Some well-known alternatives are pointed out, and compared to the proposed partitioning method, in Section 4.2. An addition to the partitioning approach to better handle targets which are spatially close is described in Section 4.3.

4.1 Distance Partitioning

Consider a set of measurements $\mathbf{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^{N_z}$. Our partitioning algorithm relies on the following theorem.

Theorem 1. *Let $\mathbf{d}(\cdot, \cdot)$ be a distance measure and $d_\ell \geq 0$ be an arbitrary distance threshold. Then there is one and only one partition in which any pair of measurements $\mathbf{z}^{(i)}, \mathbf{z}^{(j)} \in \mathbf{Z}$ that satisfy*

$$\mathbf{d}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \leq d_\ell \quad (15)$$

are in the same cell. □

Proof: The proof is given in Appendix A.1 for the sake of clarity. □

Given a distance measure $\mathbf{d}(\cdot, \cdot)$, the distances between each pair of measurements can be calculated as

$$\Delta_{ij} \triangleq \mathbf{d}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}), \text{ for } 1 \leq i \neq j \leq N_z. \quad (16)$$

Theorem 1 says that there is a unique partition that leaves all pairs (i, j) of measurements satisfying $\Delta_{ij} \leq d_\ell$ in the same cell. An example algorithm that can be used to obtain this unique partition is given in Table 1. This algorithm is used to generate N_d alternative partitions of the measurement set \mathbf{Z} , by selecting N_d different thresholds

$$\{d_\ell\}_{\ell=1}^{N_d}, \quad d_\ell < d_{\ell+1}, \text{ for } \ell = 1, \dots, N_d - 1. \quad (17)$$

The alternative partitions contain fewer cells as the d_ℓ 's are increasing, and the cells typically contain more measurements.

The thresholds $\{d_\ell\}_{\ell=1}^{N_d}$ are selected from the set

$$\mathcal{D} \triangleq \{0\} \cup \{\Delta_{ij} | 1 \leq i < j \leq N_z\} \quad (18)$$

where the elements of \mathcal{D} are sorted in ascending order. If one uses all of the elements in \mathcal{D} to form alternative partitions, $|\mathcal{D}| = N_z(N_z - 1)/2 + 1$ partitions are obtained. Some partitions resulting from this selection might still turn out to be

identical, and must hence be discarded so that each partition at the end is unique. Among these unique partitions, the first (corresponding to the threshold $d_1 = 0$) would contain N_z cells with one measurement each. The last partition would have just one cell containing all N_z measurements. Notice that this partitioning methodology already reduces the number of partitions tremendously.

In order to further reduce the computational load, partitions in this work are computed only for a subset of thresholds in the set \mathcal{D} . This subset is determined based on the statistical properties of the distances between the measurements belonging to the same target.

Suppose we select the distance measure $\mathbf{d}(\cdot, \cdot)$ as the Mahalanobis distance, given by

$$\mathbf{d}_M(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \sqrt{(\mathbf{z}^{(i)} - \mathbf{z}^{(j)})^T R^{-1} (\mathbf{z}^{(i)} - \mathbf{z}^{(j)})}. \quad (19)$$

Then, for two target-originated measurements $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$ belonging to the same target, $\mathbf{d}_M(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ is χ^2 distributed with degrees of freedom equal to the measurement vector dimension. Using the inverse cumulative χ^2 distribution function, denoted $\text{invchi2}(\cdot)$, a unitless distance threshold,

$$\delta_{P_G} = \text{invchi2}(P_G), \quad (20)$$

can be computed for a given probability P_G . Simulations have shown that good target tracking results are achieved with partitions computed using the subset of distance thresholds in \mathcal{D} satisfying the condition $\delta_{P_L} < d_\ell < \delta_{P_U}$, for lower probabilities $P_L \leq 0.3$ and upper probabilities $P_U \geq 0.8$.

As a simple example, if there are four targets present, each with expected number of measurements 20, and clutter measurements are generated with $\beta_{FA} V_s = 50$, then the mean number of measurements collected each time step would be 130. For 130 measurements, the number of all possible partitions is given by the Bell number $B_{130} \propto 10^{161}$ (Rota, 1964). Using all of the thresholds in the set \mathcal{D} , 130 different partitions would be computed on average. Using the upper and lower probabilities $P_L = 0.3$ and $P_U = 0.8$, Monte Carlo simulations show that on average only 27 partitions are computed, representing a reduction of computational complexity several orders of magnitude.

4.2 Alternative Partitioning Methods

An alternative to using the proposed algorithm is to use a method which takes as input the final desired number of cells, denoted K , and then divides the set of measurements into K cells. The most well-known example of such a method is perhaps K -means clustering, see e.g. the textbooks (Bishop, 2006; Hastie et al., 2009). In the ETGMPHD-filter, one needs to generate alternative partitions, corresponding to different values of K between a lower and an upper threshold, denoted K_L and K_U . While the values for the partitioning parameters δ_{P_L} and δ_{P_U} in Distance Partitioning can be chosen using some intuitive arguments as above, it is less clear how K_L and K_U should be selected. One idea is to set $K_L = 1$ and

Table 1: *Distance Partitioning*

Require: $d_\ell, \Delta_{i,j}, 1 \leq i \neq j \leq N_z$.

- 1: CellNumbers(i) = 0, $1 \leq i \leq N_z$ {Set cells of all measurements to null}
- 2: CellId = 1 {Set the current cell id to 1}

Ensure: %Find all cell numbers

- 3: **for** $i = 1 : N_z$ **do**
 - 4: **if** CellNumbers(i) = 0 **then**
 - 5: CellNumbers(i) = CellId
 - 6: CellNumbers = FindNeighbors(i , CellNumbers, CellId)
 - 7: CellId = CellId+1
 - 8: **end if**
 - 9: **end for**
-

The recursive function FindNeighbors(\cdot, \cdot, \cdot) is given as

- 1: **function** CellNumbers = FindNeighbors(i , CellNumbers, CellId)
 - 2: **for** $j = 1 : N_z$ **do**
 - 3: **if** $j \neq i$ & $\Delta_{ij} \leq d_\ell$ & CellNumbers(j) = 0 **then**
 - 4: CellNumbers(j) = CellId
 - 5: CellNumbers = FindNeighbors(j , CellNumbers, CellId)
 - 6: **end if**
 - 7: **end for**
-

$K_U = |\mathbf{Z}_k|$, which corresponds to $\delta_{P_U} = \infty$ and $\delta_{P_L} = 0$ in Distance Partitioning. Doing so would significantly increase the computational complexity compared to Distance Partitioning, since a considerably higher number of partitions must be considered.

Another major difference between the suggested distance partitioning and K -means clustering is highlighted in Figure 1, which shows a measurement set that consists of $N_{z,k} = 13$ measurements, 10 of which are clustered in the northeast of the surveillance region and the other three are scattered individually. The intuitive way to cluster this set of measurements is into 4 clusters, which is achieved by Distance Partitioning using a distance threshold of about 25 m, as shown in the left plot of Figure 1. When there is a large number of measurements concentrated in one part of the surveillance area, as is the case in this example, K -means clustering tends to split those measurements into smaller cells, and merge remaining but far away measurements into large cells. This is illustrated in the right plot of Figure 1.

One reason behind this shortcoming of K -means is the initialization of the algorithm, where the initial cluster centers are chosen by uniform sampling. In order to overcome this problem, modifications to the standard K -means algorithm have been suggested, where initial clusters are chosen as separated as possible, see (Arthur and Vassilvitskii, 2007; Ostrovsky et al., 2006). This improved version of

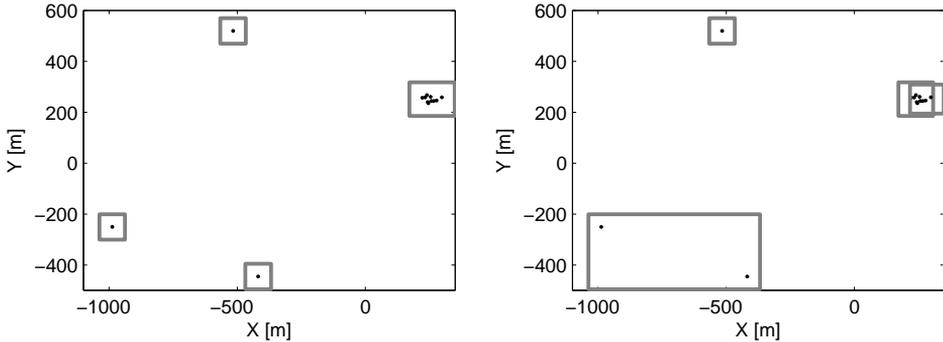


Figure 1: Set of $N_{z,k} = 13$ measurements. **Left:** The measurements partitioned using the suggested distance partitioning method with a distance threshold of 25 m. **Right:** The measurements partitioned using K -means clustering with $K = 4$.

K -means is called K -means++.

In simulations, Distance Partitioning was compared to K -means++ (using an implementation available online (Sorber, 2011)). The results, see Section 6.2, show that K -means++ fails to compute informative partitions much too often, except in scenarios with very low $\beta_{FA,k}$. This can be attributed to the existence of counter-intuitive local optima for the implicit cost function involved with K -means++ (or K -means). Distance Partitioning on the other hand can handle both high and low $\beta_{FA,k}$, and always gives an intuitive and unique partitioning for a given d_ℓ .

Therefore, we argue that a hierarchical method, such as the suggested Distance Partitioning, should be preferred over methods such as K -means. However, it is important to note here again, that regarding partitioning of the measurement set, the contribution of the current work lies mainly not in the specific method that is suggested, but rather in showing that all possible partitions can efficiently be approximated using a subset of partitions.

4.3 Sub-Partitioning

Initial results using ETGMPHD showed problems with underestimation of target set cardinality in situations where two or more extended targets are spatially close (Granström et al., 2010). The reason for this is that when targets are spatially close, so are their resulting measurements. Thus, using Distance Partitioning, measurements from more than one measurement source will be included in the same cell W in all partitions p , and subsequently the ETGMPHD filter will interpret measurements from multiple targets as having originated from just one target. In an ideal situation, where one could consider all possible partitions of the measurement set, there would be alternative partitions which would contain the subsets of a wrongly merged cell. Such alternative partitions would dominate the output of the ETGMPHD filter towards the correct estimated number of tar-

gets. Since we eliminate such partitions completely using Distance Partitioning, the ETGMPPHD filter lacks the means to correct its estimated number of targets.

One remedy for this problem is to form additional partitions after performing Distance Partitioning, and to add them to the list of partitions that ETGMPPHD filter considers at the current time step. Obviously, this should be done only when there is a risk of having merged the measurements belonging to more than one target, which can be decided based on the expected number of measurements originating from a target. We propose the following procedure for the addition of more partitions.

Suppose that we have computed a set of partitions using Distance Partitioning, e.g. with the algorithm in Table 1. Then, for each generated partition p_i , we calculate the maximum likelihood (ML) estimates \hat{N}_x^j of the number of targets for each cell W_j^i . If this estimate is larger than one, we split the cell W_j^i into \hat{N}_x^j smaller cells, denoted

$$\{W_s^+\}_{s=1}^{\hat{N}_x^j}. \quad (21)$$

We then add a new partition, consisting of the new cells along with the other cells in p_i , to the list of partitions obtained by Distance Partitioning.

We illustrate the Sub-Partition algorithm in Table 2, where the splitting operation on a cell is shown by a function

$$\text{split}\left(\hat{N}_x^j, W_j^i\right). \quad (22)$$

We give the details for obtaining the ML estimate \hat{N}_x^j and choosing the function $\text{split}(\cdot, \cdot)$ in the subsections below.

Computing \hat{N}_x^j

For this operation, we assume that the function $\gamma(\cdot)$ determining the expected number measurements generated by a target is constant, i.e. $\gamma(\cdot) = \gamma$. Each target generates measurements independently of the other targets, and the number of generated measurements by each target is distributed with the Poisson distribution, $\text{Pois}(\cdot, \gamma)$. The likelihood function for the number of targets corresponding to a cell W_j^i is

$$p\left(\left|W_j^i\right| \middle| N_x^j = n\right) = \text{Pois}\left(\left|W_j^i\right|, \gamma n\right). \quad (23)$$

Here, we assume that the volume covered by a cell is sufficiently small such that the number of false alarms in the cell is negligible, i.e. there are no false alarms in W_j^i . The ML estimate \hat{N}_x^j can now be calculated as

$$\hat{N}_x^j = \arg \max_n p\left(\left|W_j^i\right| \middle| N_x^j = n\right). \quad (24)$$

Table 2: Sub-Partition

Require: Partitioned set of measurements $\mathbf{Z}^p = \{p_1, \dots, p_{N_p}\}$, where N_p is the number of partitions.

- 1: **Initialize:** Counter for new partitions $\ell = N_p$.
- 2: **for** $i = 1, \dots, N_p$ **do**
- 3: **for** $j = 1, \dots, |p_i|$ **do**
- 4: $\hat{N}_x^j = \arg \max_n p \left(|W_j^i| \mid N_x^j = n \right)$
- 5: **if** $\hat{N}_x^j > 1$ **then**
- 6: $\ell = \ell + 1$ {Increase the partition counter}
- 7: $p_\ell = p_i \setminus W_j^i$ {Current partition except the current cell}
- 8: $\{W_k^+\}_{k=1}^{\hat{N}_x^j} = \text{split} \left(\hat{N}_x^j, W_j^i \right)$ {Split the current cell}
- 9: $p_\ell = p_\ell \cup \{W_k^+\}_{k=1}^{\hat{N}_x^j}$ {Augment the current partition}
- 10: **end if**
- 11: **end for**
- 12: **end for**

Note that other alternatives can be found for calculating the estimates of N_x^j , e.g. utilizing specific knowledge about the target tracking setup, however both simulations and experiments have shown that the above suggested method works well.

The $\text{split}(\cdot, \cdot)$ function

An important part of the Sub-Partition function in Table 2 is the function

$$\text{split}(\cdot, \cdot), \quad (25)$$

which is used to divide the measurements in a cell into smaller cells. In both simulations and experiments, we have used K -means clustering to split the measurements in the cell, results shows that this works well. However note that other methods to split the measurements are possible.

Remark 1 (Limitations of Sub-Partition). Notice that the Sub-Partition algorithm given in this section can be interpreted to be only a first-order remedy to the problem, and hence have limited correction capabilities. This is because we do not consider the combinations of the cells when we are adding new partitions. In the case, for example, where there are two pairs of close targets whose cells are merged wrongly by Distance Partitioning, the sub-partitioning algorithm presented above would add an additional partition for each of the target pairs (i.e. for each of the wrongly merged cells), but not an additional partition that contains the split versions of both cells. Consideration of all combinations of (the wrongly merged) cells seems to be prohibitive, due to the combinatorial growth in the number of additional partitions. An idea for the cases where there can be more than one

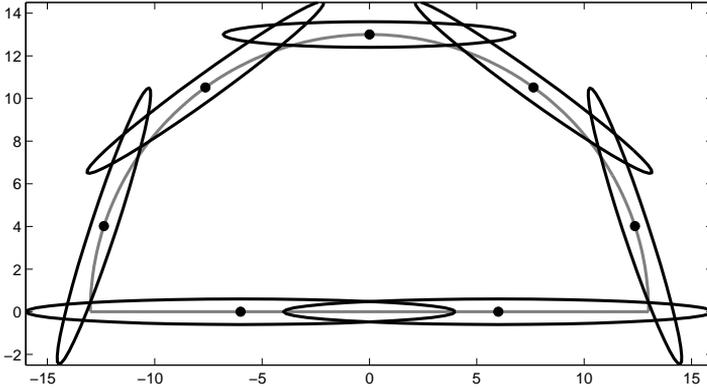


Figure 2: Birth intensity used in experiments.

wrongly merged cells is to add a single additional partition, which contains split versions of all such cells.

5 Target Tracking Setup

The presented tracking approach is exemplified with a laser sensor tracking humans at short distance. In this section the tracking setup is defined for both a pure simulation environment and an experimental realization with laser data. The targets are modeled as points with state variables

$$\mathbf{x}_k = [x_k \quad y_k \quad v_k^x \quad v_k^y]^T, \quad (26)$$

where x_k and y_k are the planar position coordinates of the target, and v_k^x and v_k^y are the corresponding velocities. The sensor measurements are given in batches of Cartesian x and y coordinates as follows;

$$\mathbf{z}_k^{(j)} \triangleq [x_k^{(j)} \quad y_k^{(j)}]^T. \quad (27)$$

A constant velocity model (Rong Li and Jilkov, 2003), with sampling time T is used. In all simulations the probability of detection and probability of survival are set to $p_D = 0.99$ and $p_S = 0.99$, respectively. The algorithm parameters for the simulation and experiment are given in Table 3. The surveillance area is $[-1000, 1000](\text{m}) \times [-1000, 1000](\text{m})$ for the simulations, and for the real data experiments the surveillance area is a semi circle located at the origin with range 13 m. Unless otherwise stated, in the simulations clutter was generated with a Poisson rate of 10 clutter measurements per scan, and each target generated measurements with a Poisson rate of 10 measurements per scan. The birth intensity in the simulations is

$$D_b(\mathbf{x}) = 0.1\mathcal{N}(\mathbf{x}; m_b, P_b) + 0.1\mathcal{N}(\mathbf{x}; -m_b, P_b), \quad (28a)$$

$$m_b = [250, 250, 0, 0]^T, P_b = \text{diag}([100, 100, 25, 25]). \quad (28b)$$

Table 3: Parameter values used for simulations (s) and experiments (e).

	T	Q_k	R_k	$\gamma^{(i)}$	w_β	Q_β
S	1	$2^2\mathbf{I}_2$	$20^2\mathbf{I}_2$	10	0.05	$\text{blkdiag}(100\mathbf{I}_2, 400\mathbf{I}_2)$
E	0.2	$2^2\mathbf{I}_2$	$0.1^2\mathbf{I}_2$	12	0.01	$0.01\mathbf{I}_4$

For the experiments, the birth intensity Gaussian components are illustrated with their corresponding one standard deviation ellipsoids in Fig. 2. Each birth intensity component has a weight $w_b^{(j)} = \frac{0.1}{J_b}$, where the number of components is $J_b = 7$. The spawn intensity is

$$D_\beta(\mathbf{x}|\mathbf{y}) = w_\beta \mathcal{N}(\mathbf{x}; \xi, Q_\beta), \quad (29)$$

where ξ is the target from which the new target is spawned and the values for w_β and Q_β are given in Table 3.

6 Simulation Results

This section presents results from simulations using the presented extended target tracking method. Section 6.1 presents three simulation scenarios that are used several times, and Section 6.2 presents a comparison of Distance Partitioning and K -means++. In Section 6.3 a comparison of Distance Partitioning and Distance Partitioning with Sub-Partition is presented, the results show the increased performance when using Sub-Partition. A comparison between ETGMPHD and GM-PHD is presented in Section 6.4, where it is shown that ETGMPHD as expected outperforms GM-PHD for extended targets. Section 6.5 presents a comparison of ETGMPHD and GM-PHD for targets that give rise to at most one measurement per time step. Finally, detailed investigations are carried out about the effects of the possibly unknown parameter γ in Section 6.6.

6.1 True target tracks

Three different scenarios are used in several simulations. The first two both have two targets. The true x, y positions and the distance between the targets are shown in Fig. 3a and Fig. 3b. At the closest points the targets are 60m and 50m apart, respectively. In the third scenario there are four targets in total, the true x, y positions are shown in black in Figure 3c. Around time 50–52 two target tracks cross at a distance of just over 50m, at time 67 a new target is spawned 20m from a previous one. Together the three scenarios present challenges that are typical in multiple target applications. In the simulations, the targets are modeled as points that generate measurements with standard deviation 20m in both x and y direction. Thus, a measure of target extent can be taken as the two standard deviation measurement covariance ellipses, in this case circles of radius 40m. In all three scenarios these circles partly overlap when the targets are closest to each other.

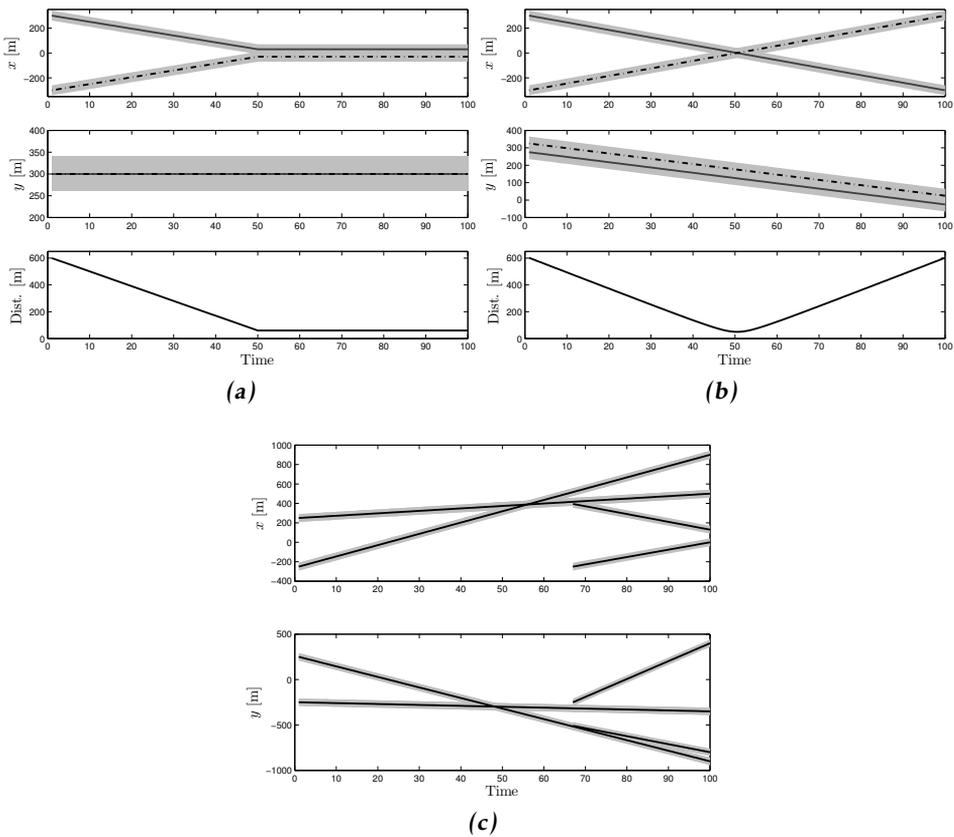


Figure 3: (a) Two targets move closer to each other and then stand still at a distance of 60m apart. Note that the true y position was 300m for both targets for the entire simulation. (b) Two targets cross paths, at the closest point they are 50m apart. (c) Four targets, with a target spawning event at time 67. The x and y positions are shown as lines, the light gray shaded areas show the target extent, taken as two measurement noise standard deviations (40m). In (a) and (b), the bottom row shows the distance between the two targets over time.

6.2 Comparison of Distance Partitioning and K -means++

The scenario in Figure 3b was used to compare Distance Partitioning and K -means++. In order to make the comparison as fair as possible, the upper and lower thresholds were set to $K_L = 1$, $K_U = |\mathbf{Z}_k|$, $\delta_{P_U} = \infty$ and $\delta_{P_L} = 0$, respectively. The scenario was simulated with a Poisson rate of 1 and 10 clutter measurements per scan. For each clutter rate, the scenario was simulated 100 times, Figure 4 shows the resulting sum of weights. At the lower clutter rate, K -means++ yields a small positive bias in estimated target number, but the performance is otherwise

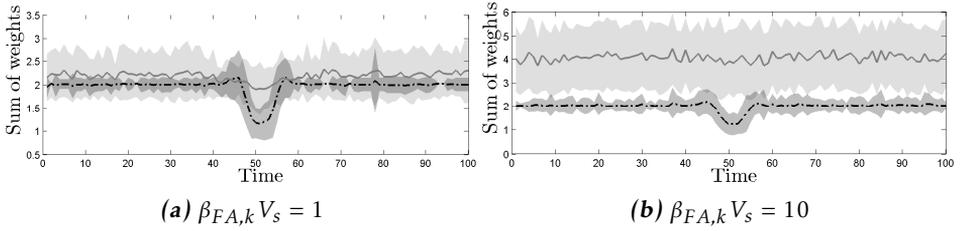


Figure 4: Results from the comparison of Distance Partitioning (black dash-dotted line) and K-means++ (gray solid line), the shaded areas correspond to \pm one standard deviation. At the lower clutter rate, K-means++ performs adequately, however at the higher clutter rate the performance is unacceptable. Distance Partitioning on the other hand handles both the lower and higher clutter rate, and has a much smaller uncertainty area.

good. However, at the higher clutter rate the performance using K-means++ is far from acceptable. Distance Partitioning, on the other hand, handles both clutter rates equally well, except for when the targets are close around time 50. Note also that using Distance Partitioning, the sum of weights uncertainty area is considerably smaller. The case of close targets is investigated further in the next subsection, using the countermeasure introduced in Section 4.3.

6.3 Benefits of Sub-Partition

As was noted in Section 4.3, as well as in previous work (Granström et al., 2010), using only Distance Partitioning to obtain a subset of all possible partitions is insufficient when the extended targets are spatially close. For this reason, Sub-Partition was introduced to obtain more partitions. In this section, we present results from simulations that compare the performance of ETGM-PHD tracking with partitions computed using only Distance Partitioning and with partitions computed using Distance Partitioning and Sub-Partition. The scenarios in Fig. 3a and Fig. 3b are considered.

Each scenario was simulated 100 times with a constant expected number of measurements per target ($\gamma(\cdot) = \gamma$) of 5, 10 and 20, respectively. Fig. 5 shows the resulting sum of weights of the ETGM-PHD algorithm. As can be seen, using Sub-Partition the average sum of weights is closer to the true number of targets. This is especially clear for targets that generate more measurements per time step, i.e. when γ is higher.

6.4 Comparison with GM-PHD

This section presents results from a simulation comparison of ETGM-PHD and GM-PHD. Note here that the GM-PHD filter is applied naively to the simulated measurements, i.e. it is applied under the (false) assumption that each target produces at most one measurement per time step.

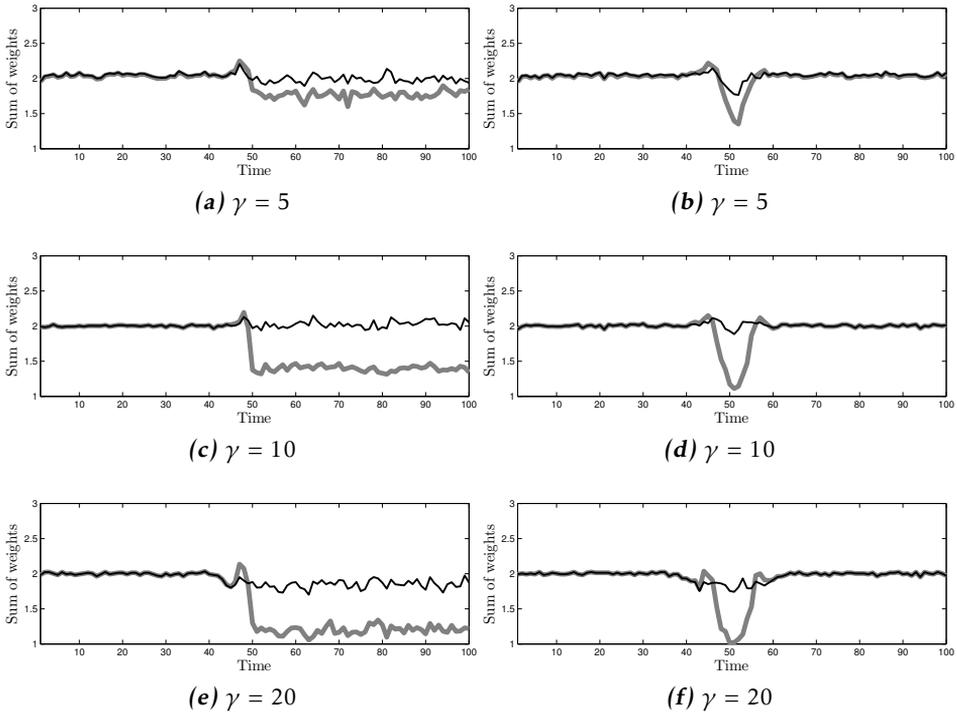


Figure 5: Simulation results for two of the scenarios in Fig. 3, comparing different partitioning methods for different values of the expected number of measurements γ . The left column, (a), (c) and (e), is for the true tracks in Fig. 3a. The right column, (b), (d) and (f), is for the true tracks in Fig. 3b. Black shows Distance Partitioning with Sub-Partition, gray is only Distance Partitioning. It is clear from the plots that using Sub-Partition gives significantly better results, especially when γ is higher.

The scenario in Figure 3c is considered. In total 100 Monte Carlo simulations were performed, each with new measurement noise and clutter measurements. The results are shown in Figure 6a and Figure 6b, which show the corresponding multi-target measure optimal subpattern assignment metric (OSPA) (Schuhmacher et al., 2008), and the cardinality, respectively. In the OSPA metric the parameters are set to $p = 2$, corresponding to using the 2-norm which is a standard choice, and $c = 60$, corresponding to a maximum error equal to three measurement noise standard deviations. Here, the cardinality is computed as $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$. This sum can be rounded to obtain an integer estimate of target number (Vo and Ma, 2006).

It is evident from the two figures that the presented ETGMPHD significantly outperforms the standard GM-PHD, which does not take into account the possibility

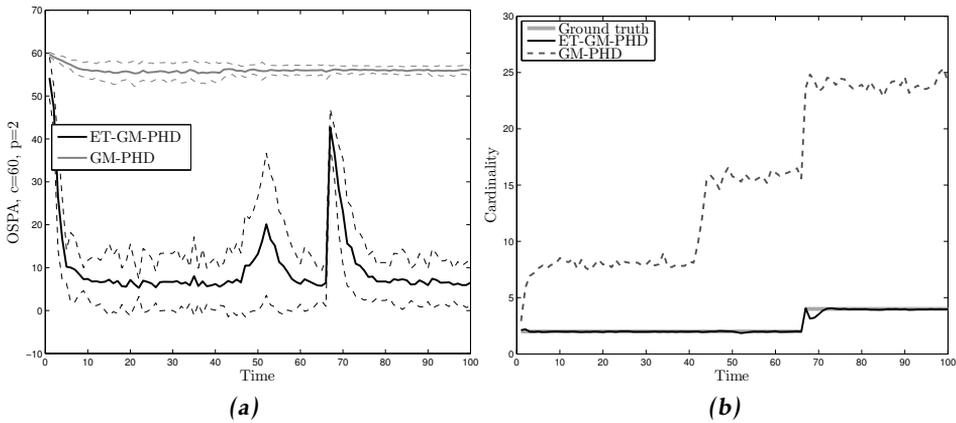


Figure 6: Results from multiple target tracking using the true tracks in Figure 3c. (a) Mean OSPA (solid lines) ± 1 standard deviation (dashed lines). (b) Mean cardinality compared to the true cardinality.

of the multiple measurements from single targets. The main difference between the two filters is the estimation of cardinality, i.e. the number of targets. The ET-GMPHD-filter correctly estimates the cardinality with the exception of when the new target is spawned – after time 67 there is a small dip in the mean estimated cardinality, even though Sub-Partition is used. The reason for this is that the targets are only 20m apart. With the target extension being a circle of 40m radius, at 20m distance the measurements overlap significantly and the probability that the new target’s measurements were also generated by the old target, as computed in (12e), is large. As the targets separate, this probability decreases and the ETGM-PHD filter recovers the correct cardinality. It should still be noted that, in reality, where the targets would probably be rigid bodies, this type of very close situation is highly unlikely and the results of the ETGMPHD filter with Sub-Partition would be close to those presented in Section 6.3.

6.5 Standard single measurement targets

This section investigates how ETGMPHD handles standard targets that produce at most one measurement per time step, in comparison to standard GM-PHD which is designed under the standard target measurement generation assumption. Note that the measurement set cardinality distribution (i.e. the probability mass function for the number of measurements generated by a target) for a standard target contains only a single nonzero element at cardinality 1, which is impossible to model with a Poisson distribution underlying the ETGMPHD filter. A standard target always generates a single measurement. Whether no measurement or a single measurement is obtained from the standard target is determined by the detection process. Hence, the case where each target generates measurements whose number is Poisson distributed with rate $\gamma = 1$ is very different from the

standard target measurement generation.

Four targets were simulated in 100 Monte Carlo simulations, and all the targets were separated, i.e. there were no track crossings or new target spawn. Initially, in the ETGMPPHD filter, $\gamma^{(j)}$ are all set as $\gamma^{(j)} = 1$ in the comparison. The average sum of weights and the average number of extracted targets (obtained by rounding the weights to the nearest integer) for the algorithms are shown in Fig. 7a and Fig. 7b respectively. As is shown, the sum of weights and number of extracted targets are too large for the ETGMPPHD filter. The reason for this is that when the expected number of measurements per target (i.e. $\gamma^{(j)}$) is small, the effective probability of detection

$$p_{D,\text{eff}}^{(j)} = \left(1 - e^{-\gamma^{(j)}}\right) p_D^{(j)} \quad (30)$$

becomes significantly smaller than one. More correctly, $p_{D,\text{eff}}^{(j)}$ in (30) is the probability of the event that at least one measurement from the (j th) target is obtained by the sensor. For example, the case $\gamma^{(j)} = 1$ and $p_D^{(j)} = 0.99$ gives $p_{D,\text{eff}}^{(j)} = 0.6258$. This low effective probability of detection is what causes the weights in the ETGMPPHD filter to become too large.

Actually, this problem has been seen to be inherited by the ETGMPPHD filter from the standard PHD filter. We here give a simple explanation to the problem with low (effective) probability of detection in the PHD filter. Assuming no false alarms, and a single target with existence probability p_E , ideally a single detection should cause the expected number of targets to be unity. However, applying the standard PHD formulae to this simple example, one can calculate the updated expected number of targets to be $1 + p_E(1 - p_D)$ whose positive bias increases as p_D decreases. We have seen that when the (effective) probability of detection is low, the increase in $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ is a manifestation of this type of sensitivity of the PHD type filters. Some extreme versions of this phenomenon for lower p_D values are illustrated and investigated in detail in (Orguner et al., 2011). A similar sensitivity issue is mentioned in (Erdinc et al., 2009) for the case of no detection.

This problem can be overcome by increasing $\gamma^{(j)}$ slightly in the ETGMPPHD filter, e.g. $\gamma^{(j)} = 2$ gives $p_{D,\text{eff}}^{(j)} = 0.8560$ which gives sum of weights and number of extracted targets that better match the results from GM-PHD, see Fig. 7c and Fig. 7d. Using $\gamma^{(j)} = 3$ gives results that are more or less identical to GM-PHD, thus a conclusion that can be drawn is that when tracking standard targets with an ETGMPPHD filter, the parameter $\gamma^{(j)}$ should not be set too small. The following subsection investigates the issue of selecting the parameter γ in more detail.

6.6 Unknown expected number of measurements γ

In the simulations above, the parameters $\gamma = \gamma^{(j)}$ were assumed to be known a priori. Further, in Section 4.3 where Sub-Partition is presented, the knowledge of the Poisson rate γ was used to determine whether a cell should be split or not to create an additional partition. In this section, some scenarios where γ is

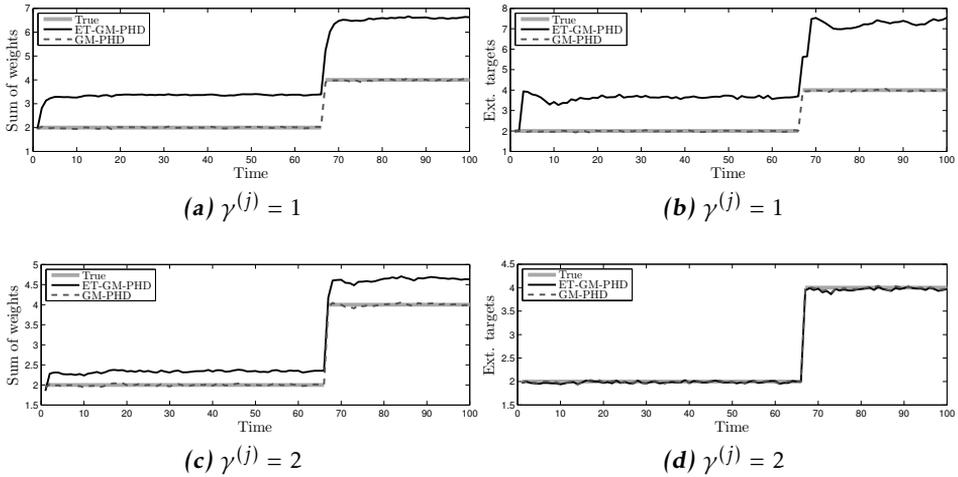


Figure 7: Simulation results, comparison of ETGM-PHD and GM-PHD for standard targets that produce at most one measurement per time step. The top row shows results when the parameter $\gamma^{(j)}$ is set to one, the bottom row shows results when it is set to two. Due to the low effective probability of detection, the ETGM-PHD weights become too large, resulting in sum of weights larger than the true number of targets. When each weight is rounded to the nearest integer to extract targets, results for $\gamma^{(j)} = 2$ gives the correct number of extracted targets.

not known a priori are investigated. For the sake of clarity, γ is used to denote the true Poisson rate with which measurements were generated, and $\hat{\gamma}$ is used to denote the corresponding parameter in the ETGM-PHD-filter.

In many real world scenarios, the number of measurements generated by a target is dependent on the distance between the target and the sensor. Typically, the longer the distance, the lower the number of measurements generated by the targets. This is true for sensors such as laser range sensors, radars and even cameras. Thus, it is of interest to evaluate the ETGM-PHD-filter in a scenario where the number of generated measurements varies with the target to sensor distance. This is simulated in Section 6.6, where the ETGM-PHD filter is compared for the cases when the parameter $\hat{\gamma}$ is constant, and when the parameter is modeled as distance varying. Section 6.6 presents results from simulations where the parameter $\hat{\gamma}$ is set incorrectly, and Section 6.6 presents results with targets of different sizes. Finally, Section 6.6 presents a discussion about the results from the simulations, and supplies some guidelines into the selection of $\hat{\gamma}$.

Distance varying γ

A scenario was constructed where a target moved such that the target to sensor distance first decreased, and then increased. The sensor was simulated such that

the true parameter γ depended on the target to sensor distance ρ as follows.

$$\gamma(\rho) = \begin{cases} 1, & \text{if } \rho > 400\text{m} \\ \lfloor -0.08\rho + 33.5 \rfloor, & \text{if } 100\text{m} \leq \rho \leq 400\text{m} \\ 25, & \rho < 100\text{m} \end{cases} \quad (31)$$

where $\lfloor \cdot \rfloor$ is the floor function. Thus, at distances larger than 400m, with $p_D^{(j)} = 0.99$, the effective probability of detection is only 0.6258 (as in the previous subsection). Note that the scenario is different from a target that always generates one measurement, which is detected with probability $p_D^{(j)} = 0.99$.

Monte Carlo simulations were made with two ETGMPHD-filters: one with constant value $\hat{\gamma} = 10$ and another where $\hat{\gamma}$ was selected to be dependent on the state of the targets via the function (31). The results are shown in Figure 8. For constant $\hat{\gamma}$, the number of targets is underestimated when the true γ is low. This is due to the fact that the filter expects a target to generate more measurements, and thus the likelihood that some small number of measurements are all clutter is higher. However, at distances ρ such that $\gamma(\rho) > 5$, $\hat{\gamma} = 10$ works quite well. When the model (31) for distance dependent γ is assumed known, the results are much more reasonable and acceptable. The only, and possibly negligible, drawback seems to be the number of targets being slightly overestimated. There are two main reasons for this. The first reason is the low effective probability of detection when $\hat{\gamma}$ is low. When $\hat{\gamma}$ becomes smaller than 5, this behavior is more evident. The second reason is that the clutter falling into the region $\rho > 400\text{m}$ (i.e. when the true parameter is $\gamma = 1$) is interpreted as targets to some extent, which causes a positive, though small, bias in the estimation of number of targets. In that region, the target behavior is fairly similar to the clutter behavior which results in some Gaussian components with small weights surviving until the situation is resolved.

Incorrect γ parameter

In this simulation study, the target tracks in Figure 3b were used. Each target generated measurements with a Poisson rate of $\gamma = 20$ and eleven different ETGMPHD-filters, each using a different $\hat{\gamma}$ value, were run. The set of $\hat{\gamma}$ values used is given as

$$\hat{\gamma} = 10, 12, \dots, 28, 30. \quad (32)$$

The results, in terms of the sum of weights averaged over the Monte Carlo runs, are shown in Figure 9. The figure shows that for sufficiently separated targets, the ETGMPHD-filter correctly estimates the number of targets for all values of $\hat{\gamma}$. However, for spatially close targets, the ETGMPHD-filter overestimates the number of targets when $\hat{\gamma}$ is set too low, and underestimates the number of targets when $\hat{\gamma}$ is set too high. This result is actually expected, and is a direct consequence of the simple Sub-Partition algorithm which is used. When $\hat{\gamma}$ is too low, Sub-Partition creates an additional partition with too many cells, causing the overestimation of number of targets. Conversely, when $\hat{\gamma}$ is too high, Sub-Partition does not create

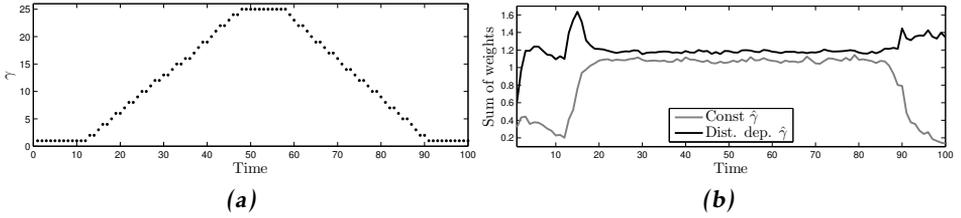


Figure 8: Results from the simulation scenario where γ is dependent on the target to sensor distance. In (a), the true γ is plotted against time, and in (b) the mean sum of weights is plotted against time. The ETGMPHD-filter is compared for the cases when the parameter $\hat{\gamma}$ is held constant (gray) or is set to the true distance dependent value (black). The correct target number is one, thus the sum of weights should be around one. In both cases, at the beginning and the end of the scenario when the distance is largest and $\gamma = 1$, tracking performance gets worse.

partitions with sufficient number of cells, causing the underestimation of number of targets. It is very important to note here that Sub-Partition runs even when the targets are well separated and does not cause any overestimation. Our observations show that this is a result of the fact that additional partitions created (when $\hat{\gamma}$ is selected too low) cannot win over single target partitions when the targets measurements are distributed in a blob shape. It is only when the two targets approach each other, resulting in an eight-shaped cluster of measurements, that the additional partition can gain dominance over the single target partition. This property, though not proved mathematically, is considered to be a manifestation of the Poisson property and the Gaussian assumptions underlying the measurements.

If the cardinality estimates of the algorithms are rounded to the nearest integer, an interesting property observed with Figure 9 is that no cardinality error appears for the cases that satisfy

$$\hat{\gamma} - \sqrt{\hat{\gamma}} \leq \gamma \leq \hat{\gamma} + \sqrt{\hat{\gamma}}. \quad (33)$$

Thus, when the true parameter γ lies in the interval determined by the mean ($\hat{\gamma}$) \pm one standard deviation ($\sqrt{\hat{\gamma}}$), cardinality is expected to be estimated correctly even for close targets.

Targets of different size

In many scenarios, it is possible to encounter multiple targets of different sizes, thus producing a different number of measurements. This means that two targets would not have the same Poisson rate γ . In this section, results are presented for a scenario with two targets with measurement generating Poisson rates of 10 and 20, respectively. In Monte Carlo simulations, three ETGMPHD-filters were run with the parameter $\hat{\gamma}$ set to 10, 15 and 20, respectively. This corresponds to

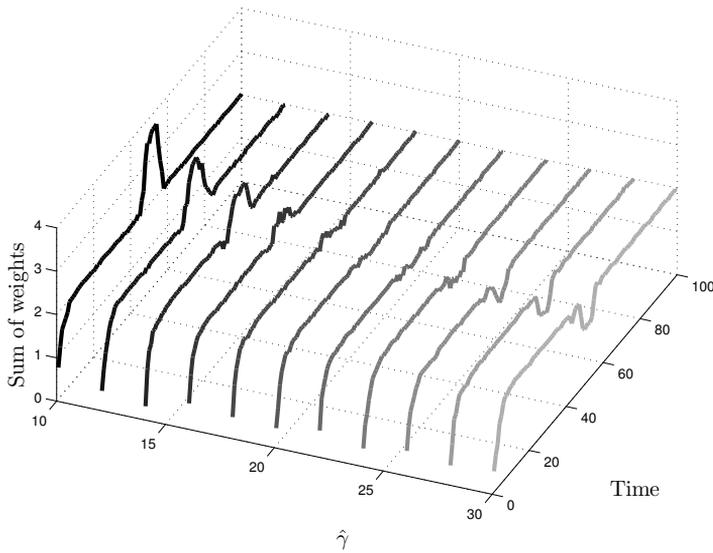


Figure 9: Simulation results for various values of the ETGMPHD-filter parameter $\hat{\gamma}$. There are two targets, the true Poisson rate used to generate measurements for both targets was $\gamma = 20$.

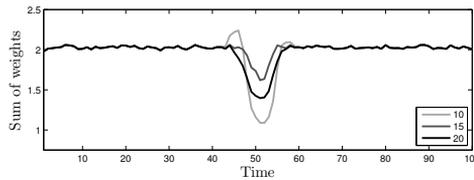


Figure 10: Simulation results from a scenario with two targets of different sizes. The two targets have the true Poisson rates $\gamma = 10$ and $\gamma = 20$, respectively. The legend refers to the filter parameter $\hat{\gamma}$.

using either the true value of the smaller target, the mean of both, or the true value of the larger target. The results, in terms of average sum of weights over time are shown in Figure 10. When the targets are spatially separated, all three filters perform equally well. However, when the targets are spatially close, the target with $\hat{\gamma}$ set to the mean of the true γ s performs better than the others.

Discussion

The simulation results above show that the ETGMPHD-filter works well even when $\hat{\gamma} \neq \gamma$, except when $\gamma < 5$ or targets are spatially close. For $\gamma < 5$, the filter is more sensitive, and a correct value for $\hat{\gamma}$ is increasingly important. For targets that are spatially close, it is important for $\hat{\gamma}$ to be a good estimate of γ , since the Sub-Partition algorithm relies on $\hat{\gamma}$. When such a good estimate is unavailable, a more advanced sub-partitioning algorithm seems to be necessary for robustness.

With the proposed Sub-Partition procedure, our findings support the intuitive conclusion that the true parameter γ should be in one standard deviation uncertainty region around the mean $\hat{\gamma}$ of the Poisson distribution for a reasonable performance for close targets.

The simulation with different target sizes shows that the close target case in this example is harder to tackle than the others. A possible solution is to adaptively estimate the parameters $\hat{\gamma}$ for each Gaussian mixture component, based on the previous measurements. Another solution, which is possibly more straightforward, is to use a state dependent $\hat{\gamma}$ parameter, where the state contains information about the target extent, which can readily be estimated, see e.g. (Granström et al., 2011; Lundquist et al., 2011; Baum et al., 2010b; Baum and Hanebeck, 2011; Zhu et al., 2011). Using the estimated shape and size, and a model of the sensor that is used, $\hat{\gamma}$ can then be estimated with reasonable accuracy. This has indeed recently been performed using an ETGMPHD-filter (Granström et al., 2011).

7 Experiment Results

This section presents results from experiments with data from two data sets acquired with a laser range sensor. The experiments are included more as a proof of concept and as a potential application, rather than as an exhaustive evaluation of the presented target tracking filter. The measurements were collected using a SICK LMS laser range sensor. The sensor measures range every 0.5° over a 180° surveillance area. Ranges shorter than 13 m were converted to (x, y) measurements using a polar to Cartesian transformation.

The two data sets contain 411 and 400 laser range sweeps in total, respectively. During the data collection humans moved through the surveillance area, entering the surveillance area at different times. The laser sensor was at the waist level of the humans.

Because there is no ground truth available it is difficult to obtain a definite measure of target tracking quality, however by examining the raw data we were able to observe the true cardinality, which can thus be compared to the estimated cardinality.

Section 7.1 presents results from an experiment with spatially close targets, and Section 7.2 presents results from an experiment with both spatially close targets and occlusion.

7.1 Experiment with close targets

In this experiment, a data set containing 411 laser range scans was used. The data set contains two human targets that repeatedly move towards and away from each other, moving right next to each other at several times. The two targets passed each other at close distance moving in the opposite direction, representing instances in time when the targets were close for short periods of time. The

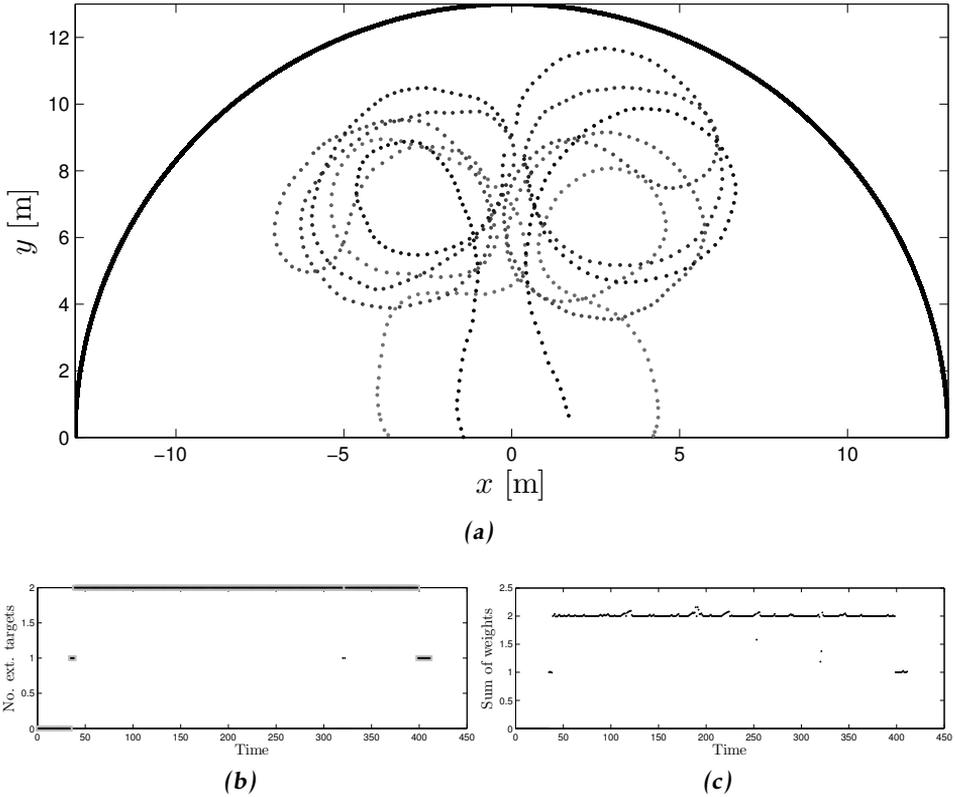


Figure 11: Experiment results, two human targets moving close to each other. Note that in (a) the gray scale indicates the time line, lighter gray is earlier time steps, darker is later time steps. In (b), the number of extracted targets (black) is compared to the ground truth (gray). In (c) the sum of weights is shown. Around time 320 the cardinality is underestimated for three time steps.

targets also moved close to each other moving in the same direction, representing instances in time when the targets were close for longer periods of time.

The locations of the extracted Gaussian components are shown in Fig. 11a, the number of extracted targets is shown in Fig. 11b and the sum of weights are shown in Fig. 11c. Around time 320 there is a decrease in the number of extracted targets for three time steps, in all other situations the filter handles the two targets without problem. Thus, using Sub-Partition with K -means as $\text{split}(\cdot, \cdot)$ function, the ETGMPHD filter can be said to handle most of the spatially close target cases.

7.2 Experiment with occlusion

In this experiment, a dataset containing 400 laser range scans was used. The data set contains four human targets that move through the surveillance area, however there are at most three targets present at any one time. The first target enters the surveillance area at time $k = 22$ and proceeds to the center of the surveillance area where he remains still for the remainder of the experiment. The second target enters the surveillance area at time $k = 38$ and repeatedly moves in front of and behind the first target. The third target enters and exits at time $k = 283$ and $k = 310$, respectively. The last target enters and exits at time $k = 345$ and $k = 362$, respectively.

This case requires a state dependent (i.e. variable) probability of detection $p_D(\cdot)$ selection for the targets. Otherwise, i.e. with a constant probability of detection assumption, when a target is occluded, this would be interpreted as the exit of the target from the area of surveillance while it is only the disappearance of the target behind another. The variable p_D is modeled as a function of the mean, covariance and the weights of the Gaussian components. The intuition behind this idea is that the knowledge of the targets that are present, i.e. the knowledge of the estimated Gaussian components of the PHD, can be used to determine what parts of the surveillance area are likely to be in view of the sensor, and which parts are not. Leaving the details of the variable p_D calculation to Appendix A.2, we present the results below.

The locations of the extracted Gaussian components are shown in Fig. 12a, the number of extracted targets is shown in Fig. 12b and the sum of weights are shown in Fig. 12c. In total, there are six situations where one target is occluded by another. The extracted number of targets is incorrect in two of these situations, where the occluded target is spatially very close to (right next to) the target which is causing the occlusion. The ETGMPHD filter correctly estimates the cardinality in four out of six occlusions.

Thus, using the suggested variable p_D , the filter can correctly predict the target while it is occluded, provided that it is not very close to another target while the occlusion is happening. If $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ is rounded to the nearest integer there is no cardinality error for the first four occlusions. However, as the target exits the occluded area there is a “jump” in $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ around times $k = 75$, $k = 125$, $k = 175$ and $k = 210$, see Fig. 12c. We have seen that this “jumping” behavior is caused by the sensitivity of the cardinality estimates of the PHD filter to detections when $p_D^{(j)}$ is set to a low value, which is the case when the target is half occluded while it gets out of the occluded region. This is the same phenomenon observed with low effective probability of detection in Section 6.5.

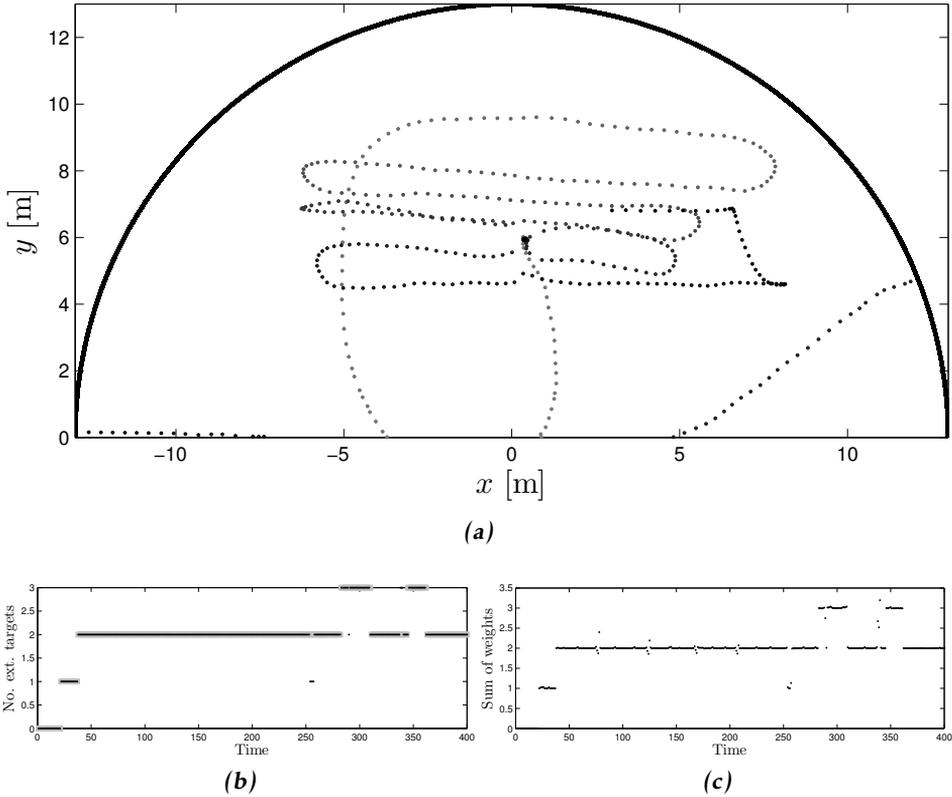


Figure 12: Experiment results, targets moving in and out of occluded regions of the surveillance area. Note that in (a) the gray scale indicates the time line, lighter gray is earlier time steps, darker is later time steps. In (b), the number of extracted targets (black) is compared to the ground truth (gray). In (c) the sum of weights is shown.

8 Conclusions

In this paper a Gaussian mixture implementation of the probability hypothesis density filter for tracking extended targets was presented. It was shown that all possible partitions of the measurement set does not need to be considered, instead it is sufficient to consider a subset of partitions, as long as this subset contains the most probable partitions. A simple method for finding this subset of all measurement set partitions was described. This partitioning method is complemented with a sub-partitioning strategy to handle the cases that involve close targets better. Simulations and experiments have shown that the proposed filter is capable of tracking extended targets in cluttered measurements. The number of targets is estimated correctly even for most of the cases when tracks are close. The detailed investigations carried out gave some guidelines about

Table 4: Find partition p that satisfies the conditions of Theorem 1

Require: Set of measurements $\mathbf{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(N_z)}\}$, where N_z is the number of measurements.

1: Set $p^0 = \{\{\mathbf{z}^{(1)}\}, \{\mathbf{z}^{(2)}\}, \dots, \{\mathbf{z}^{(N_z)}\}\}$ i.e., set $W_j^0 = \{\mathbf{z}^{(j)}\}$ for $j = 1, \dots, N_z$.

2: Set $i = 1$.

3: Calculate all the pairwise distances between the cells of p_{i-1} as

$$\eta_{st}^{i-1} = \min_{\substack{\mathbf{z}^{(m)} \in W_s^{i-1} \\ \mathbf{z}^{(n)} \in W_t^{i-1}}} \mathbf{d}(\mathbf{z}^{(m)}, \mathbf{z}^{(n)}) \quad (34)$$

4: If $\min_{1 \leq s \neq t \leq |p^{i-1}|} \eta_{st}^{i-1} > d_\ell$, then stop the algorithm, since p^{i-1} is a partition satisfying the conditions of the theorem.

5: Otherwise, combine all cells that satisfy $\eta_{st}^{i-1} \leq d_\ell$ to form a single cell.

6: Set p^i to be the set of cells obtained in Step 5.

7: Set $i = i + 1$ and go to Step 3.

the selection of the Poisson rate parameter for the cases when it is unknown. Using inhomogeneous detection probabilities in the surveillance region, it was shown that targets can be tracked as they move through occluded parts of the surveillance area.

9 Future Work

In recent work, a cardinalized PHD filter (Mahler, 2007a) for extended targets has been presented (Orguner et al., 2011). This filter has less sensitive estimates of the number of targets. Initial steps have also been taken towards including estimation of target extent in the ETGMPHD-filter (Granström et al., 2011). More work is needed in both of these research directions.

A further interesting research can be to see the potential use of the partitioning algorithms presented in this work with more conventional multiple target tracking algorithms. A comparison of such algorithms with the ETGMPHD filter can illustrate the advantage coming from the use of the random set framework.

A Appendix

A.1 Proof of Theorem 1

The proof is composed of two parts.

- We first prove that there is a partition satisfying the conditions of the theorem. The proof is constructive. Consider the algorithm in Table 4. In the algorithm, one first forms a partition formed of singleton sets of the

individual measurements and then combine the cells of this cluster until conditions of the theorem are satisfied. \square

- We need to prove that the partition satisfying the conditions of the theorem is unique. The proof is by contradiction. Suppose that there are two different partitions p_i and p_j satisfying the conditions of the theorem. Then, there must exist (at least) one measurement $\mathbf{z}^{(m)} \in \mathbf{Z}$ such that the cells $W_{m_i}^i \ni \mathbf{z}^{(m)}$ and $W_{m_j}^j \ni \mathbf{z}^{(m)}$ are different, i.e., $W_{m_i}^i \neq W_{m_j}^j$. This requires (at least) a single measurement $\mathbf{z}^{(n)} \in \mathbf{Z}$ that is in one of $W_{m_i}^i, W_{m_j}^j$ but not in the other. Without loss of generality, suppose $\mathbf{z}^{(n)} \in W_{m_i}^i$ and $\mathbf{z}^{(n)} \notin W_{m_j}^j$. Since both $\mathbf{z}^{(m)}$ and $\mathbf{z}^{(n)}$ are in $W_{m_i}^i$, there must exist a (possibly empty) subset $\{\mathbf{z}^{(r_1)}, \mathbf{z}^{(r_2)}, \dots, \mathbf{z}^{(r_R)}\} \subset W_{m_i}^i \setminus \{\mathbf{z}^{(m)}, \mathbf{z}^{(n)}\}$ such that the following conditions hold.

$$\mathbf{d}(\mathbf{z}^{(m)}, \mathbf{z}^{(r_1)}) \leq d_\ell \quad (35a)$$

$$\mathbf{d}(\mathbf{z}^{(r_s)}, \mathbf{z}^{(r_{s+1})}) \leq d_\ell \quad s = 1, 2, \dots, R-1 \quad (35b)$$

$$\mathbf{d}(\mathbf{z}^{(r_R)}, \mathbf{z}^{(n)}) \leq d_\ell \quad (35c)$$

However, (35) implies that the measurements $\{\mathbf{z}^{(m)}, \mathbf{z}^{(r_1)}, \mathbf{z}^{(r_2)}, \dots, \mathbf{z}^{(r_R)}, \mathbf{z}^{(n)}\}$ should all be in the same cell. For p_j , this is the cell $W_{m_j}^j \ni \mathbf{z}^{(m)}$, which contradicts the fact that $\mathbf{z}^{(n)} \notin W_{m_j}^j$. Thus, the initial assumption that there are two different partitions satisfying the conditions of the theorem must be wrong. The proof is complete. \square

A.2 Variable Probability of Detection for the Laser Sensor

The variable probability of detection function reduces p_D behind (i.e. at larger range from the sensor) each component of the PHD.

For a given point \mathbf{x} in the surveillance area, the probability of detection $p_D(\mathbf{x})$ is computed as

$$p_D(\mathbf{x}) = \max(p_{D,\min}, p_D^v) \quad (36a)$$

$$p_D^v = p_{D,0} - \sum_{i:r>r^{(i)}} w^{(i)} \sqrt{\frac{\sigma_s}{\bar{\sigma}_{\varphi^{(i)}}}} \exp\left(\frac{-(\varphi - \varphi^{(i)})^2}{2\bar{\sigma}_{\varphi^{(i)}}}\right) \quad (36b)$$

where

- $p_{D,\min}$ is the minimum probability of detection value a target can have;
- $p_{D,0}$ is the nominal probability of detection of the targets when they are not occluded;
- r and φ are the range and bearing, respectively, from the sensor to the point \mathbf{x} ;

- $r^{(i)}$ and $\varphi^{(i)}$ are the range and bearing, respectively, from the sensor to the i th Gaussian component;
- $w^{(i)}$ is the weight of the i th component;
- $\bar{\sigma}_{\varphi^{(i)}}$ is defined as

$$\bar{\sigma}_{\varphi^{(i)}} \triangleq \begin{cases} \sigma_{\max}, & \text{if } \sigma_{\varphi^{(i)}} > \sigma_{\max} \\ \sigma_{\min}, & \text{if } \sigma_{\varphi^{(i)}} < \sigma_{\min} \\ \sigma_{\varphi^{(i)}}, & \text{otherwise} \end{cases} \quad (37)$$

where $\sigma_{\varphi^{(i)}}$ is the bearing standard deviation of the i th component given as

$$\sigma_{\varphi^{(i)}} \triangleq \sqrt{\mathbf{u}_{\varphi^{(i)}}^T \mathbf{P}_p^{(i)} \mathbf{u}_{\varphi^{(i)}}}. \quad (38)$$

Here, $\mathbf{P}_p^{(i)}$ is the position covariance of the i th component and $\mathbf{u}_{\varphi^{(i)}}$ is the unit vector orthogonal to the range direction from the i th component to the sensor.

- The constant term σ_s is used to scale the bearing standard deviation.

Intuitively, the operation of (36b) is to reduce the nominal probability of detection at a point. The reduction depends on the weights, means and standard deviations of the components of the last estimated PHD. Reductions are only performed for the components that have smaller range values than the range of the point, and the angular proximity of the point and the components is taken into account through the exponential function in (36b).

Bibliography

- D. Arthur and S. Vassilvitskii. *k-means++: The Advantages of Careful Seeding*. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA, January 2007.
- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- M. Baum and U. D. Hanebeck. Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 338–345, Chicago, IL, USA, July 2011.
- M. Baum, M. Feldmann, D. Fränken, U. D. Hanebeck, and J. W. Koch. Extended object and group tracking: A comparison of random matrices and random hypersurface models. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Leipzig, Germany, October 2010a.
- M. Baum, B. Noack, and U. D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010b.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. A track before detect algorithm for tracking extended targets. *IEE Proceedings Radar, Sonar and Navigation*, 153(4):345–351, August 2006.
- D.E. Clark, K. Panta, and B.-N. Vo. The GM-PHD Filter Multiple Target Tracker. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Florence, Italy, July 2006.
- O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11):4232–4246, November 2009.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings Radar, Sonar and Navigation*, 152(5):364–371, October 2005.
- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended

- target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning : Data mining, inference, and prediction*. Springer, New York, 2 edition, 2009.
- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- J. W. Koch and M. Feldmann. Cluster tracking under kinematical constraints using random matrices. *Robotics and Autonomous Systems*, 57(3):296–309, March 2009.
- C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011.
- R. P. S. Mahler. Multitarget Bayes filtering via first-order multi target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.
- R. P. S. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007a.
- R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007b.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- U. Orguner, C. Lundquist, and K. Granström. Extended Target Tracking with a Cardinalized Probability Hypothesis Density Filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 65–72, Chicago, IL, USA, July 2011.

- R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 165–174, Berkeley, CA, USA, October 2006.
- K. Panta, D.E. Clark, and Ba-Ngu Vo. Data association and track management for the Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1003–1016, July 2009.
- X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking: Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, October 2003.
- G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, May 1964.
- D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):389–395, December 2003.
- D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, August 2008.
- L. Sorber. k-means++, 2011. URL [<http://www.mathworks.com/matlabcentral/fileexchange/28804-k-means++>]. Accessed 30-August-2011.
- M. Ulmke, O. Erdinc, and P. Willett. Gaussian Mixture Cardinalized PHD Filter for Ground Moving Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 1–8, Quebec City, Canada, July 2007.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- W. Wieneke and J. W. Koch. Probabilistic tracking of multiple extended targets using random matrices. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, Orlando, FL, USA, April 2010.
- H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 73–78, Chicago, IL, USA, July 2011.

Paper C

Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements

Authors: Karl Granström, Christian Lundquist, and Umut Orguner

Edited version of the paper:

K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.

Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements

Karl Granström*, Christian Lundquist*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
{karl,lundquist}@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

This paper considers tracking of extended targets using data from laser range sensors. Two types of extended target shapes are considered, rectangular and elliptical, and a method to compute predicted measurements and corresponding innovation covariances is suggested. The proposed method can easily be integrated into any tracking framework that relies on the use of an extended Kalman filter. Here, it is used together with a recently proposed Gaussian mixture probability hypothesis density (GM-PHD) filter for extended target tracking, which enables estimation of not only position, orientation, and size of the extended targets, but also estimation of extended target type (i.e. rectangular or elliptical). In both simulations and experiments using laser data, the versatility of the proposed tracking framework is shown. In addition, a simple measure to evaluate the extended target tracking results is suggested.

1 Introduction

Target tracking is the problem of estimating the states of an unknown number of targets using noisy and cluttered sets of measurements. In many typical target tracking scenarios the point target assumption is made, meaning that it is assumed that each target generates at most one measurement per time step. In recent years, tracking of extended targets have received increasing research attention. Here, extended target is defined as a target that potentially gives rise to more than one measurement per time step. Multiple measurements per target and time step enables the target tracking framework to not only estimate the location of each target, but also its spatial size and shape.

Gilholm and Salmond (2005) presented an approach for tracking extended targets under the assumption that the number of received target measurements in each time step is Poisson distributed. They show an example where they track

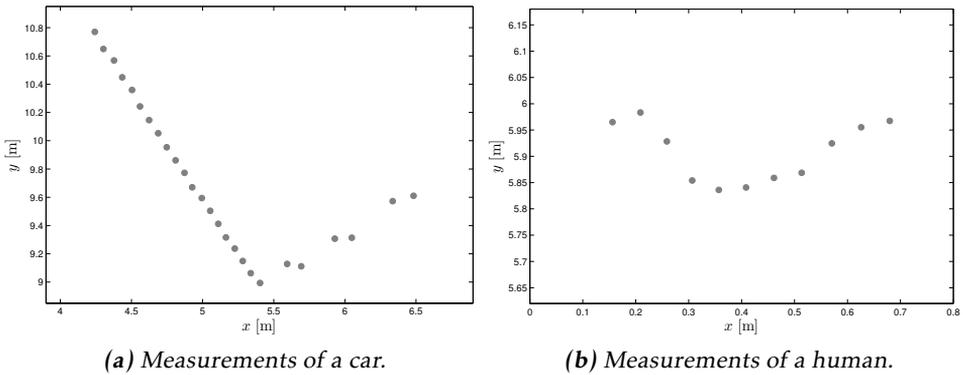


Figure 1: Motivating examples: measurements acquired outdoors with a laser range sensor. In (a) the measurements are approximately rectangular, and in (b) the measurements are approximately elliptical.

point targets which may generate more than one measurement, and an example where they track objects that have a 1-D extension (infinitely thin stick of length l). In (Gilholm et al., 2005) a measurement model was suggested which is an inhomogeneous Poisson point process. At each time step, a Poisson distributed random number of measurements are generated, distributed around the target. This measurement model can be understood to imply that the extended target is sufficiently far away from the sensor for its measurements to resemble a cluster of points, rather than a geometrically structured ensemble. A similar approach is taken in (Boers et al., 2006), where track-before-detect theory is used to track a point target with a 1-D extent.

In (Baum et al., 2010b) the authors model extended objects using an elliptic random hypersurface model (Baum and Hanebeck, 2009). Extended targets are modeled as having different measurement sources located on the target, and an ellipse is estimated that fits around the measurement sources. However, multiple extended target tracking is not treated. A Bayesian framework for estimating the location and radius of a circle from noisy measurements of the circle circumference is derived in (Baum et al., 2010a). A likelihood is not stated directly, instead the problem is posed using an errors-in-variables model.

In this work we consider estimation of extended targets using measurements from laser range sensors. Laser range sensor typically gives measurements with a high degree of structure, see examples in Figure 1, and are therefore suitable to use for the estimation of the shape and size of extended targets. In robotics laser range sensors have been used for tracking of vehicles and persons, we briefly present some more recent work here. The typical framework contains a detection algorithm that supplies a tracking algorithm with measurements belonging to some predefined class of targets. In (Petrovskaya and Thrun, 2008), vehicles are modeled as rectangles and are tracked using a particle filter framework. In

(Spinello et al., 2010), work on detection and tracking of people and cars using a 2D laser range scanner and a camera is presented. However, there is no mention of estimation of the shape and size of the targets. Detection and tracking of pedestrians using 3D laser range data is presented in (Navarro-Serment et al., 2010), where the position and velocity of the targets are tracked. Place dependent distributions of human behavior is used to improve tracking of people in (Luber et al., 2011). The place dependency encodes locations in the surveillance area where the human targets are more likely to be located, and also captures areas where the human targets are less likely to move, e.g. crossing through walls.

Two types of extended targets are considered here, rectangular targets and elliptical targets. In computer vision tracking, rectangles and ellipses have been used as target bounding boxes, for targets of different types of shape. In comparison, here we are not concerned with estimation of target bounding boxes, but rather estimation of the shape, size and location of targets that are (approximately) rectangular or elliptical using point measurements. A similar scenario occurs in vision if feature points are extracted from the images, e.g. Harris corner points (Harris and Stephens, 1988).

Here, the target type (ellipse or rectangle) is not detected from the measurements, but instead inferred in the target tracking estimation process. Bayesian estimation of extended targets from multiple measurements requires an appropriate likelihood function for the multiple measurements a target can generate. The paper presents a framework for computing these functions using the predicted measurements and corresponding innovation covariances for an extended target measured with a laser range sensor. It is also shown that this framework can successfully be integrated into an existing framework for extended target tracking, that is based on a Gaussian mixture Probability Hypothesis Density (GM-PHD) filter (Granström et al., 2010).

The extended target tracking framework is evaluated using both simulations and experiments. In simulations, single extended targets shaped as rectangles and ellipses are tracked. The results are evaluated against the ground truth. To evaluate the estimated shape and size of the extended targets, a performance metric called Intersection Over Union (IOU) is suggested. This measure is inspired by work in the computer vision research community, where it has been used to compare shapes to each other. In an outdoor experiment, up to three humans are tracked simultaneously, and the results are visually examined and shown to be good.

The paper is organized as follows: the next section defines the state representation of the extended targets, and relates the individual states in the state vector to the rectangular and elliptical shapes, respectively. Section 3 presents extended target tracking with a GM-PHD-filter, defines the main problem considered in this paper and addresses tracking of multiple extended targets with multiple shapes. In Section 4, a detailed implementational description is given of how predicted measurements and corresponding innovation covariances are computed. Section 5 presents a measure used for performance evaluation of extended target

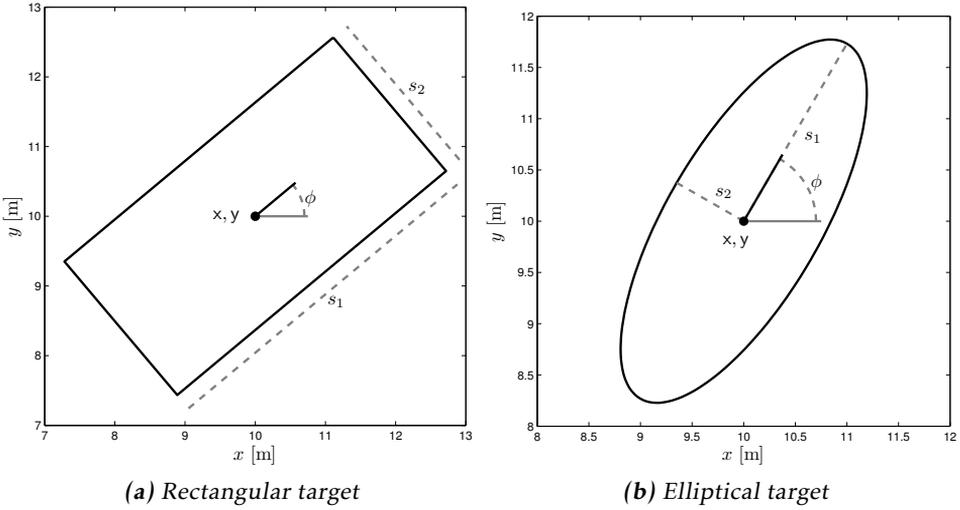


Figure 2: Relationship between target states (1) and target shape.

tracking, and Section 6 presents results from both simulations and experiments. The paper is ended with conclusions and a discussion of future work in Section 7.

2 State representation

The state vector of a general two dimensional extended target is given as

$$\mathbf{x} = [x \quad y \quad v_x \quad v_y \quad \psi \quad s_1 \quad s_2]^T, \quad (1)$$

where x, y and v_x, v_y are the Cartesian position and velocity in \mathbb{R}^2 , respectively. The parameter ψ is the orientation of the extended target shape, and s_1, s_2 represent the size of the shape. In this paper, two types of extended targets are considered, rectangular and elliptical shaped targets.

For rectangularly shaped extended targets, the two shape parameters s_1 and s_2 encode the length and width of the target, respectively, as is shown in Figure 2a. Using this particular type of shape is motivated by the fact that cars measured by laser range sensors generate measurements that are approximately rectangularly shaped, see Figure 1a.

For extended targets that are shaped like ellipsoids, the two shape parameters encode the lengths of the major and minor axis, respectively, as is shown in Figure 2b. The ellipse shape for extended targets is motivated by the fact that humans measured by laser range sensors generate measurements that are approximately elliptically shaped, see Figure 1b.

3 Extended target tracking

This section presents target tracking using GM-PHD-filters, defines the main problem addressed in the paper, and briefly considers estimation of multiple targets of multiple shapes.

3.1 GM-PHD target tracking

The aim is to estimate the state of a set of extended targets

$$\mathbf{X}_k = \left\{ \mathbf{x}_k^{(j)} \right\}_{j=1}^{N_{x,k}} \quad (2)$$

using sets of noisy, possibly cluttered, measurements

$$\mathbf{Z}_k = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}}, \quad (3)$$

for discrete time instants $k = 1, \dots, K$. In this paper, data from laser range sensors are used. Laser range sensors measure range r_i to the nearest object along rays pointing from the sensor at angles α_i . The measurements in \mathbf{Z}_k can thus be sorted counter-clockwise according to the scanning angles α_i . Note that this order of the measurements does not contain information about which measurement source caused which measurement.

The target dynamics is assumed to be modeled with a function

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \quad (4)$$

where \mathbf{u}_k is an exogenous input and \mathbf{w}_k is process noise with covariance matrix Q_k . State prediction using a dynamic motion model (4) is straightforward in target tracking, thus this part of the problem will not be addressed further in this publication.

In this work we have used the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter for extended target tracking presented in (Granström et al., 2010). If $D_{k|k-1}(\mathbf{x}|\mathbf{Z})$ is the predicted PHD-intensity, the corrected PHD-intensity is

$$D_{k|k}(\mathbf{x}|\mathbf{Z}) = L_{\mathbf{Z}_k}(\mathbf{x}) D_{k|k-1}(\mathbf{x}|\mathbf{Z}), \quad (5)$$

where the measurement pseudo-likelihood function (Mahler, 2009) is given by

$$\begin{aligned} L_{\mathbf{Z}_k}(\mathbf{x}) = & 1 - \left(1 - e^{-\gamma(\mathbf{x})}\right) p_D(\mathbf{x}) \\ & + e^{-\gamma(\mathbf{x})} p_D(\mathbf{x}) \sum_{\mathbf{p} \subseteq \mathbf{Z}_k} \omega_{\mathbf{p}} \sum_{\mathbf{W} \subseteq \mathbf{p}} \frac{\gamma(\mathbf{x})^{|\mathbf{W}|}}{d_{\mathbf{W}}} \cdot \prod_{\mathbf{z} \in \mathbf{W}} \frac{\phi_{\mathbf{z}}(\mathbf{x})}{\lambda_k c_k(\mathbf{z})}. \end{aligned} \quad (6)$$

The first part of this equation, $1 - \left(1 - e^{-\gamma(\mathbf{x})}\right) p_D(\mathbf{x})$, handles the targets for which there are no detections. The second part handles targets for which there are at least one detection.

The PHD-intensity is approximated by a Gaussian mixture as in (Vo and Ma, 2006),

$$D_k(\mathbf{x}) = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}\left(\mathbf{x}; m_k^{(i)}, P_k^{(i)}\right), \quad (7)$$

where $w_k^{(i)}$, $m_k^{(i)}$ and $P_k^{(i)}$ are the weights, mean vectors and covariance matrices of the Gaussian components, respectively.

As (6) shows, the update step contains a summation over partitions p of the measurement set \mathbf{Z}_k , and a summation over the cells W in each partition p . Let the measurements in a cell W be denoted

$$\mathbf{z}^W \triangleq \bigoplus_{\mathbf{z} \in W} \mathbf{z}, \quad (8)$$

where \bigoplus is vertical vectorial concatenation. Then, for each cell of each partition and each predicted Gaussian component,

$$e^{-\gamma(\mathbf{x})} p_D(\mathbf{x}) \omega_p \frac{\gamma(\mathbf{x})^{|W|} \mathcal{N}\left(\mathbf{z}^W; \hat{\mathbf{z}}_{k|k-1}^{W,(i)}, S_k^{W,(i)}\right)}{d_W (\lambda_k c_k(\mathbf{z}))^{|W|}} \mathcal{N}\left(\mathbf{x}; \hat{\mathbf{x}}_{k|k}^{(i)}, P_{k|k}^{(i)}\right) \quad (9)$$

represents the corresponding updated Gaussian component, where

$$\mathcal{N}\left(\mathbf{z}^W; \hat{\mathbf{z}}_{k|k-1}^i, S_k^{W,(i)}\right) \quad (10)$$

is the likelihood of the set of measurements in the cell W . The key point of being able to use the above formulas for extended targets with structured measurements (e.g., with laser sensor reports) is to calculate the predicted measurements $\hat{\mathbf{z}}_{k|k-1}^{W,(i)}$, innovation covariances $S_k^{W,(i)}$, updated means $\hat{\mathbf{x}}_{k|k}^{(j)}$ and updated covariances $P_{k|k}^{(i)}$. The calculation of these quantities must rely on a measurement model of a form similar to

$$\mathbf{z}_k^{(j)} = h(\mathbf{x}_k, \mathbf{e}_k), \quad (11)$$

where \mathbf{e}_k is measurement noise with covariance matrix $R_k^{(j)}$. Once such a model is available,

- the calculation of the measurement prediction $\hat{\mathbf{z}}_{k|k-1}^{W,(i)}$ and innovation covariance $S_k^{W,(i)}$ can be achieved using the current estimates, and
- as in the implementation presented in (Granström et al., 2010), the updated means $\hat{\mathbf{x}}_{k|k}^{(i)}$ and updated covariances $P_{k|k}^{(i)}$ can be obtained with a Kalman filter, or one of its non-linear counterparts such as the Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF).

The construction of a measurement model of the form (11) is at the root of the problem of calculating a measurement prediction which will be posed as the main problem of our work in the next subsection.

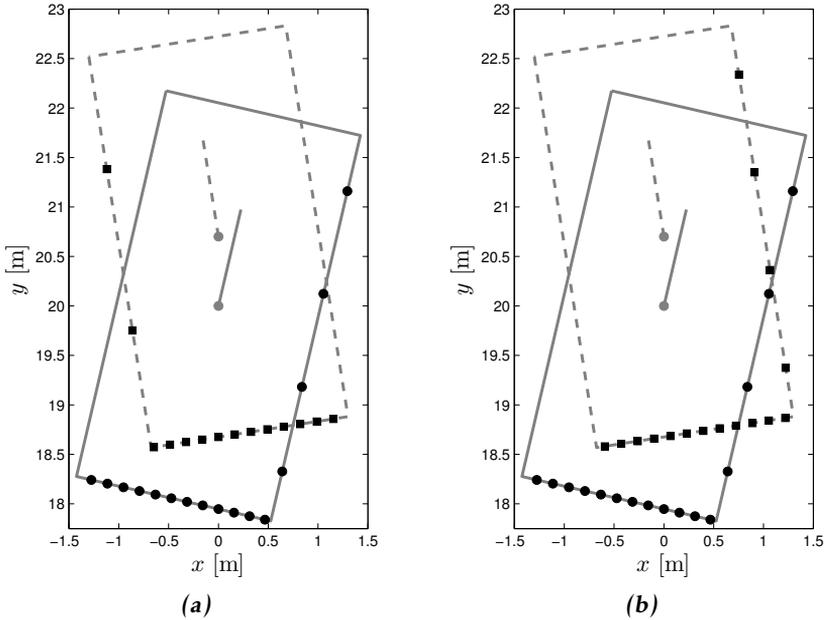


Figure 3: Comparison of estimated measurements, the sensor is located in the origin. The true extended target is shown as a solid rectangle and the set of measurements \mathbf{Z}_k are shown as filled black circles. The estimated extended target is shown as a dashed rectangle, the estimated set of measurements $\hat{\mathbf{Z}}_k$ are shown as filled black squares. In (a), using a straightforward model of the laser sensor gives a $\hat{\mathbf{Z}}_k$ which is a poor correspondence to \mathbf{Z}_k . In (b), the proposed method is used and $\hat{\mathbf{Z}}_k$ better corresponds to \mathbf{Z}_k .

3.2 Problem definition

When the extended targets are modeled as points, a measurement model can readily be constructed assuming that each measurement is close to the target's center of mass, see e.g. (Granström et al., 2010). Here however, we are concerned with tracking extended targets (1) using measurements acquired with a laser range sensor, thus the point target assumption is trivially invalid. As will be shown using an example, when the extended targets are not modeled as points but rather as geometric structures, construction of a measurement model can be significantly more complicated compared to the point target case.

Consider the plots in Figure 3. The extended target estimate is shifted in y position by 0.7m, and in heading ψ by 20° , with respect to the true extended target. Using a simple model of a laser range sensor, predicted measurements can be computed using line intersection as is shown in Figure 3a. However, due to the orientation error, the estimated target is not showing the same two sides towards the sensor as the true target, making the set of true measurements \mathbf{Z}_k fundamen-

tally different from the set of estimated measurements $\hat{\mathbf{Z}}_k$. The cardinality is different, $|\mathbf{Z}_k| = 16$ and $|\hat{\mathbf{Z}}_k| = 14$, and the measurements are located on different sides of the target shape, causing the data association between the measurements in \mathbf{Z}_k and the predicted measurements in $\hat{\mathbf{Z}}_k$ to become difficult. However, using the method proposed in this paper, described further in Section 4, a $\hat{\mathbf{Z}}_k$ that better corresponds to \mathbf{Z}_k is obtained, see Figure 3b.

The main problem addressed in this paper is thus to obtain predicted measurements and innovation covariances such that they can be integrated into an existing target tracking framework. As mentioned above, here the GM-PHD extended target tracking framework presented in (Granström et al., 2010) has been used, however any target tracking framework that relies on the EKF could have been used just as well. In order to obtain predicted measurements $\hat{\mathbf{z}}_{k|k-1}^{W,(i)}$ and corresponding innovation covariances $S_k^{W,(i)}$, information from the set of measurements in the cell \mathbf{z}^W is used, i.e. the following two approximations

$$\hat{\mathbf{z}}_{k|k-1}^{W,(i)} \approx \hat{\mathbf{z}}_{k|k-1}^{W,(i)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1}) \quad (12a)$$

$$S_k^{W,(i)} \approx S_k^{W,(i)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1}) \quad (12b)$$

are made. Thus, the predicted measurements and innovation covariances are functions of the predicted state and the set of measurements. In a sense, this is an unconventional use of an errors-in-variables framework where the measurement model would depend on the current set of measurements. In Section 4 we describe in detail the proposed method for computing $\hat{\mathbf{z}}_{k|k-1}^{W,(i)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1})$ and $S_k^{W,(i)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1})$. When these quantities are calculated, a Kalman filter (or EKF/UKF) measurement update can easily be used to obtain the updated means $\hat{\mathbf{x}}_{k|k}^{(i)}$ and updated covariances $P_{k|k}^{(i)}$.

3.3 Multiple shapes and multiple targets

An interesting aspect of extended target tracking with multiple targets and multiple shapes is how to correctly estimate the correct shape for each target. As was mentioned in the introduction, in some previous work the target tracking is preceded by a detection algorithm. Thus, a possible way to infer the type of shape is to consider the measurements and make a hard decision as to which type they represent.

In this paper, an approach that is slightly similar to track-before-detect is taken. When a new target appears, the GM-PHD-filter birth intensity is set such that one Gaussian component per target type is given birth to. Then, as the filter iterates throughout the prediction and correction steps, the weight $w_k^{(j)}$ of each Gaussian component is updated. Eventually, the weights will converge such that only one component remains, from which the target type can be found. The target types are thus, while not included in the state vectors, implicitly estimated via the weights of the Gaussian components in the PHD-intensity.

4 Computing predicted measurements and innovation covariances

In this section we present a detailed description of how predicted measurements and innovation covariances (12) are computed for extended targets (1) that are either rectangular or elliptical in shape.

4.1 Predicted measurements for rectangular targets

The first step in predicting a set of measurements is to, given the set of measurements, estimate how many sides of the measured target that are shown. For a rectangular target, it is trivial to conclude that at most two of the sides of the shape are visible to the sensor at any given moment. Given a set of measurements $\mathbf{Z} = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}}$, where each measurement is a vector $\mathbf{z}_k^{(j)} = \left[z_k^{(j),1}, \dots, z_k^{(j),n_z} \right]^T$, let $C = [c_{m,n}]$ be the sample covariance of the measurements with entries

$$c_{m,n} = \frac{1}{N_{z,k} - 1} \sum_{j=1}^{N_{z,k}} \left(z_k^{(j),m} - \bar{z}_k^m \right) \left(z_k^{(j),n} - \bar{z}_k^n \right) \quad (13)$$

where \bar{z}_k^m is the mean of the m :th component of the measurement vectors $\mathbf{z}_k^{(j)}$. For laser range measurements in 2D we have $n_z = 2$.

Further, let e_1 and e_2 be the two eigenvalues of the covariance matrix C , where $e_2 > e_1$. In the noiseless case, measurements of just one side of a rectangle will have 0 standard deviation along the direction perpendicular to the measured line, and the corresponding eigenvalue will be 0. An estimate of the number of sides N that generated the set of measurements can then be obtained as

$$N = \begin{cases} 1 & \text{if } \frac{e_2}{e_1} \geq K \\ 2 & \text{otherwise} \end{cases} \quad (14)$$

where K is a threshold. Empirically $K = 25$ was determined to be an appropriate value. Given the measurements in Figure 4a, the eigenvalues of the corresponding sample covariance matrix are $e_1 = 0.1198$ and $e_2 = 0.6629$, and the eigenvalue quota 5.5327 is less than the threshold K . Note that the measurements are a subset of the measurements in Figure 1a. A subset is used to prevent Figures 4a and 4b from being too cluttered.

If the estimated number of sides is two, the measurement closest to the corner connecting the two sides needs to be found. This is performed using the function Corner Index given in Table 1, where the distance d from a point \mathbf{z}_3 to a line defined by two points \mathbf{z}_1 and \mathbf{z}_2 is given by the function

$$d = \text{point2lineDist}(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \frac{\left| (\mathbf{z}_2^x - \mathbf{z}_1^x)(\mathbf{z}_1^y - \mathbf{z}_3^y) - (\mathbf{z}_1^x - \mathbf{z}_3^x)(\mathbf{z}_2^y - \mathbf{z}_1^y) \right|}{\sqrt{(\mathbf{z}_2^x - \mathbf{z}_1^x)^2 + (\mathbf{z}_2^y - \mathbf{z}_1^y)^2}}. \quad (15)$$

Table 1: Corner Index

1:	Input: Set of measurements $\mathbf{Z} = \{\mathbf{z}^{(j)}\}_{j=1}^{N_z}$, sorted counter clockwise according to scanning angle.
2:	Initialize: Minimum distance $d_{\min} = \infty$, first and last points $\mathbf{z}_1 = \mathbf{z}^{(1)}$, $\mathbf{z}_{N_z} = \mathbf{z}^{(N_z)}$.
3:	for $n = 2, \dots, N_z - 1$ do
4:	Let $\mathbf{z}_n = \mathbf{z}^{(n)}$ be the current point.
5:	Initialize sum of distances $d = 0$.
6:	for $k = 2, \dots, n - 1, n + 1, \dots, N_z - 1$ do
7:	$d = d + \begin{cases} \text{point2lineDist}(\mathbf{z}_1, \mathbf{z}_n, \mathbf{z}_k) & \text{if } k < n \\ \text{point2lineDist}(\mathbf{z}_n, \mathbf{z}_{N_z}, \mathbf{z}_k) & \text{if } k > n \end{cases}$
8:	end for
9:	if $d < d_{\min}$ then
10:	Set $\hat{n} = n$ and $d_{\min} = d$
11:	end if
12:	end for
13:	Output: Index to measurement closest to corner \hat{n} .

For the measurements in Figure 4a, the corner is identified as the point located in $x = 5.40, y = 8.99$. Given the identified corner, it is straightforward to compute the number of measurements belonging to each of the two sides that are seen by the sensor. In the example given in Figure 4a, there are $m_1 = 3$ measurements on one side, and $m_2 = 6$ on the other. If only one side is measured, i.e. if $\frac{e_2}{e_1} \geq K$, then trivially all measurements belong to the side that was measured by the sensor.

Assuming that two sides are shown by the set of measurements \mathbf{Z} , the set can be divided into two subsets \mathbf{Z}^1 and \mathbf{Z}^2 corresponding to the measurements that are from the two sides, i.e. $|\mathbf{Z}^1| = m_1$ and $|\mathbf{Z}^2| = m_2$. Let β_1 and β_2 be the angles of the vectors defined by the first and last point from \mathbf{Z}^1 and \mathbf{Z}^2 , respectively. Then, $\beta_i^N = \beta_i + \pi/2$ are the angles of the corresponding normal vectors. Further, given an estimated extended target state $\hat{\mathbf{x}}$, let η_1, \dots, η_4 be the surface normals of the four sides of the rectangle.

The sets \mathbf{Z}^1 and \mathbf{Z}^2 can now be associated to one of the four sides of the rectangle by finding the two sides for which $|\eta_i - \beta_1^N|$ and $|\eta_j - \beta_2^N|$ are minimized. Predicted measurements are generated for the sides that are in view by distributing m_1 and m_2 points uniformly on the two sides. An example is given in Figure 4a.

4.2 Predicted measurements for elliptical targets

Given an angle at which the sensor measures, finding the intersection between the measurement ray and an ellipse is performed as follows. Let the ellipse be given by the position x, y , orientation ψ and lengths of the major and minor axis

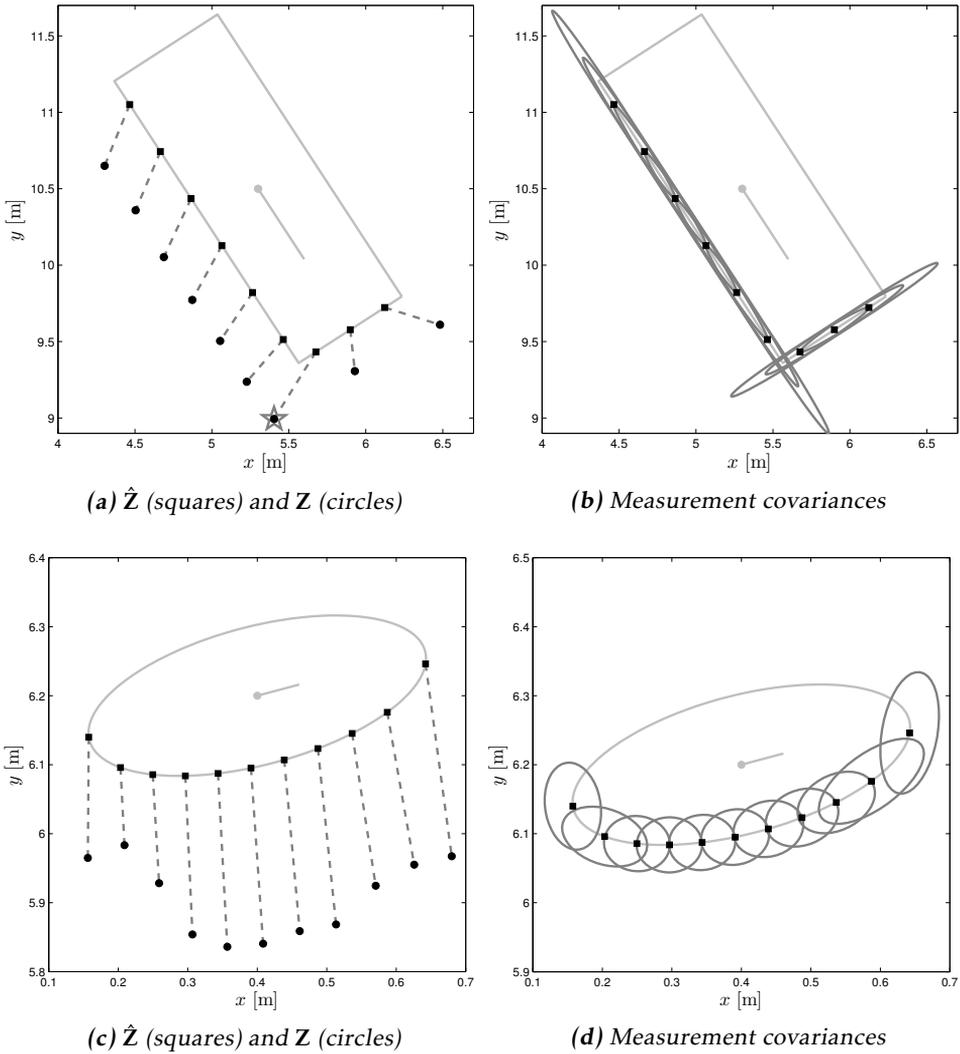


Figure 4: The rectangular and elliptical targets. (a) and (c): Example of measurements of a car and human (filled black circles), respectively, and predicted measurements located on target surface (filled black squares). The corresponding associations are shown with dashed lines. The measurement identified as being closest to the corner in the rectangular case is shown with a gray star. (b) and (d): The corresponding measurement covariances.

s_1 and s_2 , as in (1). Further, let the range measurement r from the sensor to the target surface be measured at an angle α . Thus, the intersection defines a point

in Cartesian coordinates

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} r \cos(\alpha) \\ r \sin(\alpha) \end{bmatrix}, \quad (16)$$

given in the coordinate frame defined with the sensor position as origin. The same point can be described in the coordinate frame defined by the position and heading of the ellipse, using the appropriate coordinate frame transformation. The intersection point's coordinates are now given by

$$\begin{aligned} \begin{bmatrix} x_r^e \\ y_r^e \end{bmatrix} &= (R_{-\alpha})^{-1} \begin{bmatrix} x_r - x \\ y_r - y \end{bmatrix} \\ &= \begin{bmatrix} r(c_\alpha c_\psi + s_\alpha s_\psi) - xc_\psi - ys_\psi \\ r(-c_\alpha s_\psi + s_\alpha c_\psi) + xs_\psi - yc_\psi \end{bmatrix} = \begin{bmatrix} r\theta_1 + \theta_2 \\ r\theta_3 + \theta_4 \end{bmatrix} \end{aligned} \quad (17)$$

where R_α is the rotation matrix for an angle α and e denotes the change of reference frame. This point must satisfy the ellipse equation

$$\frac{(x_r^e)^2}{s_1^2} + \frac{(y_r^e)^2}{s_2^2} = 1. \quad (18)$$

Inserting (17) into (18) gives

$$\begin{aligned} 1 &= \frac{(r\theta_1 + \theta_2)^2}{s_1^2} + \frac{(r\theta_3 + \theta_4)^2}{s_2^2} \\ &= \left(\frac{\theta_1^2}{s_1^2} + \frac{\theta_3^2}{s_2^2} \right) r^2 + 2 \left(\frac{\theta_1\theta_2}{s_1^2} + \frac{\theta_3\theta_4}{s_2^2} \right) r + \frac{\theta_2^2}{s_1^2} + \frac{\theta_4^2}{s_2^2} \\ &= Ar^2 + 2Br + C \end{aligned} \quad (19)$$

which has the two solutions $r = -\frac{B}{A} \pm \sqrt{\frac{B^2}{A^2} - \frac{C-1}{A}}$. Since the sensor measures the closest intersection with the target, the correct range r at a given angle α is $r = -\frac{B}{A} - \sqrt{\frac{B^2}{A^2} - \frac{C-1}{A}}$. Note that if no part of the extended target is located along the measurement ray defined by the angle α , r will be a complex number.

For elliptically shaped extended targets, the first step in computing a set of predicted measurements is to find the angles α_1 and α_2 within which the sensor can measure the estimated target. Next, $|\mathbf{Z}|$ estimated measurements are generated on the estimated target surface uniformly spaced in the angle dimension between α_1 and α_2 . An example of predicted measurements is given in Figure 4c.

4.3 Innovation covariances

To compute the innovation covariances

$$S_k^{W,(j)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1}), \quad (20)$$

the measurement model Jacobian H_k and the measurement covariances $R_k^{(j)}$ are needed.

As we have not derived an explicit mathematical function for the measurement model, the measurement model Jacobian

$$H_k = \left. \frac{dh}{d\mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}, \quad (21)$$

is computed numerically instead of derived analytically. By making a small permutation ε to the n :th element of the predicted state vector, the n :th column of the measurement Jacobian is approximated as

$$\frac{\hat{\mathbf{z}}_{k|k-1}^{W,(j)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1}^{\varepsilon,n}) - \hat{\mathbf{z}}_{k|k-1}^{W,(j)}(\mathbf{z}^W, \hat{\mathbf{x}}_{k|k-1})}{\varepsilon} \quad (22)$$

where $\hat{\mathbf{x}}_{k|k-1}^{\varepsilon,n}$ is the predicted state after the n :th element is permuted by adding ε . Note that in doing so, we only consider permutations in the predicted state when the Jacobian is approximated numerically, i.e. we do not consider permutations in the set of measurements \mathbf{z}^W . Considering changes in \mathbf{z}^W is a topic for future work.

Instead of modeling the covariances according to the sensor statistics, the covariances are constructed such that the uncertainty of all predicted measurements follow the surface of the target. One of the axes of the corresponding uncertainty ellipse is aligned with the surface, and the size of the uncertainty in this direction is set to the distance to the nearest measurement point. The size of the uncertainty in the direction perpendicular to the surface is set to a constant σ_r .

With multiple measurements on the target surface, this gives measurement covariance ellipses that are aligned to the surface tangent, thus giving a combined uncertainty that covers the part of the target surface that was measured by the sensor. Examples of measurement covariances for the measurements given in Figures 4a and 4c are shown in Figures 4b and 4d.

5 Extended target tracking performance evaluation

In this section we address evaluating the estimated shape of the extended target. As the velocity of the extended target does not affect its shape and size, the two velocity states v_x and v_y are excluded here for the sake of simplicity. For a rectangular or elliptical extended target with true state $\mathbf{x}^0 = [x^0 \ y^0 \ \psi^0 \ s_1^0 \ s_2^0]^T$, there are four alternative estimates that give an identical shape in the state space:

$$\hat{\mathbf{x}}^1 = [x^0 \ y^0 \ \psi^0 \ s_1^0 \ s_2^0]^T, \quad (23a)$$

$$\hat{\mathbf{x}}^2 = [x^0 \ y^0 \ \psi^0 + \pi \ s_1^0 \ s_2^0]^T, \quad (23b)$$

$$\hat{\mathbf{x}}^3 = [x^0 \ y^0 \ \psi^0 + \frac{\pi}{2} \ s_2^0 \ s_1^0]^T, \quad (23c)$$

$$\hat{\mathbf{x}}^4 = [x^0 \ y^0 \ \psi^0 - \frac{\pi}{2} \ s_2^0 \ s_1^0]^T. \quad (23d)$$

For example, let the orientation of a rectangular target be $\psi = 0\text{rad}$ and let the lengths of the two sides be $s_1 = 4\text{m}$ and $s_2 = 2\text{m}$. The estimation errors of the corresponding estimated states may be as large as $\frac{\pi}{2}\text{rad}$, -2m and 2m , despite the fact that if visualized, the estimated shape and size of the extended target is identical to the true one. Thus, only considering the estimation errors of the extended target tracking results may give a false picture of the quality of the extended target tracking results.

In this paper, we evaluate the extended target tracking results by considering the estimated x, y -position, and by considering a measure called Intersection-Over-Union (IOU). Let \hat{A} be the area of the estimated extended target, and let A_0 be the area of the true target. By computing the area of the intersection between the estimate and the true target, and dividing by the union of the two areas,

$$\frac{\hat{A} \cap A_0}{\hat{A} \cup A_0} \in [0 \ 1] \quad (24)$$

a measure is obtained, where 1 represents a perfect overlap of the estimated and true extended target, and 0 represents that there is no overlap at all. The IOU-measure captures differences in x, y -position, in shape orientation ψ and in the shape size parameters s_1 and s_2 . It should be noted though that the IOU-measure does not consider any difference in target type, thus the measure could very well be close to one despite the fact that the true target is rectangular and the estimated target is elliptical, or vice versa.

The estimated target type is evaluate by considering the sum of the Gaussian components' weights $w_k^{(j)}$ for each type. Assuming that there is only one rectangular target present, the weights for the Gaussian components representing rectangular targets should sum to one, and the weights for the components representing elliptical targets should sum to zero.

6 Results

This section presents results from simulations and experiments using the presented work.

6.1 Simulations

A number of different simulations were performed in order to assess the estimation results of the extended target tracking filter. Two simulated trajectories were used, one linear motion, shown in Figure 5a, and a combination of linear and curved motion, shown in Figure 5b.

Linear motion

In the first simulation a rectangularly shaped extended target moves from right to left through the surveillance area. The orientation of the target is 0rad and the length and width is $s_1 = 5\text{m}$ and $s_2 = 2.5\text{m}$, respectively. The estimation results

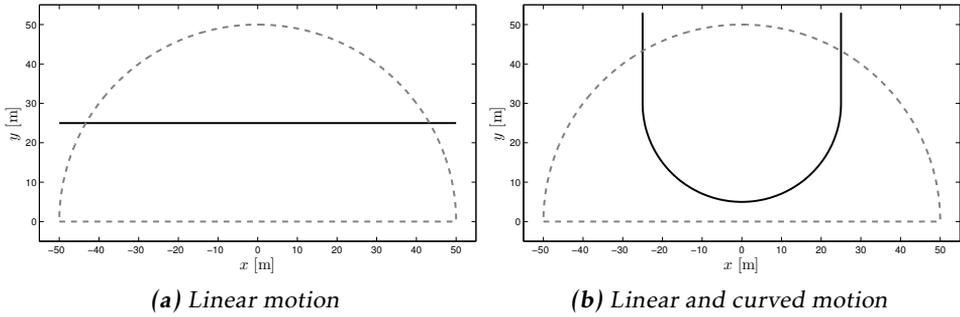


Figure 5: Trajectories used in simulations. The target trajectories are showed in black. The sensor is located in the origin, the surveillance area boundary is showed with a dashed gray line.

are shown in Figure 6a. A similar simulation was performed using an elliptical extended target. The motion is again from right to left, and the orientation is 0rad. The lengths of the major and minor axes are $s_1 = 2.5\text{m}$ and $s_2 = 1.25\text{m}$, respectively. The estimation results are shown in Figure 6b.

Linear and curved motion

In this simulation the motion of the extended target was a combination of linear and curved motion. A rectangular target with length and width $s_1 = 5\text{m}$ and $s_2 = 2.5\text{m}$ was simulated first, the estimation results are shown in Figure 6c. In a similar simulation, an elliptical target with lengths of the major and minor axes $s_1 = 2.5\text{m}$ and $s_2 = 1.25\text{m}$, respectively, was simulated. The estimation results are shown in Figure 6d.

Comments

Under linear motion, the rectangular target is estimated with high accuracy, while the elliptical target is slightly underestimated in size (s_1 and s_2 are underestimated). Under linear and curved motion the problem is slightly more complicated for the rectangular target as it moves close to the sensor, however the overall results are good. For the elliptical target, the size is underestimated again. Estimation of target type (rectangular or elliptical) is shown via the Gaussian component weights in the bottom plots in each figure. As is shown, the filter quickly converges to the correct target type.

6.2 Experiment

The suggested framework for extended target tracking was tested in an experiment using laser range data. The data set used contains 600 range scans acquired in an outdoor environment, with five persons moving through the surveillance area (at most three persons simultaneously). The first person enters the surveillance area at time 22 and moves to the center where he remains still for the re-

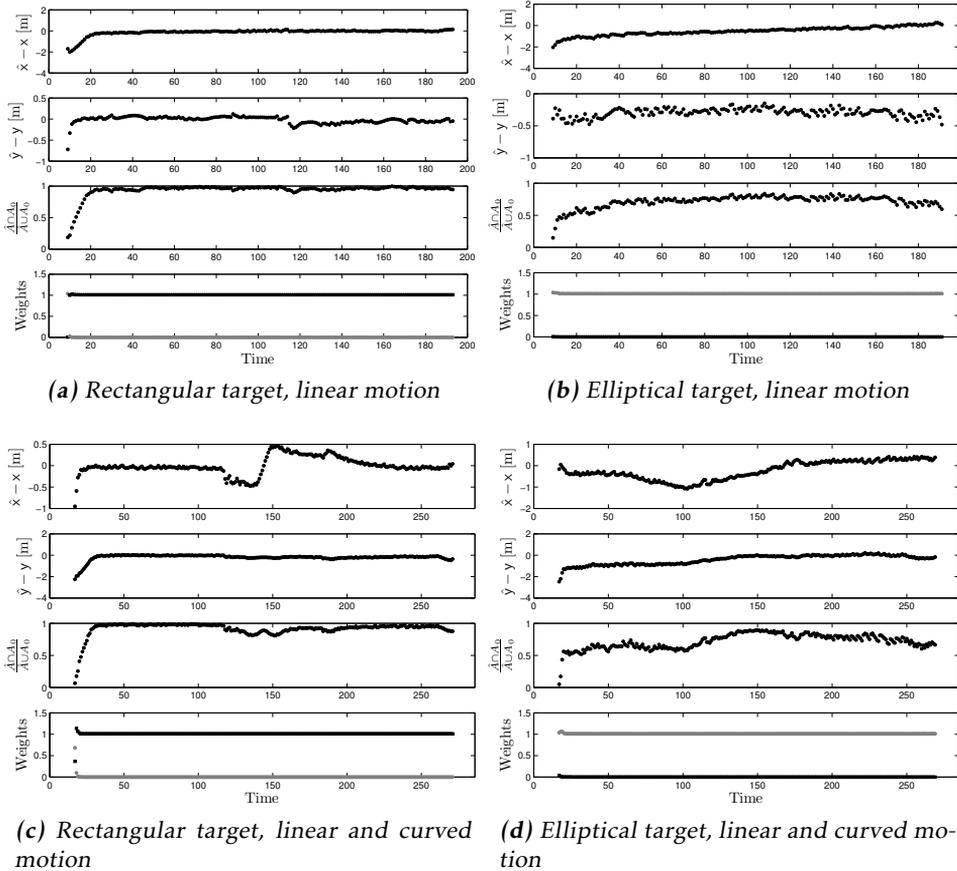


Figure 6: Simulation results: the top row shows linear motion (Figure 5a), the bottom row linear and curved motion (Figure 5b). Each figure shows the estimation error in x and y position, the IOU measure and the sum of the weights for each target type (black is rectangles, gray is ellipses).

mainder of the data. The second person enters at time 38, and proceeds to move behind the first person, both entering and exiting an occluded part of the surveillance area. Remaining three persons enters and exits the surveillance area at later times during the experiment. The results from the experiment are shown in Figures 7 and 8.

Since the second person moves through parts of the surveillance area that are occluded by the first person, the results show target loss (i.e. cardinality error) at three time steps. Using a variable probability of detection, this problem can be overcome. However we have chosen to not include this, since the space constraints do not allow a description of the method used to compute the variable probability of detection.

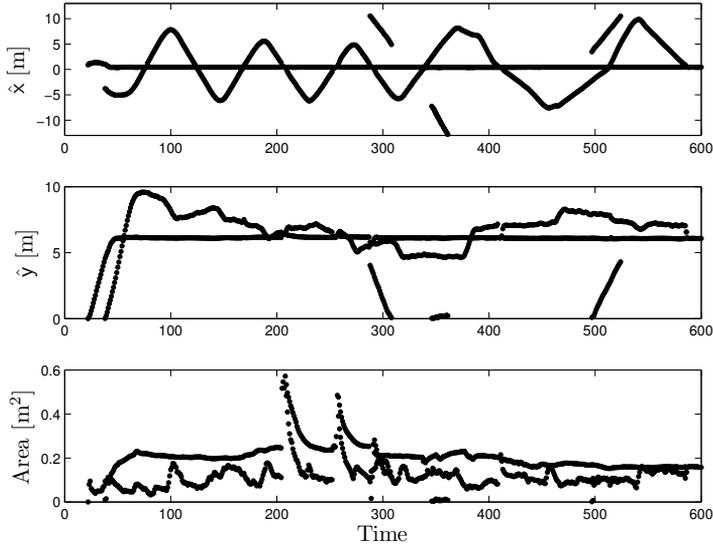


Figure 7: Experiment results, showing the estimated x and y position, as well as the area of each extended target.

As there is no ground truth, it is difficult to evaluate the quality of the estimated extended target, including the shape parameters ψ , s_1 and s_2 . In this paper we have chosen to compute the area of each extended target, since this area can be compared to a rough estimate of the area of a cross section of the human torso, under the assumption that it is elliptically shaped. Under the assumption that an average person is roughly 50cm to 60cm wide (torso and arms) and 25cm to 30cm deep, the average area can be said to be somewhere between 0.1m^2 and 0.15m^2 . Comparing these values to the estimated areas (see Figure 7) show that the ellipses representing the persons have estimated areas of reasonable size, with the exception of two instances in time between time 200 and time 300. These two times correspond to time when two targets are spatially very close, and are thus merged into just one target, producing a considerably larger target. The person standing still shows a stable estimated area, the persons that are moving throughout the whole experiment shows much more changes in the estimated area.

7 Conclusions and future work

In this paper we presented a method to compute predicted measurements and corresponding innovation covariances when rectangular and elliptical extended targets are measured by laser range sensors. The method can easily be inserted to an existing extended target tracking GM-PHD-filter, enabling efficient estimation of the extended target's location, orientation and size, as well as estimation of the extended target type.

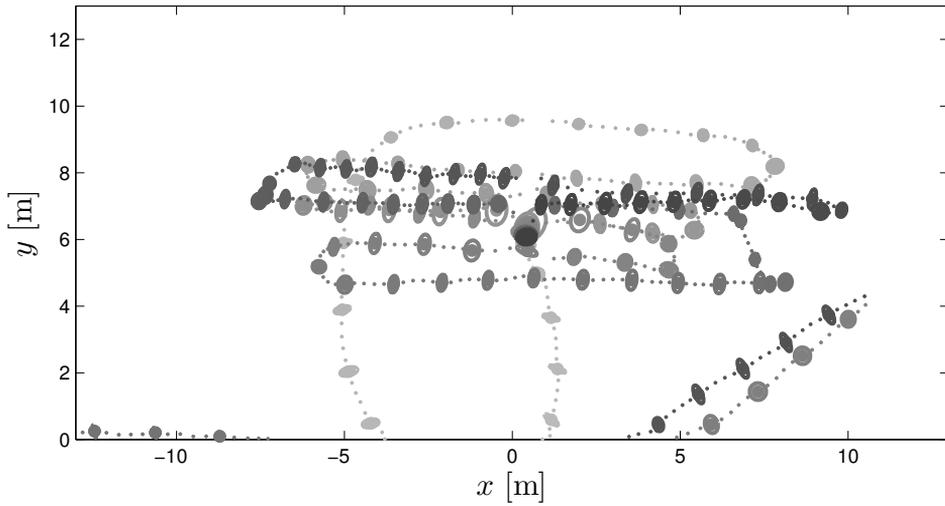


Figure 8: Experiment results, showing the trajectories of each of the two targets. A grayscale is used to highlight the different time steps, and for every 5:th time step the corresponding shape and size of each target is plotted.

In future work, we plan to investigate the reasons behind the underestimation of target size for elliptical targets. Furthermore, the presented work needs to be integrated with the variable probability of detection such that targets can be tracked while they are occluded by other targets. The target tracking framework also needs to be tested in experiments with laser range data that contains measurements of both rectangular and elliptical targets, in order to test the estimation of target type further.

Bibliography

- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- M. Baum, V. Klumpp, and U. D. Hanebeck. A Novel Bayesian Method for Fitting a Circle to Noisy Points. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010a.
- M. Baum, B. Noack, and U. D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010b.
- Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. A track before detect algorithm for tracking extended targets. *IEE Proceedings Radar, Sonar and Navigation*, 153(4):345–351, August 2006.
- K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings Radar, Sonar and Navigation*, 152(5):364–371, October 2005.
- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Alvey Vision Conference*, pages 147–152, Manchester, UK, September 1988.
- M. Luber, G. D. Tipaldi, and K. O. Arras. Place-dependent people tracking. *International Journal for Robotics Research*, 30(3):280–293, January 2011.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian Detection and Tracking Using Three-dimensional LADAR Data. *International Journal for Robotics Research*, 29(12):1516–1528, October 2010.

- A. Petrovskaya and S. Thrun. Model Based Vehicle Tracking for Autonomous Driving in Urban Environments. In *Proceedings of Robotics: Science and Systems*, Zürich, Switzerland, June 2008.
- L. Spinello, R. Triebel, and R. Siegwart. Multiclass Multimodal Detection and Tracking in Urban Environments. *International Journal for Robotics Research*, 29(12):1498–1515, October 2010.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

Paper D

A PHD filter for tracking multiple extended targets using random matrices

Authors: Karl Granström, and Umut Orguner

Edited version of the paper:

K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.

A PHD filter for tracking multiple extended targets using random matrices

Karl Granström*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
karl@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

This paper presents a random set based approach to tracking of an unknown number of extended targets, in the presence of clutter measurements and missed detections, where the targets' extensions are modeled as random matrices. For this purpose, the random matrix framework developed recently by Koch *et al.* is adapted into the extended target PHD framework, resulting in the Gaussian inverse Wishart PHD (GIWPHD) filter. A suitable multiple target likelihood is derived, and the main filter recursion is presented along with the necessary assumptions and approximations. The particularly challenging case of close extended targets is addressed with practical measurement clustering algorithms. The capabilities and limitations of the resulting extended target tracking framework are illustrated both in simulations and in experiments based on laser scans.

1 Introduction

Early target tracking often made the assumption that each target can produce at most one measurement at a given time step, see e.g. (Bar-Shalom and Fortmann, 1987). With modern and more accurate sensors, the targets may occupy multiple resolution cells of the sensor, thus potentially producing more than one measurement at a given time step. Such targets are denoted extended, and tracking of extended targets has received increasing research attention over the past decade. Examples of extended target tracking include vehicle tracking using automotive radar, tracking of sufficiently close airplanes or ships with ground or marine radar stations, and person tracking using laser range sensors.

Assuming that the received target measurements are Poisson distributed in number, Gilholm and Salmond presented an approach to extended target tracking (Gilholm and Salmond, 2005). Their approach is illustrated with two examples, one in which the target is modeled as a point that may generate more than one

measurement, and another example in which the target is an infinitely thin stick of length l . An inhomogeneous Poisson point process measurement model is suggested in (Gilholm et al., 2005), where a Poisson distributed number of measurements is distributed around the target. The model implies that the target is sufficiently far away from the sensor for the measurements to resemble a cluster rather than a geometric structure.

Another approach to extended target modeling is the random hypersurface model (Baum and Hanebeck, 2009), which has been used to estimate elliptic targets (Baum et al., 2010b). Measurements of target down-range extent are used to aid track retention in (Salmond and Parr, 2003). Further approaches to estimating the target extensions, as ellipses, rectangles, or more general shapes, are given in (Granström et al., 2011; Lundquist et al., 2011; Baum and Hanebeck, 2011; Zhu et al., 2011).

With finite set statistics (FISST), Mahler introduced a set theoretic approach in which targets and measurements are modeled using random finite sets (RFS). The approach allows multiple target tracking in the presence of clutter and with uncertain associations to be cast in a Bayesian framework (Mahler, 2007), resulting in an optimal multi-target Bayes filter. An important contribution of FISST is the statistical moments of the RFS, which enable practical implementation of the optimal multi-target Bayes filter. The first order moment of an RFS is called the *probability hypothesis density* (PHD), and is an intensity function defined over the target state space. The PHD filter propagates the target set's PHD in time (Mahler, 2003, 2007), and represents an approximation to the optimal multi-target Bayes filter. By approximating the PHD with a Gaussian mixture (GM), a practical implementation of the PHD filter is obtained, called the Gaussian mixture PHD (GM-PHD) filter (Vo and Ma, 2006). An extension of the PHD filter to handle extended targets of the type presented in (Gilholm et al., 2005) is given in (Mahler, 2009). For the closely related area of group target tracking, in which several targets move in unison, an approach using the Gaussian mixture PHD filter, where groups are identified as targets with similar position or velocity estimates, is presented in (Clark and Godsill, 2007). The individual targets in a group are predicted together using a leader-follower model. A random finite set formulation of single extended target tracking is given in (Vo et al., 2008), a particle implementation is given for the general case and a closed form solution is shown for the linear Gaussian case.

A Gaussian mixture implementation of the extended target PHD filter (Mahler, 2009), called the ETGMPHD-filter, has been presented in (Granström et al., 2012), with an early version given in (Granström et al., 2010). In both of the works (Granström et al., 2012) and (Granström et al., 2010), only the kinematic properties of the targets' centroids are estimated. Estimating the targets' extents is omitted to reduce the complexity of the presentation, however this also leads to some drawbacks. In this paper, the case where the target extents are explicitly modeled and estimated along with the kinematic target states is investigated. For this purpose, we give an extended target PHD filter implementation where the target

extents are represented with symmetric positive definite random matrices, i.e. the extensions are elliptical.

Using random matrices to track extended objects and groups of targets was suggested by Koch in 2008 (Koch, 2008). The target kinematical states are modeled using a Gaussian distribution, while the target extension is modeled using an inverse Wishart distribution. Using random matrices to track group targets under kinematical constraints is discussed in (Koch and Feldmann, 2009). Modifications and improvements to the Gaussian-inverse Wishart model of (Koch, 2008) have been suggested in (Feldmann et al., 2011), and the model (Koch, 2008) has also been integrated into a Probabilistic Multi-Hypothesis Tracking (PMHT) framework in (Wieneke and Koch, 2010). A comparison of random matrices and the random hypersurface model under single target assumption is given in (Baum et al., 2010a).

The random matrix approach (Koch, 2008), to the best of our knowledge, has previously not been used in a framework for tracking an unknown number of multiple extended targets, in the presence of missed detections and clutter. The extended target PHD filter presented in this paper is capable of estimating both the kinematic states and the extents of multiple targets, in scenarios where both missed detections and clutter are allowed. At each time step, we first assume that the last estimated PHD is approximated with an unnormalized mixture of Gaussian inverse Wishart (GIW) distributions (i.e. the weights do not have to sum up to unity). We then show how the prediction and the measurement updates can be performed as was done in the single target case in (Koch, 2008), and also give a likelihood function suitable to handle multiple extended targets. The extended target PHD filter (Mahler, 2009) requires all the partitions of the measurement set. As a feasible approximation, as in (Granström et al., 2012) we use only a subset of all partitions. In order to better handle spatially close targets, two additional approaches to measurement set partitioning are suggested. The resulting filter, called the Gaussian inverse Wishart PHD filter (GIWPHD filter), is tested in simulations and in experiments based on laser scans.

The paper is organized as follows. Section 2 clearly specifies the extended targets of interest considered in this work and the selected extent modeling methodology. We mathematically describe the addressed target tracking problem in Section 3. Section 4 first lists the assumptions made, then gives the extended target PHD filter prediction and correction equations for the GIWPHD filter, and finally presents a merging and pruning scheme for the GIW components. In this work, due to space considerations, we are not able to give all the details about the main partitioning algorithm described originally in (Granström et al., 2012). For this reason, Section 5 presents only the required modifications and additions to the measurement partitioning method of (Granström et al., 2012). Results from simulations and experiments are presented in Section 6 and Section 7, and the paper is finalized with conclusions and future work in Section 8.

2 Modeling the target extension

The extended targets considered in this work are characterized by a number of reflection points spread over their extents. Early examples of extended target tracking assume fixed measurement sources on the target, which can be tracked individually to estimate the overall lumped behavior of the extended target (Salmond and Gordon, 1999). In many practical cases such an approach might fail, because the location of the measurement sources usually change fast according to the target sensor geometry. Having few measurements from a single source might not be sufficient to generate good quality individual tracks. For these reasons, we avoid such an explicit estimation of the measurement sources, and instead model the global behavior of the measurements over the target extent.

As a general and simple model for the target extensions, we use ellipsoids represented by positive definite matrices, proposed by in the pioneering work by Koch (2008). As admitted by Koch (2008), *“ellipsoidal object shapes are certainly a major simplification in view of large target groups which can be irregular in shape and in target density”*. This remark might be considered to be true for extended targets when the targets are very close to the sensor. In this case the target features form clusters of sensor reports that are too structured to be represented accurately by ellipsoids. Nevertheless, in many real-life target tracking scenarios, the targets are neither sufficiently far from the sensors to generate only a single measurement, nor are they sufficiently close to the sensors such that their features are clearly articulated.

In this work, the targets of interest are those sufficiently far away from the sensor so that their measurements resemble a cluster of points. In Figure 1 we give an example of the ellipsoidal model applied to real laser range data. The figure shows two plots with measurements of a bicyclist and a pedestrian. While neither bikes nor humans are shaped as ellipses, we see that, given the measurements, the random matrix model is a reasonable approximation of the extensions of the bicyclist and pedestrian. In the results section of this work we also present results from experiments where multiple humans are tracked in laser range data using the ellipsoidal models.

When the target extents are modeled as ellipsoids, clearly there are many different ways to estimate the parameters of the ellipses. Classically, the target tracking problem is considered in a Bayesian framework utilizing state estimators such as Kalman filters, its extensions, and particle filters. We follow this tradition and use Koch's Bayesian random matrix methodology, where the random matrices are inverse-Wishart distributed. The inverse Wishart probability density function is a convenient prior for the considered types of measurements, and iterative update formulae for the inverse-Wishart parameters are obtained. The Bayesian framework used conveniently supplies probabilistic uncertainty measures to describe the extension estimates.

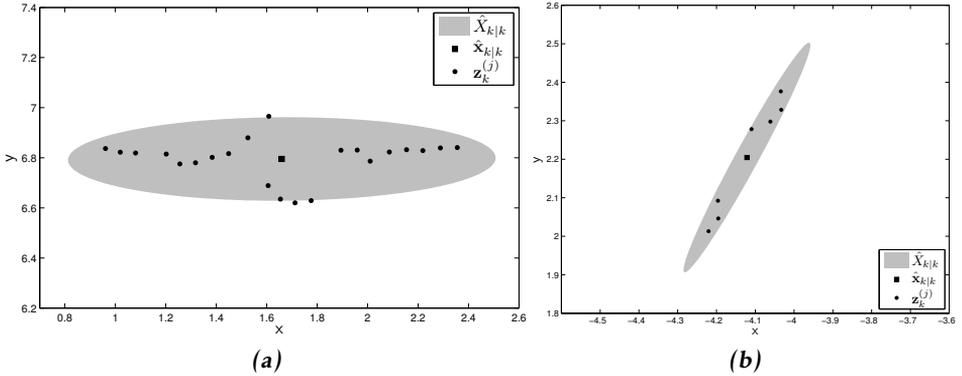


Figure 1: The ellipsoidal extension model applied to laser range data. Measurements of a bicyclist (a) and a pedestrian (b). Both legs of the pedestrian are measured, explaining the two distinct clusters of three and four measurements, respectively. The measurements $z_k^{(j)}$ are shown as black dots, the kinematical state estimates $\hat{\mathbf{x}}_{k|k}$ are shown as a black squares, and the representative extension state estimates $\hat{X}_{k|k}$ are shown as gray ellipses.

3 Target Tracking Problem Formulation

The set of extended targets at time k is denoted

$$\mathbf{X}_k = \left\{ \xi_k^{(i)} \right\}_{i=1}^{N_{x,k}}, \quad \xi_k^{(i)} \triangleq \left(\mathbf{x}_k^{(i)}, X_k^{(i)} \right), \quad (1)$$

where $N_{x,k}$ is the unknown number of targets, and, in accordance with (Koch, 2008), $\mathbf{x}_k^{(i)}$ is referred to as the kinematical state of the i :th target, and $X_k^{(i)}$ is referred to as the extension state. We denote the augmented state composed of the kinematic and extension states by $\xi_k^{(i)}$. Let the operation $|\cdot|$ denote set cardinality, i.e. $|\mathbf{X}_k| = N_{x,k}$. The target dynamic motion model is defined as (Koch, 2008)

$$\mathbf{x}_{k+1}^{(i)} = \left(F_{k+1|k} \otimes \mathbf{I}_d \right) \mathbf{x}_k^{(i)} + \mathbf{w}_{k+1}^{(i)} \quad (2)$$

where $\mathbf{w}_{k+1}^{(i)}$ is zero mean Gaussian process noise with covariance $\Delta_{k+1|k}^{(i)} = \mathbf{Q}_{k+1|k} \otimes X_k^{(i)}$ and d is the dimension of the target extent, i.e. $X_k^{(i)}$ is a $d \times d$ symmetric positive definite matrix and \mathbf{I}_d is an identity matrix of dimension d . The notation $A \otimes B$ denotes the Kronecker product of matrices A and B . The object kinematics are modeled up to the $(s-1)$:th derivative, i.e. the length of the kinematic state vector is $n_x = s \times d$. Here $s = 3$, and $F_{k+1|k}$ and $\mathbf{Q}_{k+1|k}$ are given by (Koch, 2008)

$$F_{k+1|k} = \begin{bmatrix} 1 & T_s & \frac{1}{2} T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & e^{-T_s/\theta} \end{bmatrix}, \quad (3a)$$

$$\mathbf{Q}_{k+1|k} = \Sigma^2 \left(1 - e^{-2T_s/\theta}\right) \text{diag}([0 \ 0 \ 1]), \quad (3b)$$

where T_s is the sampling time, Σ is the scalar acceleration standard deviation and θ is the maneuver correlation time.

The set of measurements obtained at time k is denoted

$$\mathbf{Z}_k = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}} \quad (4)$$

where $N_{z,k} = |\mathbf{Z}_k|$ is the number of measurements. The measurement model is defined as (Koch, 2008)

$$\mathbf{z}_k^{(j)} = (H_k \otimes \mathbf{I}_d) \mathbf{x}_k^{(i)} + \mathbf{e}_k^{(j)}, \quad (5)$$

where $\mathbf{e}_k^{(j)}$ is white Gaussian noise with covariance given by the target extension matrix $X_k^{(i)}$, and $H_k = [1 \ 0 \ 0]$ as in (Koch, 2008). Each target generates a Poisson distributed number of measurements, where the Poisson rate $\gamma(\xi_k)$ is a function of the augmented state.

Clutter measurements are modeled as being Poisson distributed in number, with rate parameter $\beta_{FA,k}$ clutter measurements per surveillance volume per scan. With surveillance volume \mathcal{S} , the mean number of clutter measurements is $\beta_{FA,k} \mathcal{S}$ clutter measurements per scan. The clutter measurements are modeled as being uniformly distributed over the surveillance area.

The goal at each time step is to estimate the set of targets \mathbf{X}_K given the sets of measurements $\mathbf{Z}^K = \{\mathbf{Z}_k\}_{k=1}^K$. This is achieved by propagating the predicted and updated PHDSs of the set of targets \mathbf{X}_k , denoted $D_{k|k-1}(\cdot)$ and $D_{k|k}(\cdot)$, respectively, using the extended target PHD filter presented in (Mahler, 2009).

4 The Gaussian inverse Wishart PHD filter

For the multi-target tracking problem described in Section 3, the extended target PHD filter prediction equations are given as follows (Mahler, 2003).

$$D_{k+1|k}(\xi_{k+1}) = \int p_S(\xi_k) p_{k+1|k}(\xi_{k+1}|\xi_k) D_{k|k}(\xi_k) d\xi_k + D_{k+1}^b(\xi_{k+1}), \quad (6)$$

where we omitted new target spawning, and

- $p_S(\cdot)$ is the probability of survival as a function of the augmented target state;
- $p_{k+1|k}(\cdot|\cdot)$ is the state transition density, describing the transition from state ξ_k to state ξ_{k+1} ;
- $D_k^b(\cdot)$ is the birth PHD, representing new targets.

More details on target spawning are given in Section 6.4. The correction equations for the extended target PHD filter has the following form (Mahler, 2009),

$$D_{k|k}(\xi_k | \mathbf{Z}^k) = L_{\mathbf{Z}_k}(\xi_k) D_{k|k-1}(\xi_k | \mathbf{Z}^{k-1}). \quad (7)$$

The measurement pseudo-likelihood function $L_{\mathbf{Z}_k}(\cdot)$ in (7) is defined as

$$L_{\mathbf{Z}_k}(\xi_k) \triangleq (1 - e^{-\gamma(\xi_k)}) p_D(\xi_k) + e^{-\gamma(\xi_k)} p_D(\xi_k) \sum_{\mathbf{p} \in \mathcal{Z}_{\mathbf{Z}_k}} \omega_{\mathbf{p}} \sum_{W \in \mathbf{p}} \frac{\gamma(\xi_k)^{|W|}}{d_W} \prod_{\mathbf{z}_k \in W} \frac{\phi_{\mathbf{z}_k}(\xi_k)}{\lambda_k c_k(\mathbf{z}_k)}, \quad (8)$$

where

- $\lambda_k \triangleq \beta_{FA,k} \mathcal{S}$ is the mean number of clutter measurements;
- $c_k(\mathbf{z}_k) = 1/\mathcal{S}$ is the spatial distribution of the clutter over the surveillance volume;
- the notation $\mathbf{p} \in \mathcal{Z}_{\mathbf{Z}_k}$ denotes that \mathbf{p} partitions the measurement set \mathbf{Z}_k into non-empty cells W . When used under a summation sign, the summation is over all possible partitions;
- the notation $W \in \mathbf{p}$ denotes that the set W is a cell in the partition \mathbf{p} . When used under a summation sign, the summation is over all sets in the partition;
- the quantities $\omega_{\mathbf{p}}$ and d_W are non-negative coefficients defined, for each partition \mathbf{p} and cell W respectively, as

$$\omega_{\mathbf{p}} = \frac{\prod_{W \in \mathbf{p}} d_W}{\sum_{\mathbf{p}' \in \mathcal{Z}_{\mathbf{Z}_k}} \prod_{W' \in \mathbf{p}'} d_{W'}}, \quad (9)$$

$$d_W = \delta_{|W|,1} + D_{k|k-1} \left[p_D \gamma^{|W|} e^{-\gamma} \prod_{\mathbf{z}_k \in W} \frac{\phi_{\mathbf{z}_k}(\cdot)}{\lambda_k c_k(\mathbf{z}_k)} \right], \quad (10)$$

where $\delta_{i,j}$ is the Kronecker delta and the notation $f[g]$ denotes the integral $\int f(x)g(x)dx$.

- $\phi_{\mathbf{z}_k}(\xi_k) \triangleq p(\mathbf{z}_k | \xi_k)$ is the likelihood function for a single target generated measurement. Under the measurement model (5) it is given as

$$\phi_{\mathbf{z}_k}(\xi_k) = \mathcal{N}(\mathbf{z}_k; (H_k \otimes \mathbf{I}_d) \mathbf{x}_k, X_k). \quad (11)$$

In the following subsections, we are going to assume that we are at an intermediate stage of estimation at time t_k and the current estimated PHD $D_{k|k}(\cdot)$ can be approximated as an unnormalized mixture of Gaussian inverse Wishart (GIW) distributions as follows.

$$D_{k|k}(\xi_k) \approx \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k) \mathcal{IW}(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}), \quad (12)$$

where

- $J_{k|k}$ is the number of components;
- $w_{k|k}^{(j)}$ is the weight of the j :th component;
- the notation $\mathcal{N}(x; m, P)$ denotes a Gaussian distribution defined over the variable x with mean m and covariance P ;
- the notation $\mathcal{IW}(X; \nu, V)$ denotes an inverse Wishart distribution defined over the variable X with degrees of freedom ν and inverse scale matrix V ;
- $m_{k|k}^{(j)}$ and $P_{k|k}^{(j)} \otimes X_k$ are the mean and covariance of the j :th component;
- $\nu_{k|k}^{(j)}$ and $V_{k|k}^{(j)}$ are the degrees of freedom and inverse scale matrix of the j :th component.

Further, let $\xi_{k|k}^{(j)}$ be an abbreviation of the sufficient statistics of the j :th GIW component, i.e.

$$\xi_{k|k}^{(j)} \triangleq \left(m_{k|k}^{(j)}, P_{k|k}^{(j)}, \nu_{k|k}^{(j)}, V_{k|k}^{(j)} \right). \quad (13)$$

Note that the distribution for the kinematical state x_k depends on the extension state X_k . Estimates of the kinematic state uncertainty and of the target extent are obtained as in (Koch, 2008),

$$\hat{P}_{k|k}^{(j)} = \frac{P_{k|k}^{(j)} \otimes V_{k|k}^{(j)}}{\nu_{k|k}^{(j)} + s - sd - 2}, \quad (14a)$$

$$\hat{X}_{k|k}^{(j)} = \frac{V_{k|k}^{(j)}}{\nu_{k|k}^{(j)} - 2d - 2}, \quad (14b)$$

for $\nu_{k|k}^{(j)}$ such that the denominators are positive. In the following, we give the assumptions made in the derivation of the GIWPHD filter in Section 4.1. The prediction and update formulas for the PHD representation in (12) are then presented in Section 4.2 and Section 4.3. Finally, GIW mixture reduction using a pruning and merging scheme is addressed in Section 4.4.

4.1 Assumptions

In order to derive prediction and correction equations for the GIWPHD filter, a number of assumptions are made. The first four assumptions are standard in most target tracking applications, see e.g. (Bar-Shalom and Fortmann, 1987).

Assumption D.1. Each target evolves and generates observations independently of all other targets. _____

Assumption D.2. Each target's kinematical part follows a linear Gaussian dynamical model, and the sensor has a linear Gaussian measurement model. _____

Assumption D.3. Clutter is Poisson distributed in number, and independent of target-originated measurements. _____

Assumption D.4. The survival probability is state independent, i.e. $p_S(\xi_k) = p_{S\cdot}$.

The next assumption is reasonable in scenarios where target interactions are negligible (Vo and Ma, 2006).

Assumption D.5. The predicted multi-target RFS is Poisson. _____

In (Vo and Ma, 2006; Granström et al., 2012) the PHD is represented as a mixture of Gaussian distributions, here a different assumption is made to accommodate the random matrix model.

Assumption D.6. The intensity of the birth RFS is a mixture of GIW distributions. _____

The following assumption is inherited from (Koch, 2008), where it is noted that it implies restrictions that can be justified in many practical cases.

Assumption D.7. The target augmented state transition density satisfies

$$p_{k+1|k}(\xi_{k+1}|\xi_k) \approx p_{k+1|k}^1(\mathbf{x}_{k+1}|X_{k+1}, \mathbf{x}_k) p_{k+1|k}^2(X_{k+1}|X_k) \quad (15)$$

for all ξ_k and ξ_{k+1} . _____

In addition to these, two more assumptions are made concerning the probability of detection $p_D(\cdot)$ and the rate $\gamma(\cdot)$ that governs each target's measurement generation. These assumptions require a bit more elaboration.

Assumption D.8. The following approximation about $p_D(\cdot)$ holds for all ξ_k

$$\begin{aligned} p_D(\xi_k) &\mathcal{N}\left(\mathbf{x}_k; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right) \\ &\approx p_D\left(\xi_{k|k-1}^{(j)}\right) \mathcal{N}\left(\mathbf{x}_k; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right). \end{aligned} \quad (16)$$

Let $p_D^{(j)} \triangleq p_D\left(\xi_{k|k-1}^{(j)}\right)$ abbreviate the probability of detection for the j :th GIW component. _____

In Assumption D.8 the approximation (16) is trivially satisfied when $p_D(\cdot) = p_D$, i.e. when $p_D(\cdot)$ is constant. In general, Assumption D.8 holds approximately when the function $p_D(\cdot)$ does not vary much in the uncertainty zone of a target in the augmented state space ξ_k . This is true either when $p_D(\cdot)$ is a sufficiently smooth function, or when the signal to noise ratio (SNR) is high enough such that the uncertainty zone is sufficiently small. A similar approach to variable probability of detection has been taken in order to model the *clutter notch* in ground moving target indicator target tracking (Ulmke et al., 2007).

For the expected number of measurements from the targets, represented by $\gamma(\cdot)$, similar remarks apply and the following assumption is made.

Assumption D.9. The following approximation about $\gamma(\cdot)$ holds for all ξ_k , $j = 1, \dots, J_{k|k-1}$ and all integers $n \geq 1$,

$$\begin{aligned} & e^{-\gamma(\xi_k)} \gamma^n(\xi_k) \mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; v_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right) \\ & \approx e^{-\gamma(\xi_{k|k-1}^{(j)})} \gamma^n(\xi_{k|k-1}^{(j)}) \mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; v_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right). \end{aligned} \quad (17)$$

Let $\gamma^{(j)} \triangleq \gamma(\xi_{k|k-1}^{(j)})$ abbreviate the expected number of measurements for the j :th GIW component. _____

The trivial situation $\gamma(\cdot) = \gamma$, i.e. when $\gamma(\cdot)$ is constant, is again a special case where Assumption D.9 is satisfied. In general, satisfying Assumption D.9 is more difficult than Assumption D.8. Nevertheless Assumption D.9 is expected to hold approximately either when $\gamma(\cdot)$ is a sufficiently smooth function or when the signal to noise ratio (SNR) is high enough such that the uncertainty zone of a target in the augmented state space ξ_k is sufficiently small.

4.2 Prediction

Utilizing Assumptions D.4 and D.7, the prediction of existing targets can be written as

$$\begin{aligned} & p_S \int p_{k+1|k}^1(\mathbf{x}_{k+1}|X_{k+1}, \mathbf{x}_k) p_{k+1|k}^2(X_{k+1}|X_k) D_{k|k}(\mathbf{x}_k, X_k) d\mathbf{x}_k dX_k \\ & = p_S \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \underbrace{\int \mathcal{N}\left(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_{k+1}\right) d\mathbf{x}_k}_{\text{Kinematical part}} \\ & \quad \times \underbrace{\int \mathcal{IW}\left(X_k; v_{k|k}^{(j)}, V_{k|k}^{(j)}\right) p_{k+1|k}^2(X_{k+1}|X_k) dX_k}_{\text{Extension part}}. \end{aligned} \quad (18)$$

Using the linear Gaussian model given in (2), the prediction for the kinematical part becomes (Koch, 2008)

$$\begin{aligned} & \int \mathcal{N}\left(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_{k+1}\right) p_{k+1|k}^1(\mathbf{x}_{k+1}|X_{k+1}, \mathbf{x}_k) d\mathbf{x}_k \\ & = \mathcal{N}\left(\mathbf{x}_{k+1}; m_{k+1|k}^{(j)}, P_{k+1|k}^{(j)} \otimes X_{k+1}\right), \end{aligned} \quad (19)$$

where

$$m_{k+1|k}^{(j)} = (F_{k+1|k} \otimes \mathbf{I}_d) m_{k|k}^{(j)}, \quad (20a)$$

$$P_{k+1|k}^{(j)} = F_{k+1|k} P_{k|k}^{(j)} F_{k+1|k}^T + \mathbf{Q}_{k+1|k}. \quad (20b)$$

The extension part is less straightforward. Here, we apply the same heuristic approach as in (Koch, 2008), i.e. we make the approximation

$$\int \mathcal{IW}\left(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}\right) p_{k+1|k}^2(X_{k+1}|X_k) dX_k \approx \mathcal{IW}\left(X_{k+1}; \nu_{k+1|k}^{(j)}, V_{k+1|k}^{(j)}\right) \quad (21)$$

where the predicted degrees of freedom and inverse scale matrix are approximated by

$$\nu_{k+1|k}^{(j)} = e^{-T_s/\tau} \nu_{k|k}^{(j)}, \quad (22a)$$

$$V_{k+1|k}^{(j)} = \frac{\nu_{k+1|k}^{(j)} - d - 1}{\nu_{k|k}^{(j)} - d - 1} V_{k|k}^{(j)}, \quad (22b)$$

where τ is a temporal decay constant. Thus, the PHD corresponding to predicted existing targets is

$$\sum_{j=1}^{J_{k|k}} w_{k+1|k}^{(j)} \mathcal{N}\left(\mathbf{x}_{k+1}; m_{k+1|k}^{(j)}, P_{k+1|k}^{(j)} \otimes X_{k+1}\right) \mathcal{IW}\left(X_{k+1}; \nu_{k+1|k}^{(j)}, V_{k+1|k}^{(j)}\right), \quad (23)$$

where $w_{k+1|k}^{(j)} = p_S w_{k|k}^{(j)}$, the Gaussian mean $m_{k+1|k}^{(j)}$ and covariance $P_{k+1|k}^{(j)}$ are given in (20), and the inverse Wishart degrees of freedom $\nu_{k+1|k}^{(j)}$ and inverse scale matrix $V_{k+1|k}^{(j)}$ are given in (22).

The birth PHD

$$D_k^b(\xi_k) = \sum_{j=1}^{J_{b,k}} w_{b,k}^{(j)} \mathcal{N}\left(\mathbf{x}_k; m_{b,k}^{(j)}, P_{b,k}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{b,k}^{(j)}, V_{b,k}^{(j)}\right), \quad (24)$$

represents new targets that appear at time step k . The full predicted PHD $D_{k+1|k}(\xi_{k+1})$ is the sum of the PHD of predicted existing targets (23) and the birth PHD (24), and contains a total of $J_{k+1|k} = J_{k|k} + J_{b,k+1}$ GIW components.

4.3 Correction

The corrected PHD is a GIW mixture given by

$$D_{k|k}(\xi_k) = D_{k|k}^{\text{ND}}(\xi_k) + \sum_{\mathbf{p} \in \mathcal{Z}_k} \sum_{W \in \mathbb{P}} D_{k|k}^{\text{D}}(\xi_k, W), \quad (25)$$

where $D_{k|k}^{\text{ND}}(\cdot)$, handling the no detection cases, is given by

$$D_{k|k}^{\text{ND}}(\xi_k) = \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}\left(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)}\right) \mathcal{IW}\left(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}\right), \quad (26a)$$

$$w_{k|k}^{(j)} = \left(1 - \left(1 - e^{-\gamma^{(j)}}\right) p_{\text{D}}\right) w_{k|k-1}^{(j)}, \quad (26b)$$

$$\xi_{k|k}^{(j)} = \xi_{k|k-1}^{(j)}. \quad (26c)$$

The GIW mixture $D_{k|k}^D(\xi_k, W)$, handling the detected target cases, requires the likelihood of the measurements in each cell W ,

$$\prod_{\mathbf{z}_k \in W} \frac{\phi_{\mathbf{z}_k}(\xi_k)}{\lambda_k c_k(\mathbf{z}_k)} = \beta_{FA,k}^{-|W|} \prod_{\mathbf{z}_k \in W} \mathcal{N}\left(\mathbf{z}_k^{(i)}; (H_k \otimes \mathbf{I}_d) \mathbf{x}_k, X_k\right), \quad (27)$$

multiplied with the predicted GIW components,

$$\mathcal{N}\left(\mathbf{x}_k; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right). \quad (28)$$

The product of (27) and (28) can be rewritten as

$$\beta_{FA,k}^{-|W|} \mathcal{L}_k^{(j,W)} \mathcal{N}\left(\mathbf{x}_k; m_{k|k}^{(j,W)}, P_{k|k}^{(j,W)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{k|k}^{(j,W)}, V_{k|k}^{(j,W)}\right). \quad (29)$$

The details behind the derivation are given in Appendix A.1. The corrected Gaussian mean and covariance and inverse Wishart degrees of freedom and inverse scale matrix in (29) are given by

$$m_{k|k}^{(j,W)} = m_{k|k-1}^{(j)} + \left(K_{k|k-1}^{(j,W)} \otimes \mathbf{I}_d\right) \varepsilon_{k|k-1}^{(j,W)}, \quad (30a)$$

$$P_{k|k}^{(j,W)} = P_{k|k-1}^{(j)} - K_{k|k-1}^{(j,W)} S_{k|k-1}^{(j,W)} \left(K_{k|k-1}^{(j,W)}\right)^T, \quad (30b)$$

$$\nu_{k|k}^{(j,W)} = \nu_{k|k-1}^{(j)} + |W|, \quad (30c)$$

$$V_{k|k}^{(j,W)} = V_{k|k-1}^{(j)} + N_{k|k-1}^{(j,W)} + Z_k^W, \quad (30d)$$

where the centroid measurement, scatter matrix, innovation factor, gain matrix, innovation vector and innovation matrix are defined as

$$\bar{\mathbf{z}}_k^W = \frac{1}{|W|} \sum_{\mathbf{z}_k^{(i)} \in W} \mathbf{z}_k^{(i)}, \quad (31a)$$

$$Z_k^W = \sum_{\mathbf{z}_k^{(i)} \in W} \left(\mathbf{z}_k^{(i)} - \bar{\mathbf{z}}_k^W\right) \left(\mathbf{z}_k^{(i)} - \bar{\mathbf{z}}_k^W\right)^T, \quad (31b)$$

$$S_{k|k-1}^{(j,W)} = H_k P_{k|k-1}^{(j)} H_k^T + \frac{1}{|W|}, \quad (31c)$$

$$K_{k|k-1}^{(j,W)} = P_{k|k-1}^{(j)} H_k^T \left(S_{k|k-1}^{(j,W)}\right)^{-1}, \quad (31d)$$

$$\varepsilon_{k|k-1}^{(j,W)} = \bar{\mathbf{z}}_k^W - (H_k \otimes \mathbf{I}_d) m_{k|k-1}^{(j)}, \quad (31e)$$

$$N_{k|k-1}^{(j,W)} = \left(S_{k|k-1}^{(j,W)}\right)^{-1} \varepsilon_{k|k-1}^{(j,W)} \left(\varepsilon_{k|k-1}^{(j,W)}\right)^T. \quad (31f)$$

The likelihood in (29) is given by

$$\mathcal{L}_k^{(j,W)} = \frac{1}{\left(\pi^{|W|} |W| S_{k|k-1}^{(j,W)}\right)^{\frac{d}{2}}} \frac{|V_{k|k-1}^{(j)}|^{\frac{\nu_{k|k-1}^{(j)}}{2}} \Gamma_d\left(\frac{\nu_{k|k-1}^{(j)}}{2}\right)}{|V_{k|k}^{(j,W)}|^{\frac{\nu_{k|k}^{(j,W)}}{2}} \Gamma_d\left(\frac{\nu_{k|k-1}^{(j)}}{2}\right)}. \quad (32)$$

where $|V|$ denotes the determinant of the matrix V , and $|W|$ is the number of measurements in the cell W . The updated GIW component weight is given by

$$w_{k|k}^{(j,W)} = \frac{\omega_p}{d_W} e^{-\gamma^{(j)}} \left(\frac{\gamma^{(j)}}{\beta_{FA,k}}\right)^{|W|} p_D^{(j)} \mathcal{L}_k^{(j,W)} w_{k|k-1}^{(j)}, \quad (33)$$

where

$$d_W = \delta_{|W|,1} + \sum_{\ell=1}^{J_{k|k-1}} e^{-\gamma^{(\ell)}} \left(\frac{\gamma^{(\ell)}}{\beta_{FA,k}}\right)^{|W|} p_D^{(\ell)} \mathcal{L}_k^{(\ell,W)} w_{k|k-1}^{(\ell)}. \quad (34)$$

Finally, the coefficients ω_p can be calculated by (9). The corrected PHD is of the form given in (12) with weights given by (33), and Gaussian and inverse Wishart parameters given in (30). Let $|p_p|$ denote the number of cells W in the p :th partition, and let the set of partitions contain P unique partitions. The corrected PHD then has $J_{k|k} = J_{k|k-1} + J_{k|k-1} \sum_{p=1}^P |p_p|$ GIW components.

4.4 Pruning and merging

From the prediction and correction, one quickly realizes that as time progresses, the number of GIW components increases rapidly. To keep the number of components at a tractable level, pruning and merging of GIW components is performed similarly to (Vo and Ma, 2006). Empirically we have found that the merging threshold U must be chosen conservatively to avoid merging GIW components which correspond to multiple spatially close targets, because merging such components may cause cardinality error. The details of the implemented pruning and merging scheme are given below in Table 1. Note that calculation of the merged covariance $\tilde{P}_{k|k}^{(\ell)}$ does not include the spread of means, the reason is that the means $m_{k|k}^{(i)}$ and covariances $P_{k|k}^{(i)}$ are of different dimensions ($s \times d$ and s , respectively). However, with a conservative merging threshold U , the spread of means is typically quite small and is thus negligible.

We also alert the reader about the very simple approach to merging of inverse-Wishart parameters in Table 1. This procedure is sufficient when a conservative threshold is used, and the GIWPHD filter is not very sensitive to changes in the merging algorithm. Nevertheless, finding a better method for GIW component merging is a potential subject for future research.

Table 1: Pseudo-code for GIWPHD filter pruning and merging

1: input: GIW components $\left\{w_{k k}^{(j)}, \xi_{k k}^{(j)}\right\}_{j=1}^{J_{k k}}$, a truncation threshold T , a merging threshold U and a maximum allowable number of GIW components J_{\max} .
2: initialize: Set $\ell \leftarrow 0$ and $I \leftarrow \left\{i = 1, \dots, J_{k k} \mid w_{k k}^{(i)} > T\right\}$.
3: repeat
4: $\ell \leftarrow \ell + 1$
5: $j \leftarrow \arg \max_{i \in I} w_{k k}^{(i)}$
6: Compute $\hat{P}_{k k}^{(j)}$ using (14a).
7: $L \leftarrow \left\{i \in I \mid \left(m_{k k}^{(i)} - m_{k k}^{(j)}\right)^{\top} \left(\hat{P}_{k k}^{(j)}\right)^{-1} \left(m_{k k}^{(i)} - m_{k k}^{(j)}\right) \leq U\right\}$
8: $\tilde{w}_{k k}^{(\ell)} \leftarrow \sum_{i \in L} w_{k k}^{(i)}$
9: $\tilde{m}_{k k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k k}^{(\ell)}} \sum_{i \in L} w_{k k}^{(i)} m_{k k}^{(i)}$,
10: $\tilde{P}_{k k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k k}^{(\ell)}} \sum_{i \in L} w_{k k}^{(i)} P_{k k}^{(i)}$
11: $\tilde{v}_{k k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k k}^{(\ell)}} \sum_{i \in L} w_{k k}^{(i)} v_{k k}^{(i)}$,
12: $\tilde{V}_{k k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k k}^{(\ell)}} \sum_{i \in L} w_{k k}^{(i)} V_{k k}^{(i)}$
13: $I \leftarrow I \setminus L$
14: until $I = \emptyset$
15: If $\ell > J_{\max}$ then replace $\left\{\tilde{w}_{k k}^{(j)}, \tilde{m}_{k k}^{(j)}, \tilde{P}_{k k}^{(j)}, \tilde{v}_{k k}^{(j)}, \tilde{V}_{k k}^{(j)}\right\}_{j=1}^{\ell}$ by those of the J_{\max} GIW components with largest weights.
16: output: $\left\{\tilde{w}_{k k}^{(j)}, \tilde{\xi}_{k k}^{(j)}\right\}_{j=1}^{\ell}$, $\tilde{\xi}_{k k}^{(j)} = \left(\tilde{m}_{k k}^{(j)}, \tilde{P}_{k k}^{(j)}, \tilde{v}_{k k}^{(j)}, \tilde{V}_{k k}^{(j)}\right)$

4.5 Implementation of the GIWPHD filter

To facilitate implementation, we give pseudo code for the GIWPHD filter, and address implementation issues and computational complexity, in Appendix A.

5 Partitioning the measurement set

A central part of the correction equation given in (7), (8), is the partitioning of the set of measurements \mathbf{Z}_k into partitions p containing non-empty cells W with measurements $\mathbf{z}_k^{(j)}$. For a given partition, the cells can be understood as containing measurements that are all from the same source, either a single target or a clutter source.

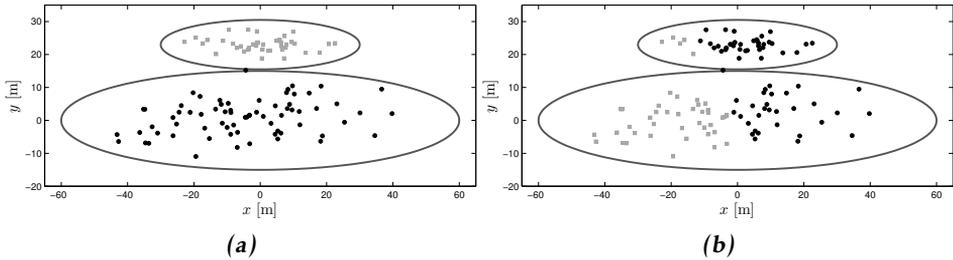


Figure 2: Illustration of Sub-Partition. (a) Two spatially close extended targets, with corresponding measurements in black and gray. (b) Sub-cells resulting from K -means++, shown in black and gray. Ideally, the measurements should be split into two sub-cells along the $y = 15$ line.

The measurement pseudo-likelihood (8) requires a summation over all possible partitions, which quickly becomes intractable because the number of possible partitions increases very rapidly as the size of \mathbf{Z}_k increases (Mahler, 2009; Granström et al., 2012). It has been noted that the full set of partitions can be approximated with a subset of partitions, so long as this subset contains the most likely ones among all of the possible partitions (Granström et al., 2010; Granström et al., 2012). A method called Distance Partition was suggested in (Granström et al., 2010), and it was augmented with the Sub-Partition algorithm in (Granström et al., 2012) to better handle the case of spatially close targets.

Distance Partition is based on the fundamental insight that measurements that are caused by the same extended target are spatially close to each other. Partitions are computed such that spatially close measurements are put into the same cell. However, a method based only on this places measurements from multiple targets in the same cell if two or more targets are spatially close, which may cause cardinality errors. In the Sub-Partition algorithm presented in (Granström et al., 2012), this problem was solved by generating additional partitions by considering the number of measurements in each cell $|W|$, and comparing it to the expected number of measurements from a single target. Given a maximum likelihood estimate K of the number of targets that caused the measurements in the cell, Sub-Partition uses K -means++ clustering to split the cell into K sub-cells. A new partition, that includes the sub-cells instead of the original cell, is added to the list of partitions. Though this method solves the cardinality issues in many practical cases, it is noted in (Granström et al., 2012) that it is only a first order solution to the problem.

Initial simulations with extended targets modeled using random matrices showed that Distance Partitioning with Sub-Partition was insufficient to handle some instances of multiple extended targets that are spatially close. The phenomenon is best explained with an example. Consider the two different sized and spatially close extended targets, with corresponding measurements, in Figure 2a. Distance

Partition would place all measurements in the same cell, due to the spatial proximity of the measurements. Compare to the division of the measurements using K -means++ in Figure 2b, which is the algorithm used in Sub-Partition. The result from K -means++ is typical, because for this type of scenario the K -means++ loss function profits much more by dividing the measurements by a vertical line, rather than a horizontal one. Because such a resulting additional partition will get a relatively lower likelihood, compared to the partition which assigns all the measurements to a single target (obtained initially by Distance Partitioning), the additional partition would not improve performance. Despite using Sub-Partition, the result would typically be a cardinality error in the filtering.

In order to be able to handle this type of true target scenario, in this paper two additional partitioning methods are suggested. The first is a method called Prediction Partition, which is based on the predicted GIWPHD components. The second method, called EM Partition, is based on the expectation maximization (EM) algorithm (Dempster et al., 1977). Both methods are based on the intuition that in order to solve the problem for situations as in Figure 2, one has to incorporate the predicted kinematic and extent states of the targets into the partitioning process.

5.1 Prediction Partition

This partitioning method uses the predicted GIW components. For components with weight $w_{k+1|k}^{(j)} > 0.5$, a d -dimensional extension estimate $\hat{X}_{k+1|k}^{(j)}$ is computed as in (14b). A corresponding position mean is obtained by taking the d first components of $m_{k+1|k}^{(j)}$, denoted $m_{k+1|k}^{(j),d}$. A partition is obtained by iterating over the components, in the order of decreasing weight, and putting all measurements $\mathbf{z}_k^{(i)}$ that fulfill

$$\left(\mathbf{z}_k^{(i)} - m_{k+1|k}^{(j),d}\right)^T \left(\hat{X}_{k+1|k}^{(j)}\right)^{-1} \left(\mathbf{z}_k^{(i)} - m_{k+1|k}^{(j),d}\right) < \Delta_d(p) \quad (35)$$

into the same cell. Here, $\Delta_d(p)$ is computed using the inverse cumulative χ^2 distribution with d degrees of freedom, for probability $p = 0.99$. If a measurement falls into two or more extension estimates, it is only put into the cell corresponding to the component with highest weight. The measurements that do not fulfill (35) for any GIW component are placed in individual cells containing only one measurement.

This method works well when the true target motion can be well modeled by the dynamic motion model (2). However, when the targets maneuver the method is expected to be insufficient because the target predictions will be significantly erroneous.

5.2 EM Partition

The reason that K -means++ were successful in some scenarios in (Granström et al., 2012) was that the targets were mainly of the same size and circular (i.e. as opposed to elliptical). A typical extended target scenario can have targets of

quite different sizes generating significantly different numbers of measurements, and the targets' measurement distributions can be significantly skewed, rather than circular. When there are targets of different sizes, the K -means++ algorithm, which does not use any measure of the clusters' physical sizes, often fails. The Expectation Maximization (EM) algorithm for Gaussian mixtures, which is a generalization of the K -means++ algorithm (see e.g. chapter 9 of (Bishop, 2006)), incorporates both cluster sizes and number of measurements in each cluster via the covariances and the mixing coefficients. The specifics of the EM algorithm for Gaussian mixtures can be found in e.g. (Bishop, 2006).

In the EM Partition algorithm, the Gaussian mixture parameters are initialized with means $\mu_\ell = m_{k+1|k}^{(j),d}$, covariances $\Sigma_\ell = \hat{X}_{k+1|k}^{(j)}$ and mixing coefficients $\pi_\ell \propto \gamma\left(\xi_{k+1|k}^{(j)}\right)$ for components j with weight $w_{k+1|k}^{(j)} > 0.5$. An additional mixture component is added with mean μ_ℓ at the center of the surveillance area, circular covariance Σ_ℓ scaled such that the corresponding 99% probability volume approximately covers the surveillance area, and mixing coefficient $\pi_\ell = 10^{-9}$. The mixing coefficients π_ℓ are normalized to satisfy $\sum_\ell \pi_\ell = 1$ before the first E-step.

The additional mixture component is added to capture the clutter measurements, such that the mixture components corresponding to the target estimates can converge approximately to the true partitioning. Note that for a given set of initial Gaussian mixture components, the EM algorithm will converge to the closest local maximum of the likelihood function, i.e. there is no guarantee that EM converges to the global maximum. Because EM Partition is initialized using the predicted GIW components, similarly to Prediction Partition it is sensitive to maneuvers that are modeled poorly by the motion model. However, because of the adaptation capability of the EM-iterations, EM Partition is slightly less sensitive than Prediction Partition.

5.3 Discussion

It is important to note that each of the three partitioning methods used in this work, i.e. Distance Partition with Sub-Partition (Granström et al., 2012), Prediction Partition and EM Partition, have its respective failure modes. A problem with Distance Partition with Sub-Partition was highlighted in Figure 2. Prediction Partition relies on the prediction of the GIW components, this method sometimes returns a non-informative partition when targets are maneuvering. EM Partition can converge to a local maximum of the likelihood function that yields a non-informative partition. For this reason, it is a better choice to use all three methods, rather than just one method on its own. The more partitions that are used, the better the full set of partitions is approximated. Indeed, it is possible that adding further partitioning methods would improve performance, however this should also be balanced against the fact that considering more partitions requires more computations.

6 Simulation Results

This section presents results from extended target tracking simulations. The target tracking setup is presented in the next section, followed by the results from four different extended target tracking scenarios.

6.1 Target tracking setup

Four different scenarios were simulated, each with two targets. The true tracks are shown in Figure 3. The true target extensions are given by

$$X_k^{(i)} = R_k^{(i)} \text{diag} \left([A_i^2 \ a_i^2] \right) \left(R_k^{(i)} \right)^T, \quad (36)$$

where $R_k^{(i)}$ is a rotation matrix applied such that the i :th extension's major axis is aligned with the i :th target's direction of motion at time step k , and A_i and a_i are the length of the major and minor axes, respectively. In all four scenarios, the major and minor axes are $(A_1, a_1) = (20, 5)$ and $(A_2, a_2) = (10, 2.5)$ for the two targets, respectively.

The expected number of measurements generated by the targets is assumed to be a function of the extended target volume

$$\mathcal{V}_k^{(i)} \triangleq \pi \sqrt{|X_k^{(i)}|} = \pi A_i a_i. \quad (37)$$

This assumption is reasonable in many real world scenarios, where a smaller target would occupy fewer of the sensor's resolution cells than a larger target, thus yielding fewer measurements. Here we adopt the following simple model for the expected number of measurements that the targets generate,

$$\gamma_k^{(i)} = \left\lfloor \sqrt{\frac{4}{\pi} \mathcal{V}_k^{(i)} + 0.5} \right\rfloor = \left\lfloor 2\sqrt{A_i a_i} + 0.5 \right\rfloor, \quad (38)$$

where $\lfloor \cdot \rfloor$ is the floor function and $\lfloor x + 0.5 \rfloor$ rounds x to the nearest integer. This model is equivalent to assuming a uniform expected number of measurements per square root of surveillance area. In a typical real world scenario, the number of target measurements may also depend on the distance between the target and the sensor, i.e. depend on the kinematical target state $\mathbf{x}_k^{(i)}$. This case can easily be handled with a modified expected number of measurements model. For the sake of simplicity, this case is not included in this paper, and the readers are referred to (Granström et al., 2012) for such an example.

The motion model parameters are set to $T_s = 1\text{s}$, $\theta = 1\text{s}$, $\Sigma = 0.1\text{m/s}^2$ and $\tau = 5\text{s}$. In three of four scenarios, the parameters of the $J_{b,k} = 2$ birth PHD components are set as follows, $w_{b,k}^{(j)} = 0.1$, and

$$m_{b,k}^{(j)} = \left[\left(\mathbf{x}_0^{(j)} \right)^T \mathbf{0}_4^T \right]^T, \quad P_{b,k}^{(j)} = \text{diag} \left([100^2 \ 25^2 \ 25^2] \right),$$

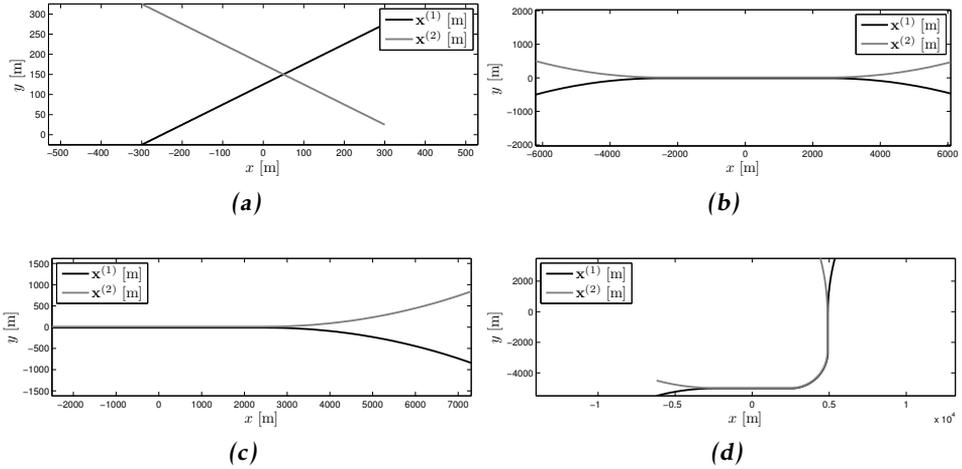


Figure 3: True target tracks used in simulations. (a) Crossing tracks. (b) Parallel tracks. (c) Separating tracks. (d) Turning tracks.

$$v_{b,k}^{(j)} = 7, \quad V_{b,k}^{(j)} = \text{diag}([1 \ 1]). \quad (39)$$

The mean vectors $m_{b,k}^{(j)}$ are set such that they correspond to the starting points of the true targets. In the fourth scenario, there is $J_{b,k} = 1$ birth component, with mean vector set to the mean of the two targets' starting points. Knowing the starting points of the targets *a priori* is naturally not possible in many real world scenarios. In the experiment section, we elaborate further on how the birth PHD can be constructed in a real scenario.

A total of 100 Monte Carlo simulations were performed for each scenario, with a clutter rate of 10 clutter measurements per time step. The results are presented in terms of the multi-target measure optimal subpattern assignment metric (OSPA) (Schuhmacher et al., 2008), cardinality and length of the estimated major and minor axes of the extension matrices.

6.2 Crossing tracks

In this scenario, the target tracks cross at close distance, see Figure 3a. The results are shown in Figure 4. The plots clearly show that straight line motion can be readily handled by the presented filter, even when the targets are spatially close. Noteworthy are the estimates of the major and minor axes, A_i and a_i , respectively. The results show that extensions which do not change over time (e.g. do not grow, shrink or rotate) can be estimated with low error.

6.3 Parallel tracks

In the second scenario, the two targets move closer and then move in parallel, before separating again, see Figure 3b. While moving in parallel, the true target

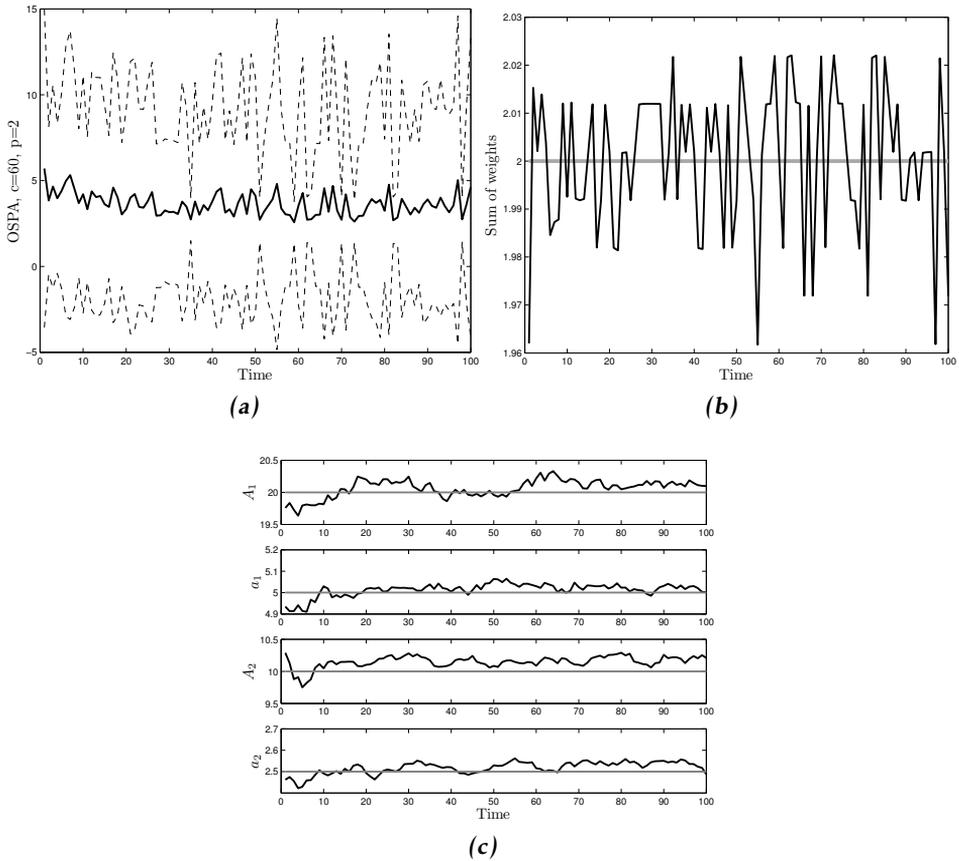


Figure 4: (a) Mean OSPA (solid line) \pm one standard deviation (dashed lines). (b) Cardinality estimate, taken as the sum of weights (black), compared to true cardinality (gray). (c) Estimates of the major and minor axes, A_i and a_i , respectively. Mean estimates (black) compared to true value (gray).

extensions' three standard deviation ellipses (corresponding to the 99% probability volume) are separated by 2.5m. The results are shown in Figure 5. The mean sum of weights is close to the true value, however there is a downward trend while the targets are moving in parallel. This is caused mainly by missed detections, which often causes the PHD filter to lose the target estimate corresponding to the target that was not detected. In the subsequent time steps, when the target is detected again, the measurements from both targets are typically treated as being caused by one target.

This can also be seen in the OSPA value, which increases during parallel motion, and also has larger standard deviation. The estimates of A_i and a_i are slightly worse than in the previous scenario. In the beginning and end of the simulation

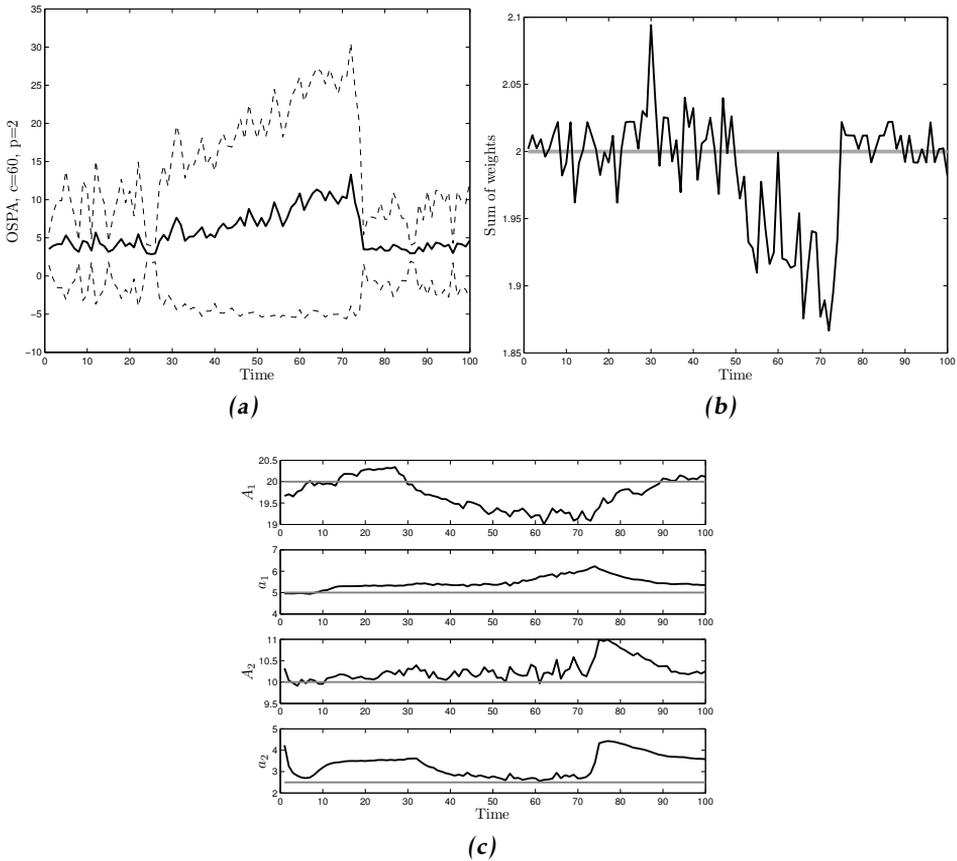


Figure 5: (a) Mean OSPA (solid line) \pm one standard deviation (dashed lines). (b) Cardinality estimate, taken as the sum of weights (black), compared to true cardinality (gray). (c) Estimates of the major and minor axes, A_i and a_i , respectively. Mean estimates (black) compared to true value (gray).

this is caused by the rotation of the extension matrices X_k . This result is intuitive – the turning makes the extension more difficult to track since the prediction (22) does not account for the rotation of the extension.

6.4 Separating tracks

In the third scenario, the two targets start such that their respective three standard deviation ellipsoids, computed from the true extensions X_k , are touching. First the targets move in parallel, after about half the scenario they separate, see Figure 3c. For this scenario, a birth PHD with $J_{b,k} = 1$ component was used. The results are shown in Figure 6. For the first half of this scenario, when the target extents are touching, the filter incorrectly estimates one large extended target, instead of two smaller ones, in about 60% of the Monte Carlo simulations. This is

what causes the mean sum of weights to be around 1.4. The targets start to separate at time 52, and from time 57 the cardinality is estimated correctly. Because the cardinality is underestimated in about 60% of the Monte Carlo simulations, the estimated major and minor axes, A_i and a_i , are difficult to interpret for the first half of the scenario. When the GIWPHD filter estimates only one target, the major axis of the extracted target is estimated to be slightly lower than $A_1 = 20\text{m}$. We had expected the major axis to be estimated as 20m or more when the targets are combined. However, this “underestimation” appears to be a property of the particular prediction and correction equations used for the inverse Wishart parameters. The major axes of the targets are, in a way, averaged to obtain a smaller estimate than 20m. For the second half, when the targets are separated, the results are better.

Furthermore, the results show that there is little need for a specific model for target spawning. As soon as the two targets are slightly separated, the partitioning algorithm (Distance Partition) automatically starts to generate a partition that suits the spawning event. This partition then dominates the other partitions, which can be seen e.g. in terms of the partition weights ω_p . This process is evident also from the cardinality estimates, which are corrected shortly after the targets separate.

6.5 Closely spaced targets

In the fourth scenario, the targets move closer and then move in parallel, both in straight lines and through a curve, before separating again, see Figure 3d. Estimating cardinality correctly becomes increasingly difficult as multiple targets move close to each other. Early tests showed that Distance Partition with Sub-Partition was insufficient to handle some cases of spatially close extended targets modeled as random matrices. To improve performance, Prediction Partition and EM Partition was implemented.

To test the PHD-filters capability of tracking multiple closely spaced targets, the scenarios in Figure 3b and Figure 3d were simulated when the targets’ extents were separated by a distance d . The tracks in Figure 3b were simulated for separating distances $d = 0, 0.5, 1, \dots, 5$ [m], and the mean sum of weights is shown in Figure 7a. When rounded to the nearest integer there is no cardinality error at any distance d , however estimating cardinality correctly becomes increasingly difficult at closer distances, which is shown by the lower mean value for $d < 2.5\text{m}$. Without Prediction Partition and EM Partition, simulations show that the cardinality is underestimated for distances $d < 7\text{m}$.

The tracks in Figure 3d contain a turn, making prediction of the extended target estimates harder, because the dynamic motion model is constant velocity and predicts target motion in a straight line. This scenario was simulated at two different speeds, 125m/s and 62.5m/s. The separating distances were $d = 0, 0.5, 1, \dots, 25$ [m] for the faster speed and $d = 0, 0.5, 1, \dots, 5$ [m] for the slower speed.

At the higher speed, target prediction is more difficult, especially during the turn,

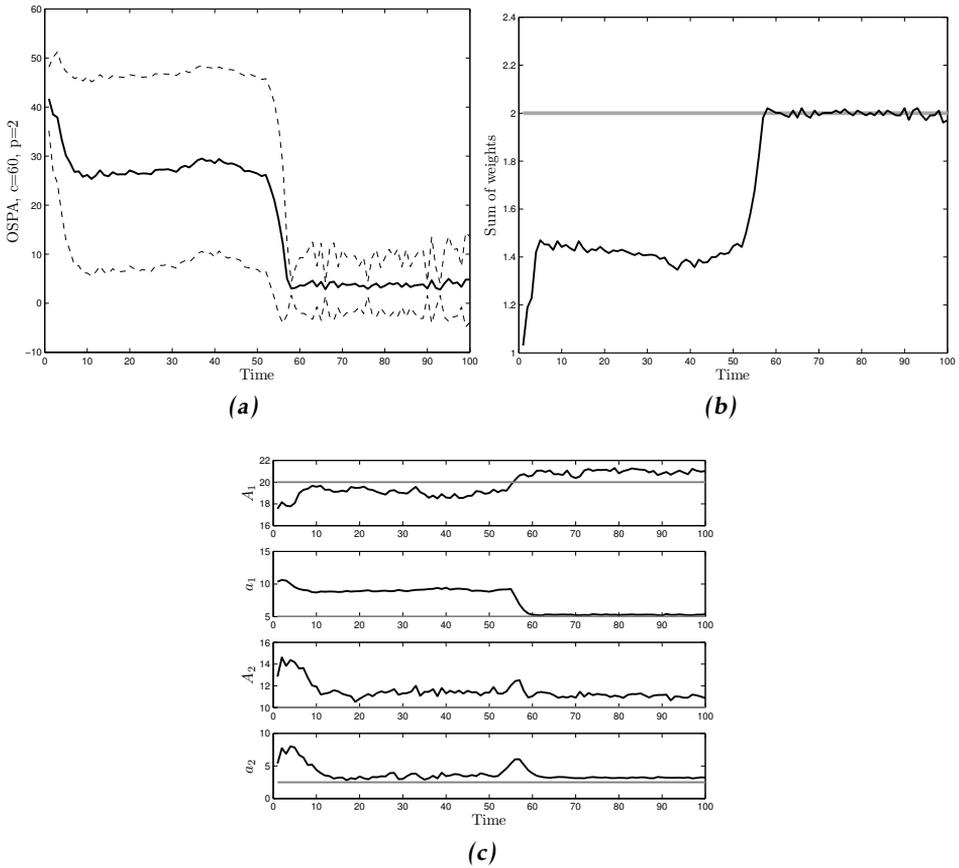


Figure 6: (a) Mean OSPA (solid line) \pm one standard deviation (dashed lines). (b) Cardinality estimate, taken as the sum of weights (black), compared to true cardinality (gray). (c) Estimates of the major and minor axes, A_i and a_i , respectively. Mean estimates (black) compared to true value (gray).

and subsequently Prediction Partition and EM Partition fails to compute informative partitions more often. This is a common cause of cardinality error. However, at the lower speed, the true target motion per time step is smaller, and the linear constant velocity prediction is good enough for Prediction Partition and EM Partition to compute informative partitions. The mean sum of weights at both speeds is shown in Figure 7b and Figure 7c, respectively. The filter can handle the maneuver, i.e. there is no cardinality error when the mean sum of weights is rounded to the nearest integer, when the targets are separated by $d \geq 21\text{m}$ at the higher speed. At the lower speed only $d \geq 2\text{m}$ separation is needed.

To conclude, the results show that when a constant velocity motion model is used, the presented extended target tracking filter can handle all scenarios except the

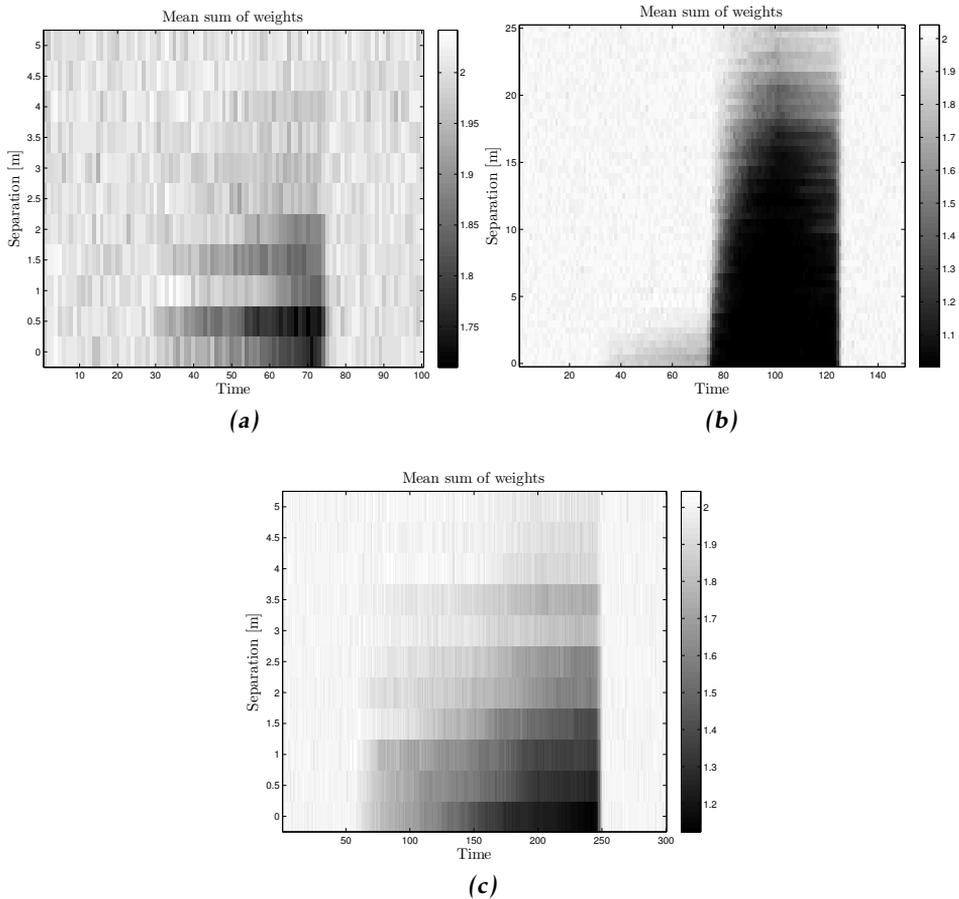


Figure 7: Mean sum of weights for closely spaced targets, at various separating distance. The true cardinality is 2 for all three plots. (a) Tracks in Figure 3b, note that if the mean sum of weights is rounded to the nearest integer there is no cardinality error. (b) Tracks in Figure 3d at speed 125m/s. At distances $d \geq 21\text{m}$ the cardinality is estimated correctly. (c) Tracks in Figure 3d at speed 62.5m/s. At distances $d \geq 2\text{m}$ the cardinality is estimated correctly.

ones where multiple spatially close targets are maneuvering quickly. In scenarios where the target maneuvers are dominant, the use of interacting multiple models (IMM) (Blom and Bar-Shalom, 1988) for motion prediction seems to be a reasonable solution, e.g. this was done in Section V of (Feldmann et al., 2011). The presented tracking filter can easily be generalized to use an IMM filter, however this was considered to be beyond the scope of this paper.

7 Experiment results

This section presents results from experiments based on data from a laser range sensor. Measurements were collected using a SICK LMS laser range sensor, which measures range every 0.5° over a 180° surveillance area. Ranges shorter than 13m were converted to (x, y) measurements using a polar to Cartesian transformation. The two data sets contain 411 and 400 laser range sweeps, respectively. Human targets entered the surveillance area at different times, and were measured by the sensor at waist level. There is no ground truth available for the data, however by examining the measurements the true cardinality can be observed.

These two data sets have previously been used in (Granström et al., 2012), where only the kinematical part of the target state is tracked. A comparison of the results for the presented GIWPHD filter to those for the ETGMPHD filter from (Granström et al., 2012) is performed.

7.1 Target tracking setup

The sensor's sampling time is $T_s = 0.2s$. The motion model parameters are set to $\theta = 1s$, $\Sigma = 2m/s^2$ and $\tau = 5s$. For the sensor used, new targets will appear somewhere along the edge of the semi circular surveillance area. Therefore, the birth PHD has $J_{b,k} = 20$ components located along the edge of the surveillance area, the intensity $D_k^b(\cdot)$ in the (x, y) dimension is shown in Figure 8. The birth components' weights are set to $w_{b,k}^{(j)} = 0.1/J_{b,k}$ and the inverse Wishart parameters are set to $\nu_{b,k}^{(j)} = 7$ and $V_{b,k}^{(j)} = \text{diag}([0.25^2 \ 0.1^2])$.

For the sensor used here, the expected number of target generated measurements γ varies rapidly with the distance between the target and the sensor. We have found that the correction weight update (33) is not sensitive to setting the corresponding filter parameter constant, however Sub-Partition needs a reasonable estimate of γ in order to compute a maximum likelihood estimate K of the number of targets that generated the measurements in a cell W . To facilitate this, in the Sub-Partition algorithm we have estimated γ by assuming that a 50cm wide target is located at the particular cell's centroid $\bar{\mathbf{z}}_k^W$. A width of 50cm roughly corresponds to the size of an average person, who is facing the sensor. This simple heuristic works well for the particular experiments presented here, however it remains within future work to design a method which does not rely on *a priori* information of the tracking scenario. A study of the extended target PHD filter's performance for incorrect values of the filter parameter corresponding to γ is given in (Granström et al., 2012).

7.2 Experiment with close targets

This data set contains 411 laser range scans. Two humans walked through the surveillance area, repeatedly moving towards and away from each other, both in the same direction and in the opposite direction. Thus, the data set contains situations where the targets are spatially close for both longer and shorter periods

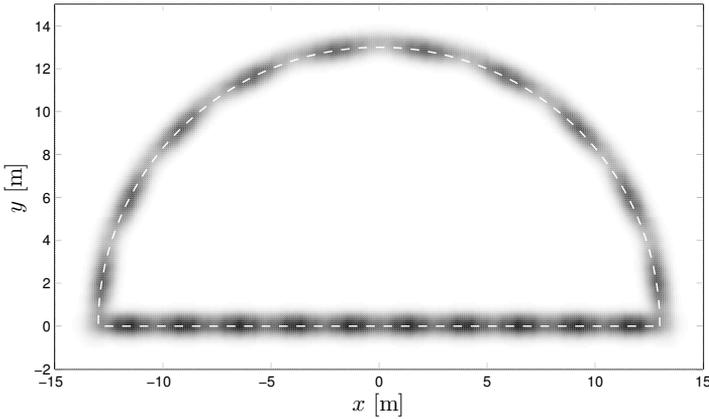


Figure 8: Birth PHD used in experiments. The dark areas are locations which the birth PHD models as likely locations for new targets to appear. The edge of the surveillance area is shown as a dashed white line.

of time. The positions of the extracted targets are shown in Figure 9a, the number of extracted targets are compared to the ground truth in Figure 9b and the sum of weights is shown in Figure 9c. There is no cardinality error for the entire length of the experiment, however at time 164 there is an unexpected increase in the sum of weights to 2.4. The sum of weights increases because the target generated measurements for one of the targets, at that time step, resemble two small clusters rather than one larger cluster. The GIWPHD filter interprets this as an increased likelihood of an additional target being present. These results are a small improvement over the results in (Granström et al., 2012), where the ETGMPHD filter underestimates the cardinality for three consecutive time steps when the targets are close and moving in the same direction.

7.3 Experiment with occlusion

This data set contains 400 laser range scans. Four humans walked through the surveillance area, however at most three humans were present at any one time. The first target stands still at the position $(x, y) \approx (0.4, 6)$ for most of the experiment. The second target walks behind the first target, causing the second target to be fully occluded (i.e. the second target is not measured), and also walks in front of the first target, causing the first target to be partially occluded (i.e. only parts of the first target are measured). With a constant probability of detection, the occlusion would cause target loss. To handle the occlusion without target loss, a state dependent probability of detection is implemented. The variable probability is based on a simple understanding of the sensor – objects that are located behind other objects cannot be measured by the sensor, therefore the probability of detection behind a target should be (close to) zero. The details of the variable probability of detection are given in Appendix A.2.

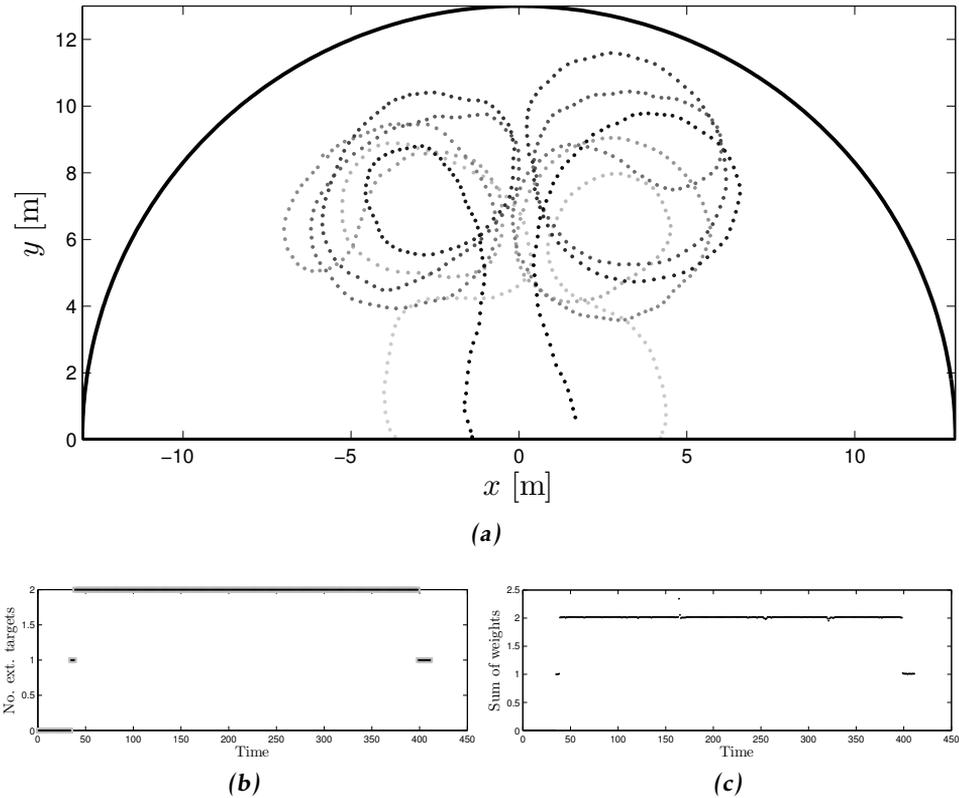


Figure 9: Results from the experiment with close targets (Section 7.2). (a) shows the positions of the extracted targets' kinematical states. Light gray points corresponds to earlier time steps, dark gray corresponds to later time steps. (b) shows the number of extracted targets in black, compared to the true cardinality in gray. (c) shows the sum of weights over time for the two experiments.

The positions of the extracted targets are shown in Figure 10a, the number of extracted targets are compared to the ground truth in Figure 10b and the sum of weights is shown in Figure 10c. At time 345, the time step when the fourth target enters the surveillance area, the cardinality is underestimated by 1. At this time step, the fourth target only generates one measurement, which the GIWPHD filter interprets as clutter.

These results are a considerable improvement over the results in (Granström et al., 2012), where the ETGMPHD filter underestimated the cardinality in two situations where two targets are spatially close, such that one target is partially occluded. In (Granström et al., 2012), a variable probability of detection was also used. However with the target centroid occluded, the probability of detection of the partially occluded target is incorrectly set close to zero, causing cardinal-

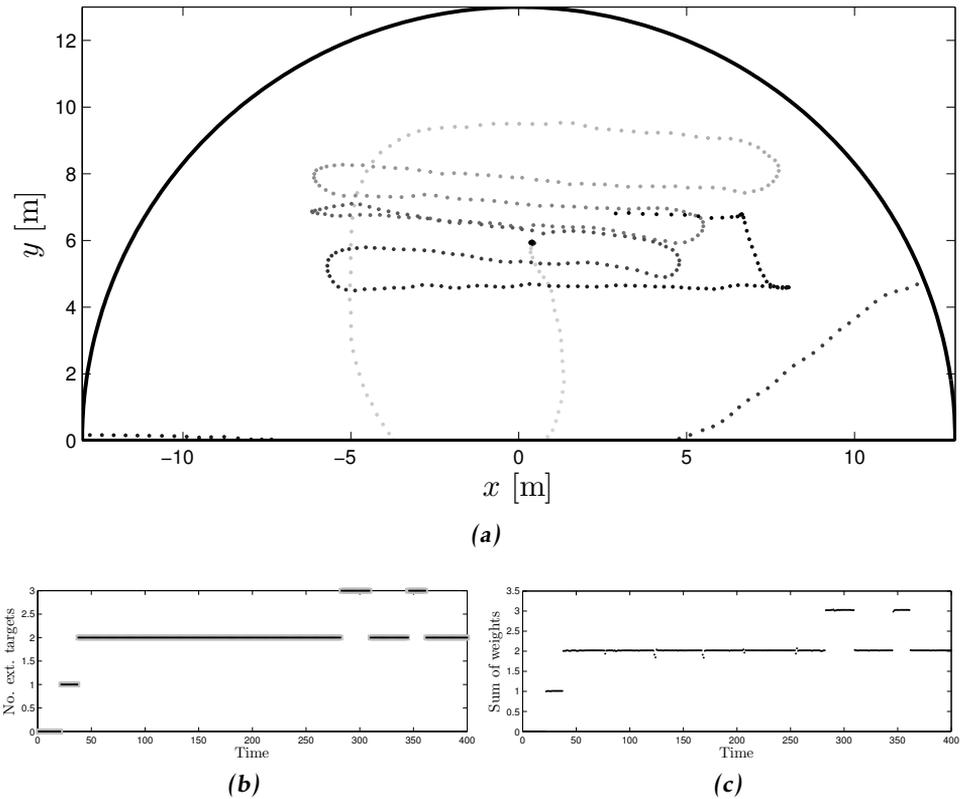


Figure 10: Results from the experiment with occlusion (Section 7.3). (a) shows the positions of the extracted targets' kinematical states. Light gray points corresponds to earlier time steps, dark gray corresponds to later time steps. (b) shows the number of extracted targets in black, compared to the true cardinality in gray. (c) shows the sum of weights over time for the two experiments.

ity error. With an estimate of the target extension, the partially occluded target can still be found to be detectable using the variable probability of detection in Appendix A.2. Thus, the experiment shows that for the data used, the GIWPHD filter can handle occlusion and spatially close targets simultaneously. The experiment also shows the benefit of estimating both the kinematical and extent states, compared to only estimating the kinematical state.

7.4 Discussion

The two experiments above are not an exhaustive evaluation of the GIWPHD filter, but they serve as a proof of concept and a potential application (e.g. person tracking for mobile robots). The comparison to the results in (Granström et al., 2012), in which the target extent is not explicitly estimated, support the intuitive

hypothesis that estimating the size of the extended targets improves the tracking performance. Initial steps have been taken toward including extension parameters in the target state in the ETGMPHD filter (Granström et al., 2011). Thus, more experiments that compare the ETGMPHD filter and the GIWPHD filter are needed, e.g. for data that contains more clutter than typical laser data does.

8 Conclusions and future work

This paper presented a PHD filter for multiple extended target tracking, in the presence of clutter and missed detection. The target extensions are modeled as random matrices (Koch, 2008), and a suitable likelihood function is derived. The PHD is approximated using Gaussian inverse Wishart distributions, and the assumptions necessary to obtain a computationally tractable PHD filter are presented. Two methods for measurement set partitioning are suggested to be added to the methods presented in (Granström et al., 2012). The first method is based on the predicted Gaussian inverse Wishart PHD components and the second method is based on the well known EM algorithm. Adding the two partitioning methods improves tracking of multiple targets of different sizes when they are spatially close. A simulation study confirms that the presented PHD filter can handle spatially close targets, with the exception of when the targets maneuver quickly. It is further shown that target spawning can be handled without the use of a specific spawning model. The spawning is instead implicitly handled by the measurement partitioning. A potential application, person tracking using laser range sensors, is presented. Two experiments show the benefit of estimating the size of the target extents, compared to only tracking the kinematical states of the target centroid, as is performed in (Granström et al., 2012).

The presented target tracking filter estimates the target extent as a random matrix, giving an elliptical extended target shape. Alternatively, the target ellipse could be explicitly parametrized, included in the state vector, and estimated with the kinematical states. Such an example is given in (Granström et al., 2011), where elliptical shapes are tracked using a laser range sensor and a ETGMPHD filter. Ellipse tracking is also performed using random hypersurface models in (Baum et al., 2010b), a comparison between random matrices and random hypersurface models for the single target case is given in (Baum et al., 2010a). A comparison of the presented GIWPHD filter, and elliptic random hypersurface models using the ETGMPHD filter, could be interesting for the multiple target case.

The importance of measurement set partitioning was highlighted in the paper, and the case of close targets maneuvering quickly was shown to be difficult to handle. This could possibly be improved through the use of additional partitioning methods, or via modeling different type of motions using an IMM type filter. A prediction model that allows transformations of the extension, e.g. rotations, would possibly improve the filter performance.

Target spawning is not explicitly modeled in this work, however, it could be handled implicitly via the partitioning methods used. The targets must be separated

sufficiently for the spawning to be detected, and the spawning event is therefore detected with a small time delay. It is not obvious how a Gaussian inverse Wishart distribution can be split into two Gaussian inverse Wishart distributions, however devising such a method could possibly improve performance for target spawning events.

A heuristic for determining the parameter γ in the Sub-Partition algorithm was suggested. A method which does not rely on either assumptions or *a priori* knowledge of the tracking scenario would be useful in the general case. Finally, an improved method for merging of GIW components is needed.

A Appendix

A.1 Derivation of the correction

Under the measurement model (5), the likelihood of n measurements \mathbf{z}_j is

$$\begin{aligned} \prod_{j=1}^n \mathcal{N}(\mathbf{z}_j; (H \otimes \mathbf{I}_d) \mathbf{x}, X) &= \prod_{j=1}^n \mathcal{N}(\mathbf{z}_j; \tilde{H} \mathbf{x}, X) \\ &= (2\pi)^{-nd/2} |X|^{-n/2} \text{etr} \left(-\frac{1}{2} \left(\sum_{j=1}^n (\mathbf{z}_j - \tilde{H} \mathbf{x})(\mathbf{z}_j - \tilde{H} \mathbf{x})^\top \right) X^{-1} \right), \end{aligned} \quad (40)$$

where $\text{etr}(\cdot) = \exp(\text{Tr}(\cdot))$ is exponential trace. Define the centroid measurement as $\bar{\mathbf{z}} \triangleq \frac{1}{n} \sum_{j=1}^n \mathbf{z}_j$ and the scatter matrix as $Z \triangleq \sum_{j=1}^n (\mathbf{z}_j - \bar{\mathbf{z}})(\mathbf{z}_j - \bar{\mathbf{z}})^\top$, and rewrite the summation as

$$\sum_{j=1}^n (\mathbf{z}_j - \tilde{H} \mathbf{x})(\mathbf{z}_j - \tilde{H} \mathbf{x})^\top = Z + n(\bar{\mathbf{z}} - \tilde{H} \mathbf{x})(\bar{\mathbf{z}} - \tilde{H} \mathbf{x})^\top. \quad (41)$$

Inserting (41) into (40) gives

$$\prod_{j=1}^n \mathcal{N}(\mathbf{z}_j; \tilde{H} \mathbf{x}, X) \quad (42a)$$

$$= (2\pi)^{-nd/2} |X|^{-n/2} \text{etr} \left(-\frac{1}{2} Z X^{-1} \right) \text{etr} \left(-\frac{1}{2} (\bar{\mathbf{z}} - \tilde{H} \mathbf{x})(\bar{\mathbf{z}} - \tilde{H} \mathbf{x})^\top \left(\frac{X}{n} \right)^{-1} \right) \quad (42b)$$

$$= (2\pi)^{-(n-1)d/2} |X|^{-(n-1)/2} n^{-d/2} \text{etr} \left(-\frac{1}{2} Z X^{-1} \right) \mathcal{N} \left(\bar{\mathbf{z}}; \tilde{H} \mathbf{x}, \frac{X}{n} \right) \quad (42c)$$

$$= \mathcal{L}_{\text{aux}} \mathcal{N} \left(\bar{\mathbf{z}}; \tilde{H} \mathbf{x}, \frac{X}{n} \right). \quad (42d)$$

Now, let the predicted target distribution be

$$\mathcal{N}(\mathbf{x}; m, P \otimes X) \mathcal{IW}(X; \nu, V). \quad (43)$$

The product of the measurement likelihood (42) and the predicted distribution (43) is

$$\begin{aligned} & \mathcal{L}_{\text{aux}} \mathcal{N}\left(\bar{\mathbf{z}}; \tilde{H}\mathbf{x}, \frac{X}{n}\right) \mathcal{N}(\mathbf{x}; m, P \otimes X) \mathcal{IW}(X; \nu, V) \\ &= \mathcal{N}(\mathbf{x}; m_+, P_+ \otimes X) \mathcal{N}(\bar{\mathbf{z}}; \tilde{H}m, SX) \mathcal{IW}(X; \nu, V) \mathcal{L}_{\text{aux}} \end{aligned} \quad (44)$$

where we have

$$m_+ = m + (K \otimes \mathbf{I}_d)(\bar{\mathbf{z}} - \tilde{H}m), \quad (45a)$$

$$P_+ = P - KSK^T. \quad (45b)$$

with innovation factor $S = HPH^T + 1/n$ and gain matrix $K = PH^T S^{-1}$.

This result is easy to derive using the product formula for Gaussian distributions, and using some basic properties of the Kronecker product. We thus have the corrected Gaussian distribution with mean and covariance (45), multiplied with

$$\mathcal{N}(\bar{\mathbf{z}}; \tilde{H}m, SX) \mathcal{IW}(X; \nu, V) \mathcal{L}_{\text{aux}} \quad (46a)$$

$$\begin{aligned} &= (2\pi)^{-d/2} |SX|^{-1/2} \frac{|V|^{v/2} |X|^{-(v+d+1)/2}}{2^{vd/2} \Gamma_d(v/2)} \text{etr}\left(-\frac{1}{2}(\bar{\mathbf{z}} - \tilde{H}m)(\bar{\mathbf{z}} - \tilde{H}m)^T (SX)^{-1}\right) \\ &\quad \times \text{etr}\left(-\frac{1}{2}VX^{-1}\right) \text{etr}\left(-\frac{1}{2}ZX^{-1}\right) (2\pi)^{-(n-1)d/2} |X|^{-(n-1)/2} n^{-d/2} \end{aligned} \quad (46b)$$

$$\begin{aligned} &= (2\pi)^{-nd/2} (nS)^{-d/2} \frac{|V|^{v/2}}{2^{vd/2} \Gamma_d(v/2)} \frac{2^{(v+n)d/2} \Gamma_d((v+n)/2)}{|V+N+Z|^{(v+n)/2}} \\ &\quad \times \frac{|V+N+Z|^{(v+n)/2} |X|^{-(v+n+d+1)/2}}{2^{(v+n)d/2} \Gamma_d((v+n)/2)} \text{etr}\left(-\frac{1}{2}(V+N+Z)X^{-1}\right) \end{aligned} \quad (46c)$$

$$= (\pi^n nS)^{-d/2} \frac{|V|^{v/2}}{|V+N+Z|^{(v+n)/2}} \frac{\Gamma_d((v+n)/2)}{\Gamma_d(v/2)} \mathcal{IW}(X; \nu_+, V_+) \quad (46d)$$

$$= \mathcal{L} \times \mathcal{IW}(X; \nu_+, V_+) \quad (46e)$$

where we have $N = S^{-1}(\bar{\mathbf{z}} - \tilde{H}m)(\bar{\mathbf{z}} - \tilde{H}m)^T$ and

$$\nu_+ = \nu + n, \quad (47a)$$

$$V_+ = V + N + Z, \quad (47b)$$

and the likelihood function \mathcal{L} is defined as

$$\mathcal{L} = (\pi^n nS)^{-d/2} \frac{|V|^{v/2}}{|V_+|^{v_+/2}} \frac{\Gamma_d(v_+/2)}{\Gamma_d(v/2)}. \quad (48)$$

The likelihood function can be shown to be proportional to a generalized matrix variate beta type II distribution (Gupta and Nagar, 2000). We have thus shown how the likelihood of n measurements \mathbf{z}_j multiplied with a predicted GIW distribution can be rewritten as a corrected GIW distribution multiplied with a likelihood function.

A.2 Variable probability of detection for the laser range sensor

The variable probability of detection used here is similar to the one presented in (Granström et al., 2012), however it relies on less assumptions, and instead utilizes the estimated target extensions. The idea is to decrease the probability of detection behind (i.e. at larger range from the sensor) each GIW component. In doing so, the function considers the component weight, the size of the estimated extension and the uncertainty in bearing (i.e. the polar angle from the sensor to the component).

For a given point (x, y) in the surveillance area, the probability of detection is computed as

$$p_D(x, y) = \max(p_{D,\min}, p_{D,0} - \tilde{p}_D), \quad (49a)$$

$$\tilde{p}_D = \sum_{i:r>r^{(i)}} w^{(i)} \max(\mathcal{G}_1, \mathcal{G}_2, \mathcal{A}), \quad (49b)$$

$$\mathcal{G}_g = \exp\left(-\frac{(\varphi - \varphi^{(i)} + (-1)^g 2\sigma_{\varphi,e})^2}{0.01\sigma_{\varphi,p}}\right), \quad (49c)$$

$$\mathcal{A} = |\varphi - \varphi^{(i)}| < 2\sigma_{\varphi,e}, \quad (49d)$$

where

- $p_{D,\min}$ is the minimum probability of detection value allowed;
- $p_{D,0}$ is the nominal probability of detection of targets which are not occluded;
- $r = \sqrt{x^2 + y^2}$ and $\varphi = \tan^{-1}(y/x)$ is the range and bearing to the point (x, y) ;
- $w^{(i)}$, $r^{(i)}$ and $\varphi^{(i)}$ is the weight, range and bearing to the i :th GIW component's kinematical state;
- $\sigma_{\varphi,e}$ is the cross range size of the i :th component, computed by Cartesian to polar conversion of the extent matrix estimated as in (14b);
- $\sigma_{\varphi,p}$ is the bearing standard deviation of the i :th components kinematical state, computed by Cartesian to polar conversion of the position uncertainty estimated as in (14a).

To obtain the probability of detection for a GIW component $\xi_{k|k-1}^{(i)}$, the ellipsoid corresponding to two standard deviations of the estimated extent (14b) is discretized into points (x, y) , and for each discrete point along the extent a probability of detection is computed. The probability of detection for the GIW component $p_D(\xi_{k|k-1}^{(i)})$ is then given as the maximum of the probabilities computed along the discretized extent. In comparison, the variable probability of detection is computed only for the kinematical state position in (Granström et al., 2012). Taking the maximum probability along the extent is what enables the GIWPHD filter to handle partial target occlusion.

Table 2: Pseudo-code for the Gaussian inverse Wishart PHD filter

1:	input: Sequence of measurement sets $\{\mathbf{Z}_k\}_{k=1}^K$.
2:	initialize: Set $J_{0 0} = 0$.
3:	for $k = 1, \dots, K$ do
4:	Compute measurement set partition $\{p_p\}_{p=1}^P$, see Paper B and Section 5.
5:	Predict and construct correction components, Table 3.
6:	Correct, Table 4.
7:	Prune and merge, Table 1.
8:	Extract estimated target set, Table 5.
9:	end for
10:	output: Sequence of estimated target sets $\{\hat{\mathbf{X}}_k\}_{k=1}^K$

A.3 Pseudo-code

The main filter recursion is given in Table 2, prediction and construction of correction components is given in Table 3, and correction is given in Table 4. The pruning and merging scheme is given in Table 1, and target extraction is given in Table 5. Note that pruning and merging, and target extraction, is performed similarly to Vo and Ma (2006).

A.4 Implementation issues

When computing the corrected weight (Table 4, Line 14), the likelihood often includes ratios of large numbers, leading to numerical overflow. Because of this, computing the log-likelihood is recommended, and then updating the weight with the exponential of the log likelihood. Similarly, the quantities $d_{W_w^p}$ (Table 4, Line 16) are often large, leading to numerical overflow when ω_{p_p} (Table 4, Line 19) is computed. A remedy is to store the log partition weights

$$\tilde{\omega}_{p_p} = \log(\omega_{p_p}) = \sum_{w=1}^{|p_p|} \log(d_{W_w^p}), \quad (50)$$

and to normalize the log partition weights (Table 4, Line 21) as follows,

$$\hat{\omega}_{p_p} = \tilde{\omega}_{p_p} - \log\left(\sum_{p'=1}^P \omega_{p_{p'}}\right) \quad (51a)$$

$$= \tilde{\omega}_{p_p} - \left[\tilde{\omega}_{p_1} + \log\left(1 + \sum_{p'=2}^P e^{\tilde{\omega}_{p_{p'}} - \tilde{\omega}_{p_1}}\right) \right] \quad (51b)$$

for $p = 1, \dots, P$. The partition weights are then given as $\omega_{p_p} = e^{\hat{\omega}_{p_p}}$ for $p = 1, \dots, P$.

Table 3: Pseudo-code for GIWPHD filter prediction and correction components

1: input: GIW components $\left\{ w_{k-1 k-1}^{(j)}, \xi_{k-1 k-1}^{(j)} \right\}_{j=1}^{J_{k-1 k-1}}$, and set of measurement set partitions $\left\{ \mathbb{P}_p \right\}_{p=1}^P$. 2: $i = 0$ 3: for $j = 1, \dots, J_{b,k}$ do 4: $i \leftarrow i + 1$ 5: $w_{k k-1}^{(i)} \leftarrow w_{b,k}^{(j)}, \quad \xi_{k k-1}^{(i)} \leftarrow \xi_{b,k}^{(j)}$ 6: end for 7: for $j = 1, \dots, J_{k-1 k-1}$ do 8: $i \leftarrow i + 1$ 9: $w_{k k-1}^{(i)} \leftarrow p_S w_{k-1 k-1}^{(j)}$ 10: $m_{k k-1}^{(i)} \leftarrow (F_{k k-1} \otimes \mathbf{I}_d) m_{k-1 k-1}^{(j)}, \quad P_{k k-1}^{(i)} \leftarrow F_{k k-1} P_{k k-1}^{(j)} F_{k k-1}^T + \mathbf{Q}_{k k-1}$ 11: $v_{k k-1}^{(i)} \leftarrow e^{-T_s/\tau} v_{k-1 k-1}^{(j)}, \quad V_{k k-1}^{(i)} \leftarrow \frac{v_{k k-1}^{(i)-d-1}}{v_{k-1 k-1}^{(j)-d-1}} V_{k-1 k-1}^{(j)}$ 12: end for 13: $J_{k k-1} \leftarrow i$ 14: for $p = 1, \dots, P$ do 15: for $w = 1, \dots, \mathbb{P}_p $ do 16: $\bar{\mathbf{z}}_k^{W_w^p} \leftarrow \frac{1}{ W_w^p } \sum_{\mathbf{z}_k^{(i)} \in W_w^p} \mathbf{z}_k^{(i)}, \quad Z_k^{W_w^p} \leftarrow \sum_{\mathbf{z}_k^{(i)} \in W_w^p} (\mathbf{z}_k^{(i)} - \bar{\mathbf{z}}_k^{W_w^p})(\mathbf{z}_k^{(i)} - \bar{\mathbf{z}}_k^{W_w^p})^T$ 17: end for 18: end for 19: for $j = 1, \dots, J_{k k-1}$ do 20: $\hat{K}_{k k-1}^{(j)} \leftarrow P_{k k-1}^{(j)} H_k^T, \quad \hat{S}_{k k-1}^{(j)} \leftarrow H_k \hat{K}_{k k-1}^{(j)}, \quad \hat{\mathbf{z}}_{k k-1}^{(j)} \leftarrow (H_k \otimes \mathbf{I}_d) m_{k k-1}^{(j)}$ 21: end for 22: output: GIW components $\left\{ w_{k k-1}^{(j)}, \xi_{k k-1}^{(j)} \right\}_{j=1}^{J_{k k-1}}$, and correction components $\left\{ \left\{ \bar{\mathbf{z}}_k^{W_w^p}, Z_k^{W_w^p} \right\}_{w=1}^{ \mathbb{P}_p } \right\}_{p=1}^P$ and $\left\{ \hat{K}_{k k-1}^{(j)}, \hat{S}_{k k-1}^{(j)}, \hat{\mathbf{z}}_{k k-1}^{(j)} \right\}_{j=1}^{J_{k k-1}}$
--

A.5 Computational complexity analysis

Complexity of common operations

The complexity of some common matrix and vector operations is given in Table 6. In addition, computing the inversion and the determinant of a $d \times d$ matrix V both have approximate computational complexity $\mathcal{O}(d^3)$.

Table 4: Pseudo-code for GIWPHD filter correction

1:	input: GIW components $\left\{ w_{k k-1}^{(j)}, \xi_{k k-1}^{(j)} \right\}_{j=1}^{J_{k k-1}}$, partitions $\{p_p\}_{p=1}^P$, and correction components $\left\{ \left\{ \bar{z}_k^{W_w^p}, Z_k^{W_w^p} \right\}_{w=1}^{ p_p } \right\}_{p=1}^P$ and $\left\{ \hat{K}_{k k-1}^{(j)}, \hat{S}_{k k-1}^{(j)}, \hat{z}_{k k-1}^{(j)} \right\}_{j=1}^{J_{k k-1}}$.
2:	for $j = 1, \dots, J_{k k-1}$ do
3:	$w_{k k}^{(j)} \leftarrow \left(1 - \left(1 - e^{-\gamma^{(j)}} \right) p_D^{(j)} \right) w_{k k-1}^{(j)}, \quad \xi_{k k}^{(j)} \leftarrow \xi_{k k-1}^{(j)}$.
4:	end for
5:	$\ell = 0$
6:	for $p = 1, \dots, P$ do
7:	for $w = 1, \dots, p_p $ do
8:	$\ell \leftarrow \ell + 1$
9:	for $j = 1, \dots, J_{k k-1}$ do
10:	$S \leftarrow \hat{S}_{k k-1}^{(j)} + \frac{1}{ W_w^p }, \quad K \leftarrow \hat{K}_{k k-1}^{(j)} S^{-1}$
11:	$\varepsilon \leftarrow \bar{z}_k^{W_w^p} - \hat{z}_{k k-1}^{(j)}, \quad N \leftarrow S^{-1} \varepsilon \varepsilon^T$
12:	$m_{k k}^{(j+J_{k k-1}\ell)} \leftarrow m_{k k-1}^{(j)} + (K \otimes \mathbf{I}_d) \varepsilon, \quad P_{k k}^{(j+J_{k k-1}\ell)} \leftarrow P_{k k-1}^{(j)} - K S K^T$
13:	$v_{k k}^{(j+J_{k k-1}\ell)} \leftarrow v_{k k-1}^{(j)} + W_w^p , \quad V_{k k}^{(j+J_{k k-1}\ell)} \leftarrow V_{k k-1}^{(j)} + N + Z_k^{W_w^p}$
14:	$w_{k k}^{(j+J_{k k-1}\ell)} \leftarrow \frac{e^{-\gamma^{(j)}} (\gamma^{(j)})^{ W_w^p } p_D^{(j)}}{\beta_{FA,k}^{ W_w^p } (\pi^{ W_w^p } W_w^p S)^{d/2}} \frac{ V_{k k-1}^{(j)} ^{y_{k k-1}^{(j)/2}} \Gamma_d(v_{k k}^{(j+J_{k k-1}\ell)}/2)}{ V_{k k}^{(j+J_{k k-1}\ell)} ^{y_{k k}^{(j+J_{k k-1}\ell)}/2} \Gamma_d(v_{k k-1}^{(j)}/2)} w_{k k-1}^{(j)}$
15:	end for
16:	$d_{W_w^p} \leftarrow \delta_{ W_w^p ,1} + \sum_{j=1}^{J_{k k-1}} w_{k k}^{(j+J_{k k-1}\ell)}$
17:	$w_{k k}^{(j+J_{k k-1}\ell)} \leftarrow \frac{w_{k k}^{(j+J_{k k-1}\ell)}}{d_{W_w^p}}$ for $j = 1, \dots, J_{k k-1}$
18:	end for
19:	$\omega_{p_p} \leftarrow \prod_{w=1}^{ p_p } d_{W_w^p}$
20:	end for
21:	$J_{k k} \leftarrow J_{k k-1} (\ell + 1), \quad J_{\text{aux}} \leftarrow J_{k k-1}, \quad \omega_{p_p} \leftarrow \frac{\omega_{p_p}}{\sum_{p'=1}^P \omega_{p_p'}}$ for $p = 1, \dots, P$
22:	for $p = 1, \dots, P$ do
23:	$w_{k k}^{(j+J_{\text{aux}})} \leftarrow w_{k k}^{(j+J_{\text{aux}})} \omega_{p_p}$, for $j = 1, \dots, J_{k k-1} p_p $, $J_{\text{aux}} \leftarrow J_{\text{aux}} + J_{k k-1} p_p $
24:	end for
25:	output: GIW components $\left\{ w_{k k}^{(j)}, \xi_{k k}^{(j)} \right\}_{j=1}^{J_{k k}}$

Assumptions and approximations

To simplify the notation, let $J = J_{k-1|k-1}$, $J_b = J_{b,k}$ and $J_+ = J_{k-1|k-1} = J_{k-1|k-1} + J_{b,k}$. To simplify the analysis, the following assumptions are made;

Table 5: Pseudo-code for GIWPHD filter target extraction

```

1: input: GIW components  $\left\{w_{k|k}^{(j)}, \xi_{k|k}^{(j)}\right\}_{j=1}^{J_{k|k}}$ .
2:  $\hat{\mathbf{X}}_k = \emptyset$ 
3: for  $j = 1, \dots, J_{k|k}$  do
4:   if  $w_{k|k}^{(j)} > 0.5$  then
5:     Compute  $\hat{X}_{k|k}^{(j)}$ , see Granström and Orguner (2012).
6:      $\hat{\mathbf{X}}_k \leftarrow \hat{\mathbf{X}}_k \cup \left(m_{k|k}^{(j)}, \hat{X}_{k|k}^{(j)}\right)$ 
7:   end if
8: end for
9: output: Estimated set of targets  $\hat{\mathbf{X}}_k$ 

```

1. The estimated target cardinality is approximately correct, i.e. $\hat{N}_{x,k} \approx N_{x,k}$.
2. The number of GIW components are approximately equal to the number of targets, i.e. $J \approx N_{x,k}$.
3. The true target Poisson rate γ is equal for all targets. The number of measurements is $N_{z,k} \approx \gamma N_{x,k} + \lambda_k$, where λ_k is the mean number of clutter measurements.
4. After partitioning, each cell W with target generated measurements has cardinality approximately equal to the Poisson rate γ , i.e. $|W| \approx \gamma$.
5. After the correction step, there are approximately $N_{x,k}$ clusters with GIW components, and each cluster contains approximately J_+ GIW components. Thus, in the pruning and merging step, we have $|L| \approx J_+$, where L is the set of GIW components that are merged into one component.

Table 6: Complexity of common operations, in terms of multiplications (mult) and summations (sum).

Input	Operation	Mult	Sum	Complexity
$A(m \times n), B(n \times p)$	AB	mnp	$m(n-1)p$	$m(2n-1)p$
$A(m \times n), B(p \times q)$	$A \otimes B$	$mnpq$	0	$mnpq$
$A(m \times n), B(p \times q)$ $C(nq \times t)$	$(A \otimes B)C$	$mnpq$ $\times(t+1)$	$mp(nq-1)t$	$mp[(2nq-1)t+nq]$
$A_i(p \times q)$	$\sum_{i=1}^n A_i$	0	$(n-1)pq$	$(n-1)pq$
$x_i(d \times 1), y_i(d \times 1)$	$x_i - y_i$	0	d	d
$z_i(d \times 1)$	$\sum_{i=1}^n z_i z_i^T$	nd^2	$(n-1)d^2$	$(2n-1)d^2$
$F(n \times n), P(n \times n)$	FPF^T	$2n^3$	$2(n^3 - n^2)$	$4n^3 - 2n^2$

Prediction and construction of correction components

The complexity of prediction and construction of correction components is given in Table 7. Predicting each of the J components has approximate complexity

$$\mathcal{O}(J(3n_x^2 + 4s^3 + d^2 - n_x - s^2 + 7)) \approx \mathcal{O}(Jn_x^2). \quad (52)$$

Constructing the centroid measurements and scatter matrices has approximate complexity

$$\mathcal{O}\left(\left(\sum_{p=1}^P |p_p|\right)\left(\gamma(2d^2 + d) - d^2 + d + 2\right)\right) \approx \mathcal{O}\left(\gamma d^2 \sum_{p=1}^P |p_p|\right). \quad (53)$$

Constructing the gain matrices, innovation factors and innovation vectors has approximate complexity

$$\mathcal{O}\left((J + J_b)(3n_x d + 2s^2 + s - d - 1)\right) \approx \mathcal{O}\left((J + J_b)n_x^2\right). \quad (54)$$

Thus, the overall complexity of prediction and construction of correction components is approximately

$$\mathcal{O}\left(J_+ n_x^2 + \gamma d^2 \sum_{p=1}^P |p_p|\right). \quad (55)$$

Table 7: Complexity of prediction and correction components

Operation	Multiplications	Summations	Complexity
$p_s w$	1	0	1
$(F \otimes \mathbf{I}_d)m$	$s^2 d^2 + s^2 d^2$	$sd(sd - 1)$	$3n_x^2 - n_x$
$FPF^T + Q$	$2s^3$	$2s^2(s - 1) + s^2$	$4s^3 - s^2$
$e^{-T_s/\tau} v$	–	–	$\mathcal{O}(3)$
$\frac{v-d-1}{v-d-1} V$	$d^2 + 1$	4	$d^2 + 4$
$\frac{1}{ W } \sum \mathbf{z}$	2	$(\gamma - 1)d$	$(\gamma - 1)d + 2$
$\sum (\mathbf{z} - \mathbf{z})(\mathbf{z} - \mathbf{z})^T$	γd^2	$2d + (\gamma - 1)d^2$	$(2\gamma - 1)d^2 + 2d$
PH	s^2	$s(s - 1)$	$2s^2 - s$
HK	s	$s - 1$	$2s - 1$
$(H \otimes \mathbf{I}_d)m$	$sd^2 + sd^2$	$d(sd - 1)$	$(3n_x - 1)d$

Correction

The complexity of correction is given in Table 8. The correction update of the GIW components has approximate complexity

$$\mathcal{O}\left(\left(J_+ \sum_{p=1}^P |p_p|\right)(3n_x d + d^3 + 3d^2 + d + s + 6)\right) \approx \mathcal{O}\left(n_x^2 J_+ \sum_{p=1}^P |p_p|\right). \quad (56)$$

Computing the cell weights δ_W and partition weights ω_p has approximate complexity

$$\mathcal{O}\left((J_+ + 1) \sum_{p=1}^P |p_p| + \sum_{p=1}^P |p_p|\right) \approx \mathcal{O}\left(J_+ \sum_{p=1}^P |p_p|\right). \quad (57)$$

Normalizing the partition weights and updating the GIW components weights has approximate complexity $\mathcal{O}(P + 2 + 3P) \approx \mathcal{O}(P)$. Thus, the overall complexity of the correction is approximately

$$\mathcal{O}\left(n_x^2 J_+ \sum_{p=1}^P |p_p|\right). \quad (58)$$

Table 8: Complexity of correction

Operation	Multiplications	Summations	Complexity
$S + 1/ W $	1	1	2
KS^{-1}	$1 + s$	0	$s + 1$
$\mathbf{z} - \mathbf{z}$	0	d	d
$S^{-1} \varepsilon \varepsilon^T$	$1 + d^2 + 1$	0	$d^2 + 1$
$m + (K \otimes \mathbf{I}_d) \varepsilon$	$sd^2 + sd^2$	$sd(d - 1) + n_x$	$3n_x d$
$P - KSK^T$	$s^2 + 1$	s^2	$2s^2 + 1$
$v + W $	0	1	1
$V + N + Z$	0	$2d^2$	$2d^2$
weight update	–	–	$\mathcal{O}(d^3)$
$\delta + \sum w$	0	$J_+ + 1$	$J_+ + 1$
$\prod d_W$	$ p_p - 1$	0	$ p_p - 1$
$J(\ell + 1)$	1	1	2
$\frac{\omega_p}{\sum \omega_p}$	1	$P - 1$	P
$w\omega$	1	0	1
$J + J p $	1	1	2

Pruning and merging

The complexity of merging is given in Table 9. Determining whether or not components i and j should be merged has approximate complexity $\mathcal{O}(n_x^3)$. There are approximately $N_{x,k} J_+$ components remaining after the correction step. In the worst case, each component has to be compared to all other components, i.e. $\mathcal{O}(N_{x,k}^2 J_+^2)$ comparisons. Thus, the worst case complexity of the merging is approximately

$$\mathcal{O}(N_{x,k}^2 J_+^2 n_x^3). \quad (59)$$

Table 9: Complexity of pruning and merging

Operation	Multiplications	Summations	Complexity
\hat{P}	$s^2 d^2 + 2$	3	$n_x^2 + 5$
Merge i and j ?	–	–	$\mathcal{O}(n_x^3)$
$\sum w$	0	$J_+ - 1$	$J_+ - 1$
$\frac{1}{w} \sum wm$	$J_+ n_x + n_x + 1$	$(J_+ - 1)n_x$	$2J_+ n_x + 1$
$\frac{1}{w} \sum wP$	$J_+ s^2 + s^2 + 1$	$(J_+ - 1)s^2$	$2J_+ s^2 + 1$
$\frac{1}{w} \sum wv$	$J_+ + 2$	$J_+ - 1$	$2J_+ + 1$
$\frac{1}{w} \sum wV$	$J_+ d^2 + d^2 + 1$	$(J_+ - 1)d^2$	$2J_+ d^2 + 1$

Target extraction

Computing the extension estimate \hat{X} has approximate complexity $\mathcal{O}(d^2)$. Under the assumption that the target cardinality estimate is approximately correct, the complexity of target extraction is approximately

$$\mathcal{O}(N_{x,k} d^2). \quad (60)$$

Partitioning the measurement set

For Distance Partitioning, creating the distance matrix requires $3N_{z,k}(N_{z,k} - 1)$ multiplications and summations. The distance matrix must then, in the *worst case*, be queried for each measurement for each of the P partitions, i.e. $N_{z,k}P$ times. Note that it is difficult to give an estimate of how many partitions are created in Distance Partitioning, because P depends on the particular measurement set that is being partitioned.

However, using Distance Partitioning gives at most $N_{z,k}$ unique partitions, thus a *worst case* upper limit for P is $N_{z,k}$. The *worst case* complexity of Distance Partitioning is thus approximately $\mathcal{O}(N_{z,k}^4)$. The *worst case* $P = N_{z,k}$ gives

$$\sum_{p=1}^P |p_p| = \sum_{p=1}^{N_{z,k}} p = \frac{N_{z,k}(N_{z,k} + 1)}{2} \approx \mathcal{O}(N_{z,k}^2). \quad (61)$$

The complexity of the EM algorithm for Gaussian mixtures is given in Table 10. The computational complexity of one iteration of the EM algorithm for Gaussian mixtures is approximately $\mathcal{O}(N_{x,k}N_{z,k}d^2 + n_x^3)$. On average, in our simulations and experiments, convergence is reached in 4 iterations.

Thus, the *worst case* complexity of partitioning the measurement set is approximately

$$\mathcal{O}(N_{z,k}(N_{x,k}d^2 + N_{z,k}^3) + n_x^3). \quad (62)$$

Table 10: Complexity of the EM algorithm

Operation	Multiplications	Summations	Complexity
$p(\mathbf{z}_j m_i P_i)$	–	–	$\mathcal{O}(d^3)$
$\gamma_i(\mathbf{z}_j)$	$N_{x,k} + 3$	$N_{x,k} - 1$	$2N_{x,k} + 2$
m_i	$N_{x,k}(N_{z,k} + 2)$	$N_{x,k}N_{z,k}$	$2N_{x,k}N_{z,k} + 2N_{x,k}$
P_i	$N_{x,k}(N_{z,k}(d^2 + 1) + 2)$	$N_{x,k}[2d + d^2(N_{z,k} - 1)]$	$2N_{x,k}N_{z,k}d^2 + N_{x,k} \times (N_{z,k} - d^2 + 2d + 2)$
π_i	$N_{x,k}$	$N_{x,k}(N_{z,k} - 1)$	$N_{x,k}N_{z,k}$
log-lik	–	–	$\mathcal{O}(n_x^3)$

Overall complexity

The worst case overall complexity of one time step is approximately

$$\begin{aligned} & \mathcal{O} \left(J_+ n_x^2 + \gamma d^2 \sum_p |\mathbf{p}_p| + J_+ n_x^2 \sum_p |\mathbf{p}_p| + N_{x,k}^2 J_+^2 n_x^3 \right. \\ & \quad \left. + N_{x,k} d^2 + N_{z,k} N_{x,k} d^2 + N_{z,k}^4 + n_x^3 \right) \end{aligned} \quad (63)$$

$$\approx \mathcal{O} \left((J_+ n_x^2 + \gamma d^2) \sum_p |\mathbf{p}_p| + N_{x,k} N_{z,k} d^2 + J_+^2 N_{x,k}^2 n_x^3 + N_{z,k}^4 \right). \quad (64)$$

Inserting $J_+ = N_{x,k} + J_b$ and $N_{z,k} = \gamma N_{x,k} + \lambda_k$ into (64) gives the worst case overall complexity

$$\begin{aligned} & \mathcal{O} \left(((N_{x,k} + J_b) n_x^2 + \gamma d^2) \sum_{p=1}^P |\mathbf{p}_p| + (\gamma N_{x,k} + \lambda_k) N_{x,k} d^2 \right. \\ & \quad \left. + (N_{x,k} + J_b)^2 N_{x,k}^2 n_x^3 + P (\gamma N_{x,k} + \lambda_k)^3 \right). \end{aligned} \quad (65)$$

Bibliography

- Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- M. Baum and U. D. Hanebeck. Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 338–345, Chicago, IL, USA, July 2011.
- M. Baum, M. Feldmann, D. Fränken, U. D. Hanebeck, and J. W. Koch. Extended object and group tracking: A comparison of random matrices and random hypersurface models. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Leipzig, Germany, October 2010a.
- M. Baum, B. Noack, and U. D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010b.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, aug 1988.
- D.E. Clark and S. Godsill. Group target tracking with the gaussian mixture probability hypothesis density filter. In *International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 149–154, Melbourne, Australia, December 2007.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012. doi: 10.1109/TSP.2012.2212888.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics. Chapman & Hall, 2000. ISBN 9781584880462.
- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- J. W. Koch and M. Feldmann. Cluster tracking under kinematical constraints using random matrices. *Robotics and Autonomous Systems*, 57(3):296–309, March 2009.
- C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011.
- R. P. S. Mahler. Multitarget Bayes filtering via first-order multi target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.
- R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):389–395, December 2003.

- D.J. Salmond and N.J. Gordon. Group and extended object tracking. In *IEE Colloquium on Target Tracking: Algorithms and Applications*, pages 16/1–16/4, 1999.
- D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, August 2008.
- M. Ulmke, O. Erdinc, and P. Willett. Gaussian Mixture Cardinalized PHD Filter for Ground Moving Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 1–8, Quebec City, Canada, July 2007.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- B.-T. Vo, B.-N. Vo, and A. Cantoni. Bayesian filtering with random finite set observations. *IEEE Transactions on Signal Processing*, 56(4):1313–1326, April 2008.
- W. Wieneke and J. W. Koch. Probabilistic tracking of multiple extended targets using random matrices. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, Orlando, FL, USA, April 2010.
- H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 73–78, Chicago, IL, USA, July 2011.

Paper E

Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking

Authors: Karl Granström, and Umut Orguner

Edited version of the paper:

K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012c.

Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking

Karl Granström*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
karl@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

In Gilholm *et al.*'s extended target model, the number of measurements generated by a target is Poisson distributed with measurement rate γ . Practical use of this extended target model in multiple extended target tracking algorithms requires a good estimate of γ . In this paper, we first give a Bayesian recursion for estimating γ using the well-known conjugate prior Gamma-distribution. In multiple extended target tracking, consideration of different measurement set associations to a single target makes Gamma-mixtures arise naturally. This causes a need for mixture reduction, and we consider the reduction of Gamma-mixtures by means of merging. Analytical minimization of the Kullback-Leibler divergence is used to compute the single Gamma distribution that best approximates a weighted sum of Gamma distributions. Results from simulations show the merits of the presented multiple target measurement-rate estimator. The Bayesian recursion and presented reduction algorithm have important implications for multiple extended target tracking, e.g. using the implementations of the extended target PHD filter.

1 Introduction

In target tracking, the assumption is often made that a target gives rise to at most one measurement per time step, see e.g. (Bar-Shalom and Fortmann, 1987). However, in extended target tracking this assumption is relaxed, and the extended targets are modeled as potentially giving rise to more than one measurement per time step. In an extended target tracking scenario it is therefore of interest to model the number of measurements that each target gives rise to.

One such model is given by Gilholm et al. (2005), where the measurements are

modeled as an inhomogeneous Poisson point process. At each time step, a Poisson distributed random number of measurements are generated, distributed around the target. Mahler has given an extended target Probability Hypothesis Density (PHD) filter under this model (Mahler, 2009), and a Gaussian Mixture implementation of this filter, called the Extended Target Gaussian Mixture PHD (ETGMPHD) filter, has been presented (Granström et al., 2010; Granström et al., 2012). A Gaussian inverse Wishart implementation of (Mahler, 2009), called the Gaussian inverse Wishart PHD (GIWPHD) filter is presented in (Granström and Orguner, 2012a). The measurement model (Gilholm et al., 2005) can be understood to imply that the extended target is sufficiently far away from the sensor for the measurements to resemble a cluster of points, rather than a geometrically structured ensemble, see e.g. (Granström et al., 2010; Granström et al., 2012) for simulation examples. However, the ETGMPHD filter and GIWPHD filter have also been applied successfully to data from laser range sensors, which give (highly) structured measurements, see (Granström et al., 2012; Granström et al., 2011; Granström and Orguner, 2012a).

In the extended target PHD filter the Poisson rate γ is modeled as a function of the extended target state \mathbf{x} , i.e. $\gamma \triangleq \gamma(\mathbf{x})$. In the ETGMPHD filter γ is approximated as a function of the extended targets' state estimates (Granström et al., 2012). It has been noted that having a good estimate $\hat{\gamma}$ of the true parameter is important when multiple targets are spatially close (Granström et al., 2012). More specifically, under the assumption that the true rate is constant and equal for all targets, the true parameter must lie in the interval

$$\hat{\gamma} - \sqrt{\hat{\gamma}} \leq \gamma \leq \hat{\gamma} + \sqrt{\hat{\gamma}} \quad (1)$$

for the estimated cardinality to be correct (Granström et al., 2012). However, in the most general case the rates are neither constant over time, nor equal for all extended targets. It might also be the case that the true function $\gamma(\cdot)$ is difficult to model, or even time-varying. All of these issues raise the need for a method to estimate individual Poisson rates for multiple extended targets.

In this paper we consider multiple extended targets under the measurement model (Gilholm et al., 2005). The set of extended targets at time t_k is denoted

$$\mathbf{X}_k = \left\{ \mathbf{x}_k^{(i)} \right\}_{i=1}^{N_{x,k}}. \quad (2)$$

At each time step, the number of measurements generated by the i :th target is a Poisson distributed random variable with rate $\gamma_k^{(i)}$. The measurement set at time t_k , denoted

$$\mathbf{Z}_k = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}}, \quad (3)$$

is the union of all target measurements and the clutter measurements. The number of clutter measurements generated at each time step is assumed to be Poisson distributed with rate λ_k . Let \mathbf{Z}^k denote all measurement sets up to, and including, time t_k . We assume the existence of an underlying multiple extended target

tracker that estimates the target states $\mathbf{x}_k^{(i)}$, e.g. (Granström et al., 2010; Granström et al., 2012). The set of measurements that are used to update the state of the i :th target at time t_k is denoted $\mathbf{Z}_k^{(i)}$.

The first objective of this work is to estimate the set of measurement rates $\gamma_k^{(i)}$, given sequences of measurement sets

$$\mathbf{Z}^{k,(i)} \triangleq \left\{ \mathbf{Z}_0^{(i)}, \dots, \mathbf{Z}_k^{(i)} \right\}, \quad i = 1, \dots, n_{x,k}. \quad (4)$$

To this end, in Section 2 we give a recursive Bayesian estimator for $\gamma_k^{(i)}$, with exponential forgetting for the prediction step. We also show how the predicted likelihood is affected when the measurement rates $\gamma_k^{(i)}$ are estimated in addition to estimating the target states $\mathbf{x}_k^{(i)}$.

In the multiple target case under clutter and missed detections, there might be multiple alternative measurement sets (corresponding to different association hypotheses)

$$\mathbf{Z}_k^{(i_1)}, \mathbf{Z}_k^{(i_2)}, \dots, \mathbf{Z}_k^{(i_{N_i})} \quad (5)$$

that are used to update the i :th target state at time t_k . In this case, the state densities of the targets are represented by mixture densities. As time progresses, the number of mixture components grow. To obtain computationally tractable algorithms, hypothesis reduction must be performed, e.g. via pruning or merging.

The second objective of this work is to show how a mixture of γ estimates can be reduced. In Section 3, we consider merging a weighted sum of measurement rate estimates by minimization of the Kullback-Leibler divergence, and we also give a criterion that is used to determine whether or not two components should be merged.

The proposed Bayesian estimator and merging method is evaluated in Simulations in Section 4, and the paper is finalized with concluding remarks in Section 5.

2 Bayesian recursion for γ_k

In this section, we consider recursive estimation of the i :th target's measurement rate $\gamma_k^{(i)}$ from the sequence of measurement sets $\mathbf{Z}^{k,(i)}$. We also show how estimating the measurement rate affects the resulting extended target predicted likelihood. Since we consider only the i :th target, from this point on in this section, we suppress the superscript (i).

2.1 Measurement update and prediction

The conjugate prior to the Poisson distribution is well known to be the Gamma distribution, see e.g. (Gelman et al., 2004). Assume that at time t_k the prior dis-

tribution for the Poisson rate γ_k is a Gamma distribution,

$$p(\gamma_k | \mathbf{Z}^{k-1}) = \mathcal{GAM}(\gamma_k; \alpha_{k|k-1}, \beta_{k|k-1}) \tag{6a}$$

$$= \frac{\beta_{k|k-1}^{\alpha_{k|k-1}}}{\Gamma(\alpha_{k|k-1})} \gamma_k^{\alpha_{k|k-1}-1} e^{-\beta_{k|k-1} \gamma_k}. \tag{6b}$$

Let the k :th measurement set \mathbf{Z}_k contains $N_{z,k}$ elements, where $N_{z,k}$ is Poisson distributed with rate γ_k ,

$$p(N_{z,k} | \gamma_k) = \mathcal{PS}(N_{z,k}; \gamma_k) \tag{7a}$$

$$= \frac{\gamma_k^{N_{z,k}} e^{-\gamma_k}}{N_{z,k}!}. \tag{7b}$$

The posterior distribution is

$$p(\gamma_k | \mathbf{Z}^k) = \mathcal{GAM}(\gamma_k; \alpha_{k|k-1}, \beta_{k|k-1}) \mathcal{PS}(N_{z,k}; \gamma_k) \tag{8a}$$

$$= \frac{\beta_{k|k-1}^{\alpha_{k|k-1}} \gamma_k^{\alpha_{k|k-1} + N_{z,k} - 1} e^{-(\beta_{k|k-1} + 1) \gamma_k}}{\Gamma(\alpha_{k|k-1}) N_{z,k}!} \tag{8b}$$

$$= \mathcal{GAM}(\gamma_k; \alpha_{k|k-1} + N_{z,k}, \beta_{k|k-1} + 1) \tag{8c}$$

$$\times \frac{\Gamma(\alpha_{k|k-1} + N_{z,k}) \beta_{k|k-1}^{\alpha_{k|k-1}}}{\Gamma(\alpha_{k|k-1}) (\beta_{k|k-1} + 1)^{\alpha_{k|k-1} + N_{z,k}} N_{z,k}!}$$

$$= \mathcal{GAM}(\gamma_k; \alpha_{k|k}, \beta_{k|k}) \times \mathcal{L}_\gamma(\alpha_{k|k-1}, \beta_{k|k-1}, N_{z,k}), \tag{8d}$$

where the predicted likelihood $\mathcal{L}_\gamma(\cdot)$ is a negative binomial distribution, see e.g. (Gelman et al., 2004).

In case the true parameter is known to be constant over time, the posterior distribution can be predicted as $p(\gamma_k | \mathbf{Z}^k) = p(\gamma_k | \mathbf{Z}^{k-1})$. However, in the general case γ_k may change over time. We propose to use exponential forgetting with a forgetting factor $\frac{1}{\eta_k}$ for the prediction of γ_k ,

$$\alpha_{k+1|k} = \frac{\alpha_{k|k}}{\eta_k}, \quad \beta_{k+1|k} = \frac{\beta_{k|k}}{\eta_k}, \tag{9}$$

where $\eta_k > 1$. This prediction has an effective window of length $w_e = \frac{1}{1-1/\eta_k} = \frac{\eta_k}{\eta_k-1}$. Using exponential forgetting prediction with window length w_e approximately means that we only “trust” the information that was contained in the measurements from the last w_e time steps.

The expected value and variance of γ_k are

$$E[\gamma_k] = \frac{\alpha_{k|k}}{\beta_{k|k}}, \quad \text{Var}(\gamma_k) = \frac{\alpha_{k|k}}{\beta_{k|k}^2}. \tag{10}$$

Note that the prediction (9) corresponds to keeping the expected value constant while increasing the variance with a factor $\eta_k > 1$.

2.2 Extended target predicted likelihood

The measurement update and corresponding predicted likelihood is an important part of any framework for multiple target tracking under uncertain association and clutter. Let ξ_k denote the augmented extended target state,

$$\xi_k = (\gamma_k, \mathbf{x}_k). \quad (11)$$

Given a set of measurements \mathbf{Z}_k and a prior distribution $p(\xi_k | \mathbf{Z}^{k-1})$, the posterior distribution is

$$p(\xi_k | \mathbf{Z}^k) = p(\mathbf{Z}_k | \xi_k) p(\xi_k | \mathbf{Z}^{k-1}) \quad (12a)$$

$$= p(\mathbf{Z}_k | \xi_k) p(\gamma_k | \mathbf{Z}^{k-1}) p(\mathbf{x}_k | \mathbf{Z}^{k-1}). \quad (12b)$$

Note that there is an implicit assumption here that the prior distribution can be factorized as

$$p(\xi_k | \mathbf{Z}^{k-1}) = p(\gamma_k | \mathbf{Z}^{k-1}) p(\mathbf{x}_k | \mathbf{Z}^{k-1}). \quad (13)$$

This assumption neglects the dependence between the number of measurements and any extension parameters that are included in \mathbf{x}_k . However the probability density over the number of measurements, conditioned on the target extension, is unknown in most applications, and we believe that this assumption is valid in most cases.

Assume also that the measurement likelihood can be decomposed as

$$p(\mathbf{Z}_k | \xi_k) = p(\mathbf{Z}_k, N_{z,k} | \xi_k) = p(N_{z,k} | \gamma_k) p(\mathbf{Z}_k | \mathbf{x}_k). \quad (14)$$

The validity of this assumption is also dependent on the considerations mentioned above. The posterior distribution and predicted likelihood is

$$\underbrace{p(\gamma_k | \mathbf{Z}^k) p(\mathbf{x}_k | \mathbf{Z}^k)}_{\text{posterior}} \times \underbrace{\mathcal{L}_\gamma(\alpha_{k|k-1}, \beta_{k|k-1}, N_{z,k}) \mathcal{L}_\mathbf{x}(\check{\mathbf{x}}_{k|k-1}, \mathbf{Z}_k)}_{\text{predicted likelihood}}, \quad (15)$$

where $\check{\mathbf{x}}_{k|k-1}$ denotes the sufficient statistics of \mathbf{x}_k . Thus, any extended target tracking framework that estimates the states \mathbf{x}_k can be augmented to also include estimates of the measurement rates γ_k .

In the results section below we give an example where we integrate γ_k estimation into the ETGMPPH filter (Granström et al., 2010; Granström et al., 2012). The posterior distribution for γ_k and the corresponding predicted likelihood $\mathcal{L}_\gamma(\cdot)$ are given in (8d). The details for the posterior distribution and predicted likelihood for \mathbf{x}_k , as well as the full filter recursion, can be found in (Granström et al., 2010; Granström et al., 2012).

3 Multi-target mixture reduction

A straightforward way to model uncertainty in multiple target tracking is to use mixtures of distributions, see e.g. the Multi-hypothesis Tracking filter (Bar-Shalom and Rong Li, 1995), or the Gaussian Mixture PHD-filters (Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012). Let $p(\cdot)$ be a mixture of distributions,

$$p(\xi_k) = \sum_{j=1}^{J_{k|k}} w_j p_j(\xi_k) \quad (16)$$

$$= \sum_{j=1}^{J_{k|k}} w_j \mathcal{GAM}\left(\gamma_k; \alpha_{k|k}^{(j)}, \beta_{k|k}^{(j)}\right) p\left(\mathbf{x}_k; \check{\mathbf{x}}_{k|k}^{(j)}\right)$$

where each distribution p_j is called component. A common choice is to model the state \mathbf{x}_k as Gaussian distributed, see e.g. (Bar-Shalom and Rong Li, 1995; Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012), which would give a Gamma Gaussian (GG) distributed extended target ξ_k . In Koch's random matrix framework (Koch, 2008), the extent is modeled as an inverse Wishart distributed random matrix X_k , and the kinematic parameters, i.e. position, velocity and acceleration, are modeled as a random vector \mathbf{x}_k . In this case we have $\xi_k = (\gamma_k, \mathbf{x}_k, X_k)$, and (16) would be a mixture of Gamma Gaussian inverse Wishart distributions.

A natural consequence of the tracking frameworks (Bar-Shalom and Rong Li, 1995; Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012) is the increasing number of mixture components, or hypotheses. To keep the target tracking implementation at a tractable level, the mixture must be reduced regularly, which is typically performed via pruning or merging. The output of mixture reduction is an approximate mixture,

$$\tilde{p}(\xi_k) = \sum_{j=1}^{\tilde{J}_{k|k}} w_j \tilde{p}_j(\xi_k) \quad (17)$$

$$= \sum_{j=1}^{\tilde{J}_{k|k}} w_j \mathcal{GAM}\left(\gamma_k; \tilde{\alpha}_{k|k}^{(j)}, \tilde{\beta}_{k|k}^{(j)}\right) p\left(\mathbf{x}_k; \check{\mathbf{x}}_{k|k}^{(j)}\right),$$

where $\tilde{J}_{k|k} < J_{k|k}$ and the difference between $p(\cdot)$ and $\tilde{p}(\cdot)$ is small by some measure. Here we address mixture reduction via component merging.

One approach to merging is to successively find component pairs that are close by some merging criterion, and merge them, see e.g. (Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012). Different methods for merging of Gaussian mixtures are given in e.g. (Salmond, 1990; Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009; Crouse et al., 2011), a method for merging of Gaussian inverse Wishart mixtures is given in (Granström and Orguner, 2012c). In Section 3.1 we give a theorem which is used to find the

Gamma distribution $q(\cdot)$ that minimizes the Kullback-Leibler divergence between $\bar{w}q(\cdot)$ and the sum $p = \sum_{i \in L} w_i p_i$, where $\bar{w} = \sum_{i \in L} w_i$ and $L \subseteq \{1, \dots, J_k | k\}$.

When the extended targets are modeled with a mixture (16), the merging criterion should consider both parts of the components, i.e. the distributions of both γ_k and \mathbf{x}_k . Different merging criteria for Gaussian distributions are given in e.g. (Salmond, 1990; Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009; Crouse et al., 2011; Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012). In Section 3.2 we give a merging criterion for mixtures of Gamma distributions.

3.1 Merging N Gamma components

The Kullback-Leibler divergence (KL-DIV),

$$\text{KL}(p||q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx, \quad (18)$$

is a measure of how similar two functions p and q are. The KL-DIV is well-known in the literature for its moment-matching characteristics, see e.g. (Bishop, 2006; Minka, 2001), and for probability distributions it is considered the optimal difference measure in a maximum likelihood sense (Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009). Note that minimizing the KL-DIV between p and q w.r.t. q can be rewritten as a maximization problem,

$$\min_q \text{KL}(p||q) = \max_q \int p(x) \log(q(x)) dx. \quad (19)$$

Theorem 1. Let $p(\cdot)$ be a weighted sum of Gamma components,

$$p(\gamma) = \sum_{i=1}^N w_i \mathcal{GAM}(\gamma; \alpha_i, \beta_i) = \sum_{i=1}^N w_i p_i(\gamma), \quad (20)$$

where $\bar{w} = \sum_{i=1}^N w_i$. Let

$$q(\gamma) = \bar{w} \mathcal{GAM}(\gamma; \alpha, \beta) \quad (21)$$

be the minimizer of the KL-DIV between $p(\gamma)$ and $q(\gamma)$ among all Gamma distributions, i.e.

$$q(\gamma) \triangleq \arg \min_{q(\gamma) \in \mathcal{GAM}} \text{KL}(p(\gamma) || q(\gamma)). \quad (22)$$

Then the parameter β is given by

$$\beta = \frac{\alpha}{\frac{1}{\bar{w}} \sum_{i=1}^N w_i \frac{\alpha_i}{\beta_i}}, \quad (23)$$

and the parameter α is the solution to

$$0 = \log \alpha - \psi_0(\alpha) + \frac{1}{\bar{w}} \sum_{i=1}^N w_i (\psi_0(\alpha_i) - \log \beta_i) - \log\left(\frac{1}{\bar{w}} \sum_{i=1}^N w_i \frac{\alpha_i}{\beta_i}\right). \quad (24)$$

Proof: Given in Appendix A.

Remarks: The expression for β (23) corresponds to matching the expected values under both distributions q and p ,

$$\bar{w} E_q [\gamma] = \sum_{i=1}^N w_i E_{p_i} [\gamma]. \quad (25)$$

The expression for α (24) corresponds to matching the expected values of the logarithm under both distributions q and p ,

$$\bar{w} E_q [\log \gamma] = \sum_{i=1}^N w_i E_{p_i} [\log \gamma]. \quad (26)$$

A value for the parameter α is easily obtained by applying a numerical root finding algorithm to (24), e.g. Newton's algorithm, see e.g. (Stoer and Bulirsch, 1993).

3.2 Merging criterion for Gamma components

In this section we derive a criterion that is used to determine whether or not two Gamma components should be merged. When reducing the number of components, it is preferred to keep the overall modality of the mixture. Thus, if the initial mixture $p(\cdot)$ has M modes, then the reduced mixture $\tilde{p}(\cdot)$ should have M modes.

The optimal solution to this problem is to consider every possible way to reduce $J_{k|k}$ components, compute the corresponding KL-div's, and then find the best trade-off between low KL-DIV and reduction of $J_{k|k}$. For $J_{k|k}$ components, there are $B_{J_{k|k}}$ different ways to merge, where B_i is the i :th Bell number (Rota, 1964). Because B_i increases rapidly with i , e.g. $B_5 = 52$ and $B_{10} = 115975$, the optimal solution can not be used in practice.

Instead a merging criterion must be used to determine whether or not a pair of Gamma components should be merged. As merging criterion the KL-DIV could be used, however because it is asymmetrical,

$$\text{KL}(p\|q) \neq \text{KL}(q\|p), \quad (27)$$

it should not be used directly. Instead we use the Kullback-Leibler difference (KL-DIFF), defined for two distributions $p(\gamma)$ and $q(\gamma)$ as

$$D_{\text{KL}}(p(\gamma), q(\gamma)) = \text{KL}(p(\gamma)\|q(\gamma)) + \text{KL}(q(\gamma)\|p(\gamma)) \quad (28a)$$

$$= \int p(\gamma) \log\left(\frac{p(\gamma)}{q(\gamma)}\right) d\gamma + \int q(\gamma) \log\left(\frac{q(\gamma)}{p(\gamma)}\right) d\gamma. \quad (28b)$$

Let $p(\gamma)$ and $q(\gamma)$ be defined as

$$p(\gamma) = \mathcal{GAM}(\gamma; \alpha_1, \beta_1), \quad (29a)$$

$$q(\gamma) = \mathcal{GAM}(\gamma; \alpha_2, \beta_2). \quad (29b)$$

The KL-DIV between $p(\cdot)$ and $q(\cdot)$ is

$$\begin{aligned} & \text{KL}(p(\gamma) \| q(\gamma)) \\ &= \alpha_1 \log \beta_1 - \log \Gamma(\alpha_1) + (\alpha_1 - 1)(\psi_0(\alpha_1) - \log \beta_1) - \alpha_1 \\ & \quad - \alpha_2 \log \beta_2 + \log \Gamma(\alpha_2) - (\alpha_2 - 1)(\psi_0(\alpha_1) - \log \beta_1) + \beta_2 \frac{\alpha_1}{\beta_1} \end{aligned} \quad (30a)$$

$$= \log \left(\frac{\beta_1^{\alpha_1}}{\beta_2^{\alpha_2}} \right) + \log \left(\frac{\Gamma(\alpha_2)}{\Gamma(\alpha_1)} \right) + (\alpha_1 - \alpha_2)(\psi_0(\alpha_1) - \log \beta_1) + \alpha_1 \left(\frac{\beta_2}{\beta_1} - 1 \right), \quad (30b)$$

and the KL-DIV between q and p is defined analogously. The KL-DIFF between p and q becomes

$$\begin{aligned} & D_{\text{KL}}(p(\gamma), q(\gamma)) \\ &= (\alpha_1 - \alpha_2) \left(\psi_0(\alpha_1) - \psi_0(\alpha_2) + \log \frac{\beta_2}{\beta_1} \right) + (\beta_2 - \beta_1) \left(\frac{\alpha_1}{\beta_1} - \frac{\alpha_2}{\beta_2} \right). \end{aligned} \quad (31)$$

3.3 Merging of extended target components

When merging is used to reduce an extended target mixture (16), the merging criterion should be defined over both γ_k and \mathbf{x}_k . For example, the following merging criterion could be used

$$D_{\text{KL}}(p_i(\xi_k), p_j(\xi_k)) < U, \quad (32)$$

where $D_{\text{KL}}(\cdot)$ is the KL-DIFF between two extended target components. Owing to the assumed conditional independence of the distributions over γ_k and \mathbf{x}_k in (12), the KL-DIFF can be expressed as a sum

$$D_{\text{KL}}(p_i(\xi_k), p_j(\xi_k)) = D_{\text{KL}}^\gamma(i, j) + D_{\text{KL}}^{\mathbf{x}}(i, j), \quad (33)$$

where $D_{\text{KL}}^\gamma(i, j) = D_{\text{KL}}(p_i(\gamma), p_j(\gamma))$ is given in (31) and

$$D_{\text{KL}}^{\mathbf{x}}(i, j) = D_{\text{KL}}(p_i(\mathbf{x}), p_j(\mathbf{x})). \quad (34)$$

Thus, the following merging criterion could alternatively be used

$$\left(D_{\text{KL}}^\gamma(i, j) < U_\gamma \right) \& \left(D_{\text{KL}}^{\mathbf{x}}(i, j) < U_{\mathbf{x}} \right), \quad (35)$$

where $\&$ is the logical and operator. In case \mathbf{x}_k is Gaussian distributed, possible merging criteria $D_{\text{KL}}^{\mathbf{x}}(i, j)$ are given in e.g. (Salmond, 1990; Vo and Ma, 2006).

4 Results

In this section results from simulations are presented. The merging criterion is evaluated, and the merging algorithms are compared. Results are also presented for single and multiple target tracking.

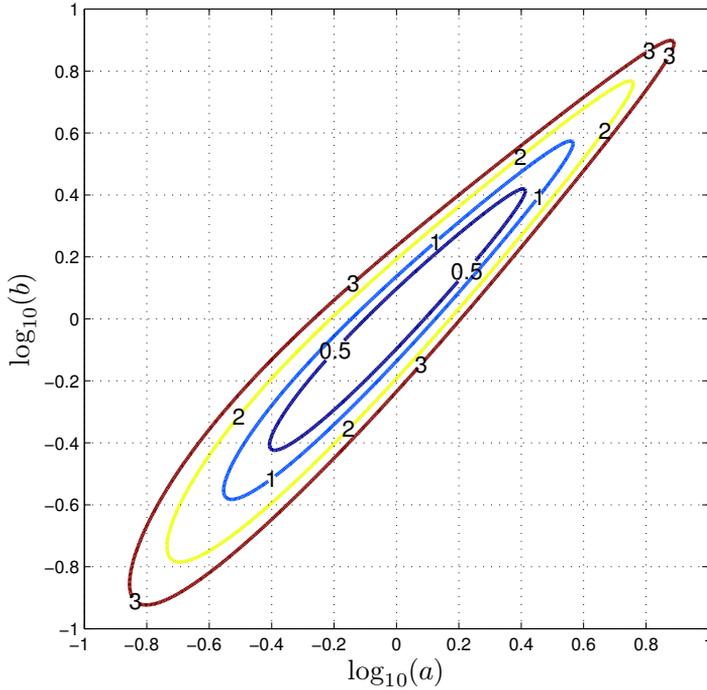


Figure 1: KL-DIFF for two Gamma distributions, when $\alpha_2 = a\alpha_1 = a10$ and $\beta_2 = b\beta_1$. When $a \approx b$, the expected value is approximately the same for both distributions, and the the ratio of the variances is $1/b$. This explains the elongated shape of the KL-DIFF along $a \approx b$.

4.1 Merging criterion

Letting $\alpha_2 = a\alpha_1$ and $\beta_2 = b\beta_1$, the KL-DIFF simplifies to

$$D_{KL}(p(\gamma), q(\gamma)) = \alpha_1(1-a)(\log b + \psi_0(\alpha_1) - \psi_0(a\alpha_1)) + \alpha_1(b-1)\left(1 - \frac{a}{b}\right), \tag{36}$$

i.e. it becomes independent of the specific value of β_1 . It can be shown that, for given a and b , a larger α_1 means a larger KL-DIFF. For $\alpha_1 = 10$, the KL-DIFF is shown in Figure 1.

4.2 Comparison of merging algorithms

An intensity $p(\gamma)$ with 20 Gamma components was reduced using the merging method and criterion presented in Section 3. The Gamma mixture parameters were sampled uniformly from the following intervals,

$$w_i \in [0.05, 0.95], \alpha_i \in [50, 2500], \beta_i \in [5, 50], \tag{37}$$

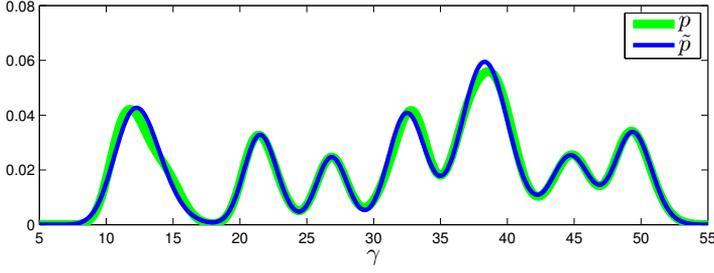


Figure 2: Merging of 20 Gamma components, using the merging method and criterion of Section 3. The reduced mixture has 7 components and has preserved the overall modality.

i.e. α_i and β_i were sampled such that the expected value and variance of γ belongs to $[10, 50]$ and $[1, 2]$, respectively. The original mixture and the approximation are shown in Figure 2. The reduced mixture has 7 components, and manages to capture the overall modality of the original mixture.

4.3 Single target results

A single target with time varying γ was simulated for 150 time steps, the true measurement rate varied with time as shown in Figure 3a. The estimation error, averaged over 10^4 Monte Carlo runs, is shown in Figures 3b, 3c and 3d, for $\eta_k = 1.10$, $\eta_k = 1.25$ and $\eta_k = 2.25$, respectively. With a higher η_k , the estimate responds faster to changes in the true parameter, at the expense of being more sensitive to noise. As with any prediction and correction recursion, setting the parameter requires a trade off between noise cancellation and tracking capabilities.

4.4 Multiple target results

The Bayesian γ_k estimator was integrated into the Gaussian Mixture Probability Hypothesis Density (ETGMPPHD) filter (Granström et al., 2010; Granström et al., 2012). A scenario with three targets was simulated for 100 time steps, the true Poisson rates were set to $\gamma_k^{(1)} = 5$, $\gamma_k^{(2)} = 15$ and $\gamma_k^{(3)} = 30$. Estimation results for $\eta_k = 1.25$ are shown in Figure 4a. The estimates

$$\hat{\gamma}_{k|k}^{(i)} = \frac{\alpha_{k|k}^{(i)}}{\beta_{k|k}^{(i)}} \quad (38)$$

are a bit noisy, however they remain within the bounds given by

$$\gamma_k^{(i)} \pm \sqrt{\gamma_k^{(i)}}, \quad (39)$$

i.e. the true mean \pm one standard deviation. With $\eta_k = 1.01$ the estimation error is much smaller, see Figure 4b. However, as discussed previously, with a low η_k

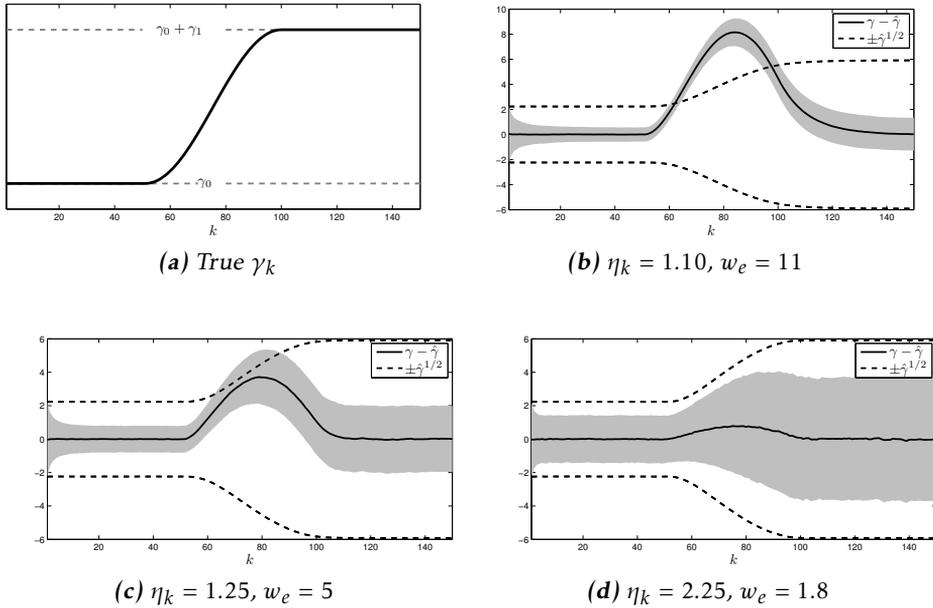


Figure 3: Single target results. (a): The true rate γ_k varied between $\gamma_0 = 5$ and $\gamma_0 + \gamma_1 = 35$. (b), (c) and (d): The solid line shows the average estimation error, the gray area is the average estimation error \pm one standard deviation, and the dashed lines are the bounds $\pm\sqrt{\hat{\gamma}}$, c.f. (1). A higher η_k gives a lower average estimation error, however the estimation error also has much larger standard deviation.

the response to changes in the true parameter would be slower.

5 Concluding Remarks

This paper presented a Bayesian estimator for the rate parameter γ of a Poisson distribution. The conjugate prior of γ is the Gamma distribution, and, using exponential forgetting prediction, it is possible to track a rate γ that changes over time. To manage multiple targets with different rates, a mixture of Gamma distributions is utilized. Mixture reduction is addressed, where components are merged via analytical minimization of the Kullback-Leibler divergence between a weighted sum of Gamma distributions and the single Gamma distribution that best approximates the sum. A simulation study was used to show the merits of the Poisson rate estimation framework.

In future work, we intend to integrate the rate estimation fully into the Gaussian mixture and Gaussian inverse Wishart extended target PHD filters. Having a good estimate of the measurement rate could have important implications for the

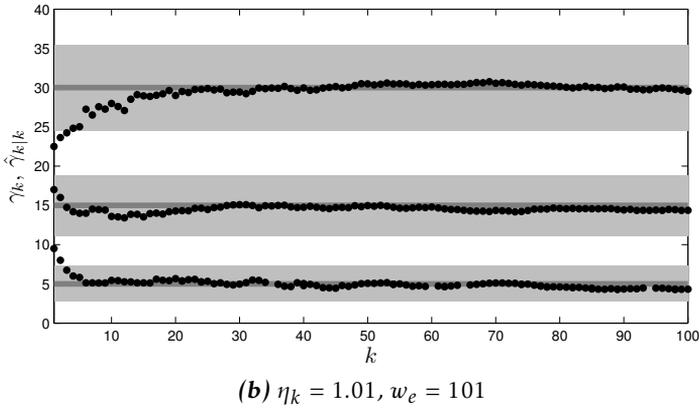
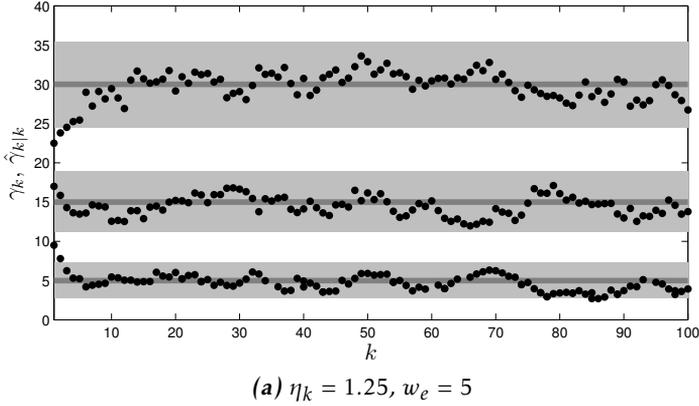


Figure 4: Three targets with true rates $\gamma_k^{(1)} = 5$, $\gamma_k^{(2)} = 15$ and $\gamma_k^{(3)} = 30$, shown as dark gray lines. The estimates $\hat{\gamma}_{k|k}^{(i)}$, shown as black dots, remain within the bounds $\gamma_k^{(i)} \pm \sqrt{\gamma_k^{(i)}}$, i.e. the true mean \pm one standard deviation, shown as light gray areas. The same sequence of measurement sets is used in (a) and (b), with different η_k .

performance, especially during the measurement partitioning step. Future work also includes improving upon the exponential forgetting prediction. The number of measurements generated can be affected by the extended target's position, as well as its shape and size. Including the estimated position, size and shape in the prediction step could possibly improve tracking of Poisson rates that change over time.

A Appendix

First we derive an expected value which is needed in the proof of Theorem 1.

A.1 Expected value of logarithm

Let y be a uni-variate random variable. The moment generating function for y is defined as

$$\mu_y(s) \triangleq E_y [e^{sy}], \tag{40}$$

and the expected value of y is given in terms of $\mu_y(s)$ as

$$E[y] = \left. \frac{d\mu_y(s)}{ds} \right|_{s=0}. \tag{41}$$

Let $y = \log \gamma$, where $p(\gamma) = \mathcal{GAM}(\gamma; \alpha, \beta)$. The moment generating function of y is

$$\mu_y(s) = E[\gamma^s] \tag{42a}$$

$$= \int \gamma^s \frac{\beta^\alpha}{\Gamma(\alpha)} \gamma^{\alpha-1} e^{-\beta\gamma} d\gamma \tag{42b}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(s+\alpha)}{\beta^{s+\alpha}} \int \mathcal{GAM}(\gamma; s+\alpha, \beta) d\gamma \tag{42c}$$

$$= \frac{\Gamma(s+\alpha)}{\Gamma(\alpha) \beta^s}. \tag{42d}$$

The expected value of y is

$$E[y] = E[\log \gamma] \tag{43a}$$

$$= \left. \frac{d}{ds} \left(\frac{\Gamma(s+\alpha)}{\Gamma(\alpha) \beta^s} \right) \right|_{s=0} \tag{43b}$$

$$= \left(\left. \frac{d}{ds} \Gamma(s+\alpha) \right|_{s=0} \right) \frac{1}{\Gamma(\alpha) \beta^s} + \left(\frac{\Gamma(s+\alpha)}{\Gamma(\alpha)} \frac{d}{ds} \beta^{-s} \right) \Big|_{s=0} \tag{43c}$$

$$= \psi_0(\alpha) - \log \beta \tag{43d}$$

where $\psi_0(\cdot)$ is the digamma function (a.k.a. the polygamma function of order 0).

A.2 Proof of Theorem 1

Proof: We have $q(\cdot)$ given as

$$q(\gamma) \triangleq \arg \min_q \text{KL}(p||q) \tag{44a}$$

$$= \arg \max_q \int p(\gamma) \log(q(\gamma)) d\gamma \tag{44b}$$

$$= \arg \max_q \sum_{i=1}^N w_i \int p_i(\gamma) \log(q(\gamma)) d\gamma, \tag{44c}$$

where the i :th integral is

$$\int p_i(\gamma) \log(q(\gamma)) d\gamma$$

$$\begin{aligned}
&= \int p_i(\gamma) [\alpha \log \beta - \log \Gamma(\alpha) + (\alpha + 1) \log \gamma - \beta \gamma] d\gamma \\
&= \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) E_i[\log \gamma] - \beta E_i[\gamma] \\
&= \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) (\psi_0(\alpha_i) - \log \beta_i) - \beta \frac{\alpha_i}{\beta_i}. \tag{45}
\end{aligned}$$

Taking the derivative of the objective function with respect to β , equating the result to zero, and solving for β , we get

$$\beta = \frac{\alpha}{\frac{1}{\bar{w}} \sum_{i=1}^N w_i \frac{\alpha_i}{\beta_i}}. \tag{46}$$

Now, we take the derivative of the objective function with respect to α and equate the result to zero to obtain

$$0 = \sum_{i=1}^N w_i (\log \beta - \psi_0(\alpha) + \psi_0(\alpha_i) - \log \beta_i) \tag{47}$$

$$= \bar{w} \log \beta - \bar{w} \psi_0(\alpha) + \sum_{i=1}^N w_i (\psi_0(\alpha_i) - \log \beta_i). \tag{48}$$

Inserting β and rearranging the terms we obtain

$$0 = \log \alpha - \psi_0(\alpha) + \frac{1}{\bar{w}} \sum_{i=1}^N w_i (\psi_0(\alpha_i) - \log \beta_i) - \log \left(\frac{1}{\bar{w}} \sum_{i=1}^N w_i \frac{\alpha_i}{\beta_i} \right). \tag{49}$$

□

Bibliography

- Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- Y. Bar-Shalom and X. Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson. A Look At Gaussian Mixture Reduction Algorithms. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Chicago, IL, USA, July 2011.
- A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2004.
- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.
- K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012b.
- K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012c.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- J. W. Koch. Bayesian approach to extended object and cluster tracking using

- random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, January 2001.
- G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, May 1964.
- A. R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, July 2007.
- D. J. Salmond. Mixture reduction algorithms for target tracking in clutter. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, pages 434–445, Orlando, FL, USA, January 1990.
- D. Schieferdecker and M. F. Huber. Gaussian Mixture Reduction via Clustering. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Seattle, WA, USA, July 2009.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, second edition, 1993.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- J. L. Williams and P. S. Maybeck. Cost-Function-Based Gaussian Mixture Reduction for Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cairns, Queensland, Australia, July 2003.

Paper F

On the Reduction of Gaussian inverse Wishart Mixtures

Authors: Karl Granström, and Umut Orguner

Edited version of the paper:

K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012d.

On the Reduction of Gaussian inverse Wishart Mixtures

Karl Granström*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
karl@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

This paper presents an algorithm for reduction of Gaussian inverse Wishart mixtures. Sums of an arbitrary number of mixture components are approximated with single components by analytically minimizing the Kullback-Leibler divergence. The Kullback-Leibler *difference* is used as a criterion for deciding whether or not two components should be merged, and a simple reduction algorithm is given. The reduction algorithm is tested in simulation examples in both one and two dimensions. The results presented in the paper are useful in extended target tracking using the random matrix framework.

1 Introduction

In a broad variety of signal processing and sensor fusion problems the state variables are modeled using mixtures. A mixture is a weighted sum of distributions, where the weights are positive. In case the weights sum to one, the mixture is also a distribution. If the weights do not sum to one, the mixture can be called intensity. The individual distributions are called components, a common component choice is the Gaussian distribution, leading to Gaussian mixtures (GM).

In target tracking, GMs are used in e.g. the Multi-hypothesis Tracking (MHT) filter (Bar-Shalom and Rong Li, 1995), and the Gaussian Mixture PHD-filters (Vo and Ma, 2006; Granström et al., 2010; Granström et al., 2012). To keep the complexity at a tractable level, the number of components must be kept at a minimum, leading to the mixture reduction problem. Mixture reduction consists of approximating the original mixture with a reduced mixture, such that the reduced mixture has (considerably) fewer components, while the difference between the two mixtures, defined by some measure, is kept to a minimum.

Several methods for GM reduction have been presented. One solution is pruning, i.e. removing components whose weight is below some threshold (and re-normalizing the weights, if needed). While being very simple, pruning means

that the information contained in the pruned components is completely lost. A possibly better choice is to merge components, because merging, to some extent, attempts to preserve some information from each of the merged components. For GM merging, there are top-down algorithms which successively remove components from the original mixture, and there are bottom-up algorithms which successively add components to the reduced mixture. In the latter case, splitting may be a more appropriate name than merging. In terms of the difference measure applied, there are local algorithms which consider only a subset of the available mixture information, and global algorithms that consider all available mixture information.

Examples of GM reduction algorithms include Salmond (1990) (local, top-down), Williams and Maybeck (2003) (global, top-down), Runnalls (2007) (localized version of global measure, top-down), Huber and Hanebeck (2008) (global, bottom-up), and Schieferdecker and Huber (2009) (global, top-down). A nice overview of the existing literature is given by Crouse et al. (2011). A local top-down approach to reduction of gamma distribution mixtures is presented by Granström and Orguner (2012b).

Gaussian inverse Wishart (GIW) densities have recently been introduced as a representation for extended targets (Koch, 2008). The inverse Wishart distribution is a matrix-variate distribution, which can be used to model the distribution of a Gaussian covariance matrix. For a detailed description of the inverse Wishart distribution, see e.g. (Gupta and Nagar, 2000, Chapter 3). A multiple extended target tracking framework, under association uncertainty and clutter, would inevitably face an increasing number of GIW mixture components. To the best of our knowledge, reduction of mixtures of GIW distributions has not been studied before.

In this paper, GIW mixture reduction via component merging is addressed. The GIW components are merged by analytically minimizing the Kullback-Leibler divergence (KL-DIV) (Kullback and Leibler, 1951) between the components and a single GIW distribution. In the presented top-down merging algorithm, a similarity measure based on the KL-DIV is used, similarly to (Runnalls, 2007). However, here it is considered locally, rather than a local approximation of the global measure as in (Runnalls, 2007). Note that, when it comes to approximating distributions in a maximum likelihood sense, the KL-DIV is considered the optimal difference measure (Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009).

The rest of the paper is organized as follows. Section 2 defines the problem at hand, and the main result of the paper is derived in Section 3. In Section 4 a merging criterion is presented, and the merging algorithm is given in Section 5. Simulation results are presented in Section 6, and concluding remarks are given in Section 7.

2 Problem formulation

The random matrix framework for extended target tracking, introduced by Koch (2008), decomposes the extended target state $\xi = (\mathbf{x}, X)$ into a kinematical state $\mathbf{x} \in \mathbb{R}^{n_x}$ and an extension state $X \in \mathbb{S}_{++}^d$, where \mathbb{R}^{n_x} is the set of real n_x -vectors, \mathbb{S}_{++}^d is the set of symmetric positive definite $d \times d$ matrices, and d is the dimension of the measurements. In (Feldmann and Fränken, 2008; Feldmann et al., 2011) the kinematical and extension state estimate at time step k is modeled as Gaussian inverse Wishart (GIW) distributed,

$$p(\xi_k) = \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \mathcal{IW}(X_k; v_{k|k}, V_{k|k}), \quad (1)$$

where $\mathcal{N}(\cdot)$ denotes a multi-variate Gaussian distribution with mean vector $m \in \mathbb{R}^{n_x}$ and covariance matrix $P \in \mathbb{S}_{++}^{n_x}$ (set of symmetric positive semi-definite $n_x \times n_x$ matrices), and $\mathcal{IW}(\cdot)$ denotes an inverse Wishart distribution with degrees of freedom $v > 2d$ and parameter matrix $V \in \mathbb{S}_{++}^d$. In this work, the inverse Wishart probability density function (pdf) from (Gupta and Nagar, 2000, Definition 3.4.1) is used. The definition is also given in (34b).

In multiple extended target tracking under clutter and association uncertainty, the target intensity can be described using a weighted sum of GIW distributions,

$$\begin{aligned} p(\xi_k) &= \sum_{i=1}^{J_{k|k}} w_i \mathcal{N}(\mathbf{x}_k; m_{k|k}^{(i)}, P_{k|k}^{(i)}) \mathcal{IW}(X_k; v_{k|k}^{(i)}, V_{k|k}^{(i)}) \\ &= \sum_{i=1}^N w_i p_i(\xi_k), \end{aligned} \quad (2)$$

where each distribution $p_i(\cdot)$ is referred to as a GIW component. Note that in some target tracking frameworks the weights do not necessarily sum to unity, and therefore $p(\cdot)$ might not be a probability density. This is the case for the PHD and CPHD filters, see e.g. (Vo and Ma, 2006; Ulmke et al., 2007; Granström et al., 2010; Granström et al., 2012; Orguner et al., 2011; Granström and Orguner, 2012a). As time passes, the number of GIW components grows larger, and approximations become necessary to keep $J_{k|k}$ at a computationally tractable level. One such approximation, called pruning, is to discard components with weights w_i lower than some truncation threshold T . In this work, we explore merging of GIW components, i.e. approximating sums of components with just one component. The result of merging a sum of GIW components (2) is a sum

$$\begin{aligned} \tilde{p}(\xi_k) &= \sum_{i=1}^{\tilde{J}_{k|k}} \tilde{w}_i \mathcal{N}(\mathbf{x}_k; \tilde{m}_{k|k}^{(i)}, \tilde{P}_{k|k}^{(i)}) \mathcal{IW}(X_k; \tilde{v}_{k|k}^{(i)}, \tilde{V}_{k|k}^{(i)}) \\ &= \sum_{i=1}^{\tilde{J}_{k|k}} \tilde{w}_i \tilde{p}_i(\xi_k), \end{aligned} \quad (3)$$

where $\tilde{J}_{k|k} < J_{k|k}$.

Our approach to GIW mixture reduction takes the following steps. First we give a theorem which is used to find the GIW distribution $q(\cdot)$ which minimizes the Kullback-Leibler divergence between $\bar{w}q(\cdot)$ and the sum $p = \sum_{i \in L} w_i p_i$, where $\bar{w} = \sum_{i \in L} w_i$ and $L \subseteq \{1, \dots, J_{k|k}\}$. Next we give a criterion which is used to determine if two GIW components $p_i(\cdot)$ and $p_j(\cdot)$ should be merged or not, and we then give an algorithm which, given a threshold U for the merging criterion, reduces the number of GIW components in the mixture.

3 Approximating a weighted sum of GIW-components with one GIW-component

This section contains the main result of the paper – a theorem that describes how a sum of an arbitrary number of GIW components can be merged into just one GIW component. This is performed via analytical minimization of the KL-DIV,

$$\text{KL}(p||q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx, \quad (4)$$

a measure of how similar two functions p and q are. The KL-DIV is well-known in the literature for its moment-matching characteristics, see e.g. (Bishop, 2006; Minka, 2001), and as mentioned above it is considered the optimal difference measure in a maximum likelihood sense (Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009). Note that minimizing the KL-DIV between p and q w.r.t. q can be rewritten as a maximization problem,

$$\min_q \text{KL}(p||q) = \max_q \int p(x) \log(q(x)) dx. \quad (5)$$

Theorem 1. *Let $p(\cdot)$ be a weighted sum of GIW components,*

$$\begin{aligned} p(\mathbf{x}, X) &= \sum_{i=1}^N w_i \mathcal{N}(\mathbf{x}; m_i, P_i) \mathcal{IW}(X; v_i, V_i) \\ &= \sum_{i=1}^N w_i p_i(\mathbf{x}, X), \end{aligned} \quad (6)$$

where $\bar{w} = \sum_{i=1}^N w_i$. Let

$$q(\mathbf{x}, X) = \bar{w} \mathcal{N}(\mathbf{x}; m, P) \mathcal{IW}(X; v, V) \quad (7)$$

be the minimizer of the KL-DIV between $p(\mathbf{x}, X)$ and $q(\mathbf{x}, X)$ among all GIW distributions, i.e.

$$q(\mathbf{x}, X) \triangleq \arg \min_{q(\mathbf{x}, X) \in \text{GIW}} \text{KL}(p(\mathbf{x}, X) || q(\mathbf{x}, X)). \quad (8)$$

Then the parameters m , P , and V are given by

$$m = \frac{1}{\bar{w}} \sum_{i=1}^N w_i m_i, \tag{9a}$$

$$P = \frac{1}{\bar{w}} \sum_{i=1}^N w_i (P_i + (m_i - m)(m_i - m)^T), \tag{9b}$$

$$V = \bar{w} (v - d - 1) \left(\sum_{i=1}^N w_i (v_i - d - 1) V_i^{-1} \right)^{-1}, \tag{9c}$$

and v is the solution to the equation

$$0 = \bar{w} d \log(v - d - 1) - \bar{w} \sum_{j=1}^d \psi_0 \left(\frac{v - d - j}{2} \right) + \bar{w} d \log \bar{w} \tag{9d}$$

$$- \bar{w} \log \left| \sum_{i=1}^N w_i (v_i - d - 1) V_i^{-1} \right| + \sum_{i=1}^N \sum_{j=1}^d w_i \psi_0 \left(\frac{v_i - d - j}{2} \right) - \sum_{i=1}^N w_i \log |V_i|,$$

where $|V|$ is the determinant of V and $\psi_0(\cdot)$ is the digamma function (a.k.a. the polygamma function of order 0). ■

Proof: Given in Appendix A.

Remarks: The expressions for m in (9a) and P in (9b) are well known, see e.g. the textbook (Gustafsson, 2010), and have been used earlier to merge Gaussian distributions in a target tracking context, see e.g. (Salmond, 1990; Williams and Maybeck, 2003; Vo and Ma, 2006; Runnalls, 2007; Schieferdecker and Huber, 2009; Granström et al., 2010; Granström et al., 2012; Crouse et al., 2011). To the best of the authors' knowledge the identities for the calculation of the parameters V and v have not been published before. The expressions for V and v in (9c) and (9d) correspond to matching the expected values of X^{-1} and $\log |X|$ under both densities,

$$\bar{w} E_q [X^{-1}] = \sum_{i=1}^N w_i E_{p_i} [X^{-1}], \tag{10a}$$

$$\bar{w} E_q [\log |X|] = \sum_{i=1}^N w_i E_{p_i} [\log |X|]. \tag{10b}$$

There is a unique solution to (9d), and a value for the parameter v is easily obtained by applying a numerical root finding algorithm to (9d), e.g. Newton's algorithm, see e.g. (Stoer and Bulirsch, 1993).

4 Merging criterion

In this section we derive a criterion that is used to determine whether or not two GIW components should be merged. When reducing the number of components, it is preferred to preserve the overall modality of the mixture. Thus, if the initial mixture $p(\mathbf{x}, X)$ has M modes, then the reduced mixture $\bar{p}(\mathbf{x}, X)$ should have M modes.

The optimal solution to this problem is to consider every possible way to reduce $J_{k|k}$ components, compute the corresponding KL-div:s, and then find the best trade-off between low KL-DIV and reduction of $J_{k|k}$. For $J_{k|k}$ components, there are $B_{J_{k|k}}$ different ways to merge, where B_i is the i :th Bell number (Rota, 1964). Because B_i increases rapidly with i , e.g. $B_5 = 52$ and $B_{10} = 115975$, the optimal solution can not be used in practice.

Instead a merging criterion must be used to determine whether or not a pair of GIW components should be merged. In what follows we present a distance measure that can be thresholded to compare two GIW components, and we also elaborate on the Gaussian and inverse Wishart parts of this distance measure.

4.1 Distance measure

As distance measure the KL-DIV could be used, however because it is asymmetrical, $\text{KL}(p\|q) \neq \text{KL}(q\|p)$, it should not be used directly. Instead we use the Kullback-Leibler difference (KL-DIFF), defined for two distributions $p(\mathbf{x}, X)$ and $q(\mathbf{x}, X)$ as

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{x}, X), q(\mathbf{x}, X)) \\ &= \text{KL}(p(\mathbf{x}, X) \| q(\mathbf{x}, X)) + \text{KL}(q(\mathbf{x}, X) \| p(\mathbf{x}, X)) \\ &= \iint p(\mathbf{x}, X) \log\left(\frac{p(\mathbf{x}, X)}{q(\mathbf{x}, X)}\right) d\mathbf{x}dX + \iint q(\mathbf{x}, X) \log\left(\frac{q(\mathbf{x}, X)}{p(\mathbf{x}, X)}\right) d\mathbf{x}dX. \end{aligned} \quad (11)$$

Let $p(\mathbf{x}, X)$ and $q(\mathbf{x}, X)$ be defined as

$$p(\mathbf{x}, X) = \mathcal{N}(\mathbf{x}; m_1, P_1) \mathcal{IW}(X; v_1, V_1), \quad (12a)$$

$$q(\mathbf{x}, X) = \mathcal{N}(\mathbf{x}; m_2, P_2) \mathcal{IW}(X; v_2, V_2). \quad (12b)$$

The KL-DIV between $p(\cdot)$ and $q(\cdot)$ is

$$\begin{aligned} &\text{KL}(p(\mathbf{x}, X) \| q(\mathbf{x}, X)) \\ &= \int \mathcal{N}(\mathbf{x}; m_1, P_1) \log\left(\frac{\mathcal{N}(\mathbf{x}; m_1, P_1)}{\mathcal{N}(\mathbf{x}; m_2, P_2)}\right) d\mathbf{x} \\ &\quad + \int \mathcal{IW}(X; v_1, V_1) \log\left(\frac{\mathcal{IW}(X; v_1, V_1)}{\mathcal{IW}(X; v_2, V_2)}\right) dX \\ &= \text{KL}(\mathcal{N}(\mathbf{x}; m_1, P_1) \| \mathcal{N}(\mathbf{x}; m_2, P_2)) \\ &\quad + \text{KL}(\mathcal{IW}(X; v_1, V_1) \| \mathcal{IW}(X; v_2, V_2)), \end{aligned} \quad (13)$$

where

$$\begin{aligned} & \text{KL}(\mathcal{N}(\mathbf{x}; m_1, P_1) \parallel \mathcal{N}(\mathbf{x}; m_2, P_2)) \\ &= \frac{1}{2} \left[\log |P_2| - \log |P_1| - n_x + \text{Tr}(P_2^{-1} P_1) + (m_1 - m_2)^\top P_2^{-1} (m_1 - m_2) \right], \end{aligned} \quad (14)$$

and

$$\begin{aligned} & \text{KL}(\mathcal{IW}(X; v_1, V_1) \parallel \mathcal{IW}(X; v_2, V_2)) \\ &= \frac{v_1 - d - 1}{2} \log |V_1| - \frac{v_2 - d - 1}{2} \log |V_2| \\ & \quad + \sum_{j=1}^d \left(\log \Gamma\left(\frac{v_2 - d - j}{2}\right) - \log \Gamma\left(\frac{v_1 - d - j}{2}\right) \right) \\ & \quad + \frac{v_2 - v_1}{2} \left(\log |V_1| - \sum_{j=1}^d \psi_0\left(\frac{v_1 - d - 1}{2}\right) \right) \\ & \quad + \text{Tr}\left(-\frac{1}{2} (v_1 - d - 1) V_1^{-1} (V_1 - V_2)\right). \end{aligned} \quad (15)$$

Showing (14) and (15) is straightforward, the tedious details are omitted. The KL-DIV between $q(\cdot)$ and $p(\cdot)$ is defined analogously.

Note that the decomposition of $\text{KL}(p(\cdot) \parallel q(\cdot))$ into a sum (13) is inherited from the separability of the Gaussian and inverse Wishart distributions in (12). From (13) it follows that the KL-DIFF is separable,

$$\begin{aligned} D_{\text{KL}}(p(\mathbf{x}, X), q(\mathbf{x}, X)) &= D_{\text{KL}}^{\mathcal{N}} + D_{\text{KL}}^{\mathcal{IW}} \\ &= D_{\text{KL}}(\mathcal{N}(\mathbf{x}; m_1, P_1), \mathcal{N}(\mathbf{x}; m_2, P_2)) + D_{\text{KL}}(\mathcal{IW}(X; v_1, V_1), \mathcal{IW}(X; v_2, V_2)), \end{aligned} \quad (16)$$

where

$$\begin{aligned} & D_{\text{KL}}(\mathcal{N}(\mathbf{x}; m_1, P_1), \mathcal{N}(\mathbf{x}; m_2, P_2)) \\ &= \frac{1}{2} (m_1 - m_2)^\top (P_1^{-1} + P_2^{-1}) (m_1 - m_2) - n_x + \frac{1}{2} \text{Tr}(P_2^{-1} P_1 + P_1^{-1} P_2), \end{aligned} \quad (17)$$

and

$$\begin{aligned} & D_{\text{KL}}(\mathcal{IW}(X; v_1, V_1), \mathcal{IW}(X; v_2, V_2)) \\ &= \frac{1}{2} \text{Tr}\left(\left[(v_1 - d - 1) V_1^{-1} - (v_2 - d - 1) V_2^{-1}\right] (V_2 - V_1)\right) \\ & \quad + \frac{v_2 - v_1}{2} \left(\log |V_1| - \sum_{j=1}^d \psi_0\left(\frac{v_1 - d - j}{2}\right) - \log |V_2| + \sum_{j=1}^d \psi_0\left(\frac{v_2 - d - j}{2}\right) \right). \end{aligned} \quad (18)$$

Note that the Gaussian KL-DIFF (17) has similarities to the merging criterion

$$(m_i - m_j)^\top P_i^{-1} (m_i - m_j), \quad w_i > w_j, \quad (19)$$

which is used to merge sums of Gaussian distributions in e.g. (Salmond, 1990; Vo and Ma, 2006; Granström et al., 2010).

Thresholding the KL-DIFF

$$D_{\text{KL}}(p(\mathbf{x}, X), q(\mathbf{x}, X)) < U \tag{20}$$

is a straightforward way to determine whether or not two Gaussian inverse Wishart distributions should be merged. Alternatively, the Gaussian and inverse Wishart KL-DIFF:s can be thresholded separately,

$$(D_{\text{KL}}^{\mathcal{N}} < U_{\mathcal{N}}) \& (D_{\text{KL}}^{\mathcal{IW}} < U_{\mathcal{IW}}), \tag{21}$$

where & is the logical and operator. In the following two subsections we will elaborate on the Gaussian and inverse Wishart KL-DIFF:s to gain a better understanding of how the merging criterion works.

4.2 A closer look at the Gaussian KL-DIFF

Under the assumption that $P_2 = \alpha P_1$, $\alpha > 0$, and $m_2 = m_1 + P_1^{1/2} m_e$, $P_1^{1/2} P_1^{1/2} = P_1$, the KL-DIFF is independent of the specific values of m_1 and P_1 ,

$$D_{\text{KL}}^{\mathcal{N}} = -n_x + \frac{1 + \frac{1}{\alpha}}{2} m_e^T m_e + \frac{\alpha + \frac{1}{\alpha}}{2} n_x. \tag{22}$$

If $m_e = \mathbf{0}$ the KL-DIFF is $D_{\text{KL}}^{\mathcal{N}} = \frac{1}{2} (\alpha + \frac{1}{\alpha}) n_x$. With a threshold $U_{\mathcal{N}}$, $D_{\text{KL}}^{\mathcal{N}} < U_{\mathcal{N}}$ is equivalent to $\alpha_1 < \alpha < \alpha_2$, where

$$\alpha_i = 1 + \frac{U_{\mathcal{N}}}{n_x} + (-1)^i \sqrt{\left(1 + \frac{U_{\mathcal{N}}}{n_x}\right)^2 - 1}. \tag{23}$$

Thus, the upper and lower limit of α is dependent on both the threshold, and on the dimension of the kinematical state n_x . For a given threshold $U_{\mathcal{N}}$, a larger n_x means that α must be closer to 1 for $D_{\text{KL}}^{\mathcal{N}} < U_{\mathcal{N}}$ to be fulfilled.

If $\alpha = 1$ the KL-DIFF is $D_{\text{KL}}^{\mathcal{N}} = m_e^T m_e$, i.e. the length of m_e squared. For a given threshold $U_{\mathcal{N}}$ the difference between m_1 and m_2 can at most be $\sqrt{U_{\mathcal{N}}}$ standard deviations. Thus, given $\alpha = 1$, the KL-DIFF can be defined in terms of the standard deviation $P_1^{1/2}$, and is independent of the size of the kinematical state \mathbf{x} .

4.3 A closer look at the inverse Wishart KL-DIFF

Under the assumption that $V_2 = \beta V_1$, the KL-DIFF becomes independent of the specific value of V_1 . If $v_2 = v_1$ the KL-DIFF is

$$D_{\text{KL}}^{\mathcal{IW}} = \frac{(v_1 - d - 1)d(\beta - 1)^2}{2\beta}. \tag{24}$$

With a threshold $U_{\mathcal{IW}}$, $D_{\text{KL}}^{\mathcal{IW}} < U_{\mathcal{IW}}$ is equivalent to $\beta_1 < \beta < \beta_2$ where

$$\beta_i = 1 + \frac{U_{\mathcal{IW}}}{(v_1 - d - 1)d} + (-1)^i \sqrt{\left(1 + \frac{U_{\mathcal{IW}}}{(v_1 - d - 1)d}\right)^2 - 1}. \tag{25}$$

The upper and lower limit of β is dependent on the threshold U_{TW} , the dimension of the measurements d , and on the inverse Wishart degrees of freedom v_1 . A higher threshold gives larger β_2 and smaller β_1 , while a higher d and/or v_1 forces both limits closer to one.

Unfortunately there is no obvious way to choose v_2 as a function of v_1 to make the KL-DIFF independent of the specific value of v_1 , making it difficult to make a similar examination of how the inverse Wishart degrees of freedom affect the KL-DIFF.

4.4 Discussion

The subsections above give some intuition as to how U (or U_N and U_{TW}) affects the merging criterion, however it is difficult to give specific hints for choosing a numerical value of U . Such a value is likely best determined empirically. In the results section below we will examine all four GIW parameters, and how they affect the KL-DIFF, in numerical examples.

5 Merging algorithm

In this section we present a merging algorithm that uses the merging method and criterion defined above, see Table 1. In the algorithm a choice is made regarding how aggressively the components are bundled for merging, i.e. how aggressively $J_{k|k}$ is reduced. There are many possible ways to do this, two are given in Table 1. Both alternatives start by picking out the GIW component with highest weight, say the j :th. The first alternative, L_1 in Table 1, then merges component j with all other components i for which it holds

$$D_{\text{KL}}(p_j(\mathbf{x}, X), p_i(\mathbf{x}, X)) < U. \quad (26)$$

The second alternative, L_2 in Table 1, finds all other components such that for each component $i \in L_2$, there exists a sequence of indices $\{i_1 = i, \dots, i_N = j\}$ such that

$$D_{\text{KL}}(p_{i_k}(\mathbf{x}, X), p_{i_{k+1}}(\mathbf{x}, X)) < U, \quad k = 1, \dots, N - 1. \quad (27)$$

L_1 is a special case of L_2 , where $\{i_1 = i, i_2 = j\}$, and it immediately follows that $|L_1| \leq |L_2|$, where $|L|$ is the cardinality of the set L . Thus L_2 merges more components than L_1 , resulting in a higher reduction of $J_{k|k}$, but also a cruder approximation of $p(\mathbf{x}, X)$.

6 Simulation results

This section presents results from numerical simulations. Simulations of the Gaussian and inverse Wishart parts of the KL-DIFF are presented in Section 6.1, and merging of GIW components in $n_x = d = 1$ and $n_x = d = 2$ dimensions are presented in Sections 6.2 and 6.3. In Section 6.4 we compare the two merging

Table 1: Gaussian inverse Wishart Reduction

-
- 1: **require:** $p(\mathbf{x}_k, X_k)$ as in (2), a merging threshold U , and $\theta \in \{1, 2\}$.
- 2: **initialize:** Set $\ell = 0$ and $I = \{1, \dots, J_{k|k}\}$.
- 3: **repeat**
- 4: Set $\ell = \ell + 1$ and $j = \arg \max_{i \in I} w_{k|k}^{(i)}$
- 5: Set $L = L_\theta$, where $L_1 = \{i \in I \mid D_j^i < U\}$,
 $L_2 = \{i \in I \mid \exists \{i_1 = i, \dots, i_N = j\} \ni D_{i_k}^{i_{k+1}} < U, k = 1, \dots, N - 1\}$,
and $D_j^i = D_{\text{KL}}(p_j(\mathbf{x}, X), p_i(\mathbf{x}, X))$.
- 6: Use Theorem 1 to compute
 $\tilde{w}_{k|k}^{(\ell)}, \tilde{m}_{k|k}^{(\ell)}, \tilde{P}_{k|k}^{(\ell)}, \tilde{v}_{k|k}^{(\ell)}, \tilde{V}_{k|k}^{(\ell)}$ (28)
for the components $i \in L$.
- 7: $I = I \setminus L$
- 8: **until** $I = \emptyset$
- 9: **output:** $\tilde{p}(\mathbf{x}_k, X_k) = \sum_{i=1}^{\tilde{J}_{k|k}} \tilde{w}_i \mathcal{N}(\mathbf{x}_k; \tilde{m}_i, \tilde{P}_i) \mathcal{IW}(X_k; \tilde{v}_i, \tilde{V}_i)$, where the number of components is $\tilde{J}_{k|k} = \ell$.
-

choices L_1 and L_2 in $n_x = d = 1$ dimension.

6.1 Merging criterion

This section presents results that evaluate the merging criterion in Section 4. Let $p_1(\mathbf{x}, X)$ and $p_2(\mathbf{x}, X)$ be defined as

$$p_1(\mathbf{x}, X) = \mathcal{N}(\mathbf{x}; m_1, P_1) \mathcal{IW}(X; v_1, V_1), \quad (29a)$$

$$p_2(\mathbf{x}, X) = \mathcal{N}(\mathbf{x}; m_2, P_2) \mathcal{IW}(X; v_2, V_2). \quad (29b)$$

The evaluation is performed *ceteris paribus*, i.e. by changing the parameters of the Gaussian while holding the parameters of the inverse Wishart equal, and vice versa.

Different Gaussian parameters

Let $P_2 = \alpha P_1$, and $m_2 = m_1 + P_1^{1/2} m_e$. A contour plot of the KL-DIFF for two univariate Gaussian distributions ($n_x = 1$) is shown in Figure 1a. In accordance with the discussion in Section 4, the KL-DIFF increases with the length of m_e , and it increases when $\alpha < 1$ or $\alpha > 1$.

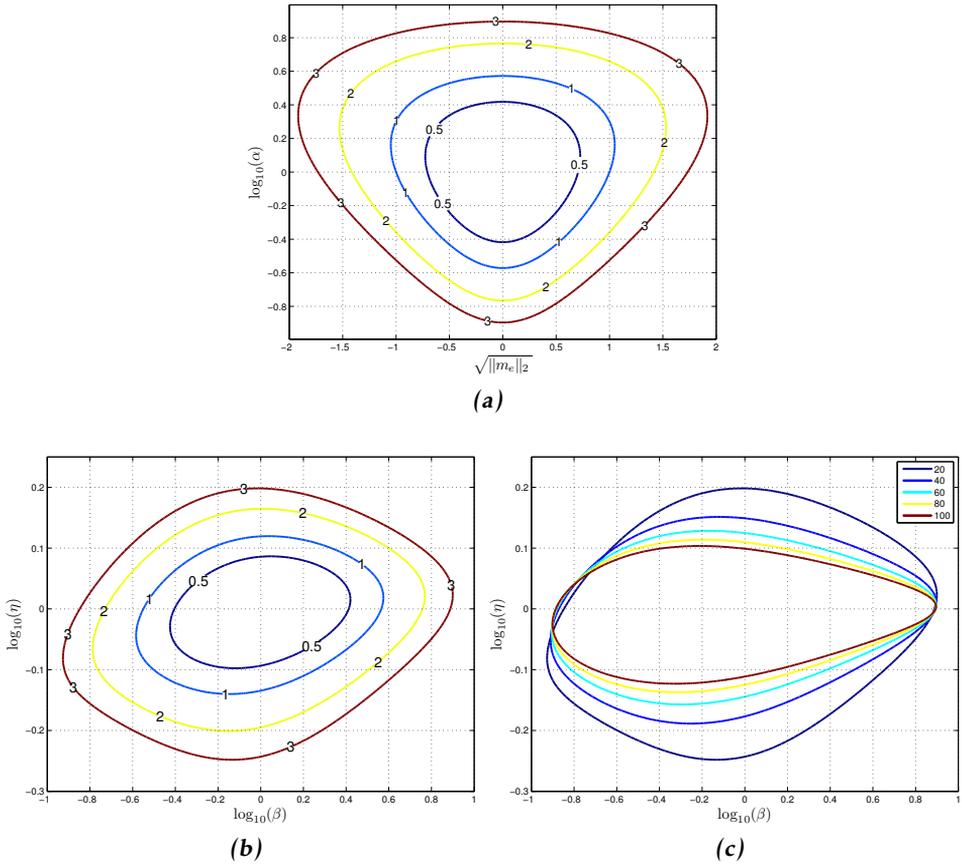


Figure 1: Contour plots showing the KL-DIFF when the Gaussian inverse Wishart parameters are changed. (a) KL-DIFF for two uni-variate Gaussian distributions. (b) KL-DIFF for two one dimensional inverse Wishart distributions, here $v_1 = 20$. (c) KL-DIFF for pairs of one dimensional inverse Wishart distributions, the outlines show $D_{KL} = 3$ and the legend shows the value of v_1 .

Different inverse Wishart parameters

Let $V_2 = \beta V_1$ to make the KL-DIFF independent of the specific value of V_1 . For a given β , setting $v_2 = 2d + 2 + \beta(v_1 - 2d - 2)$ will give correct expected value of X . We make changes to this value by multiplying with a factor η , i.e. $v_2 = \eta(2d + 2 + \beta(v_1 - 2d - 2))$. A contour plot of the KL-DIFF for a one dimensional inverse Wishart distribution is shown in Figure 1b, in this figure $v_1 = 20$. The contours $D_{KL} = 3$ are shown for $v_1 = 20, 40, 60, 80, 100$ in Figure 1c, where it shows how the area enclosed by $D_{KL} = 3$ decreases when v_1 increases.

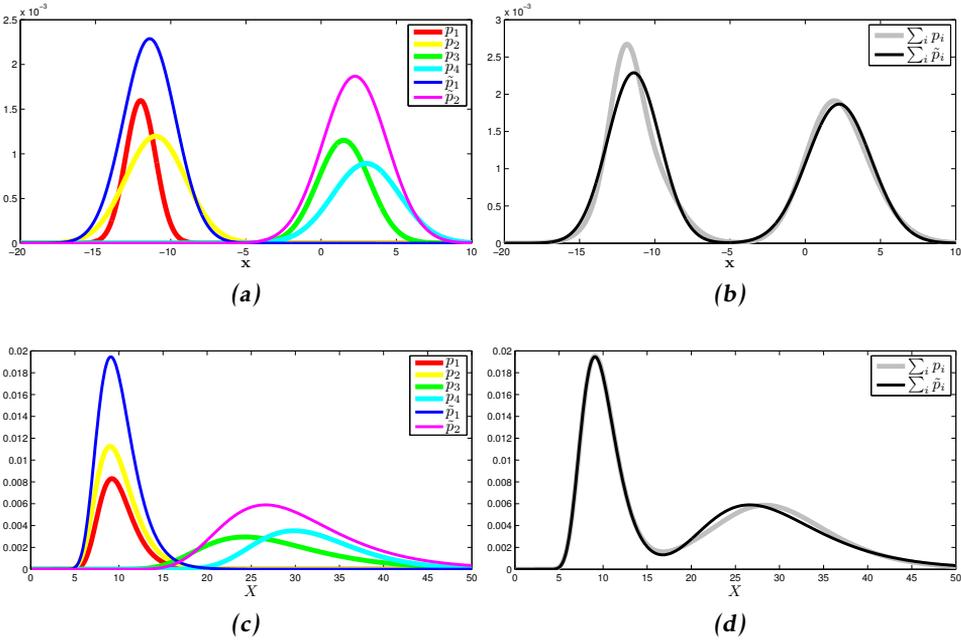


Figure 2: Four GIW components, $n_x = d = 1$, merged into two components using a threshold $U = 3$. (a) shows the Gaussian parts of the components before and after merging, and (b) shows the sums of the Gaussian distributions before and after merging. (c) shows the inverse Wishart parts of the components before and after merging, and (d) shows the sums of the inverse Wishart distributions before and after merging.

6.2 Merging of one dimensional components

An intensity $p(\mathbf{x}, X)$ with four GIW components, $n_x = d = 1$, was reduced to two components using a KL-DIFF threshold of $U = 3$. The GIW components and sums are shown before and after merging in Figure 2.

6.3 Merging of two dimensional components

An intensity $p(\mathbf{x}, X)$ with two GIW components, $n_x = d = 2$, was reduced to one component using a KL-DIFF threshold of $U = 12$. The GIW components are shown before and after merging in Figure 3.

6.4 Comparison of merging algorithms

An intensity $p(\mathbf{x}, X)$ with 50 GIW components, $n_x = d = 1$, was reduced using both L_1 and L_2 in Table 1. The GIW mixture parameters were sampled uniformly from the following intervals,

$$w_i \in [0.05 \ 0.95], m_i \in [0 \ 10], P_i \in [0.25^2 \ 0.75^2], \quad (30a)$$

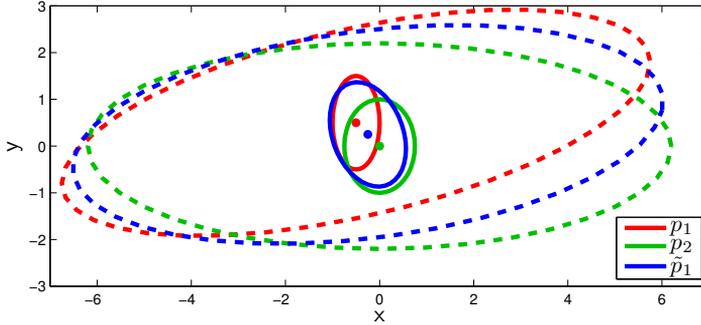


Figure 3: Two GIW components, $n_x = d = 2$, merged into one component using a threshold $U = 12$. Shown are the kinematical state means m (dots), the corresponding covariances P (solid ellipses), and the estimated expected values of the extension state X (dashed ellipses).

$$v_i \in [50 \ 250], \quad \frac{V_i}{v_i - 2d - 2} \in [15 \ 50], \quad (30b)$$

i.e. V_i was sampled such that, given a sampled v_i , the expected value of X belongs to $[15 \ 50]$. The original mixture, and the two approximations, are shown in Figure 4. Using L_1 the reduced mixture has 29 components, using L_2 gives only 23 components, but also a cruder approximation.

7 Concluding remarks

This paper presented a reduction algorithm for mixtures of Gaussian inverse Wishart distributions. A theorem was given, which is used to reduce an arbitrary number of GIW components to just one component by analytically minimizing the Kullback-Leibler divergence, in a maximum likelihood sense the optimal difference measure. Using the Kullback-Leibler difference, a merging criterion for pairs of GIW components was given. The criterion has the benefit of decomposing easily into separate criteria for the Gaussian distributions and inverse Wishart distributions, respectively. A simple algorithm for GIW mixture reduction was also given, and tested in simulation examples in both one and two dimensions.

The outlook on future work includes considering a global difference measure between the original and reduced mixture, instead of just a local measure. The reduction algorithm will be used in the Gaussian inverse Wishart PHD-filter for multiple extended target tracking under association uncertainty and clutter, see (Granström and Orguner, 2012a).

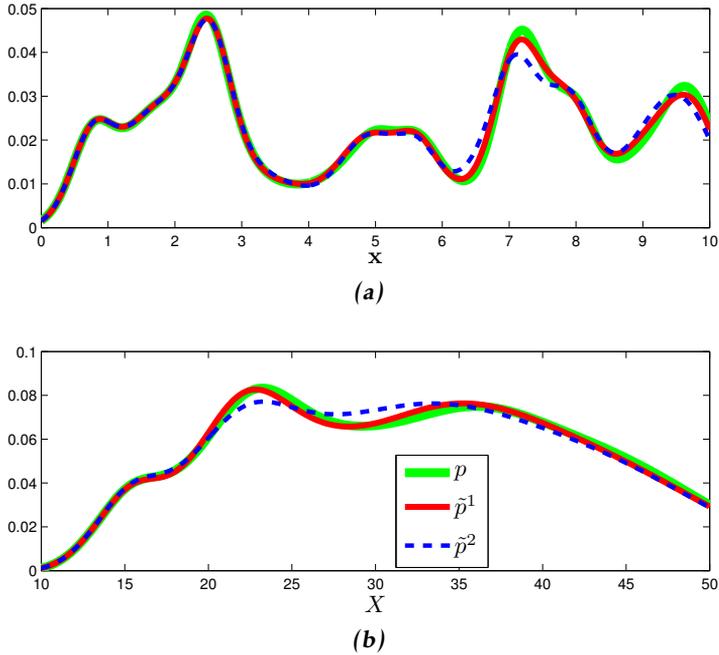


Figure 4: Merging of 50 GIW components. (a) shows the Gaussian sum, before merging (green) as well as after merging using L_1 (red) and L_2 (blue). (b) shows the corresponding inverse Wishart sum. Using L_1 results in 28 GIW components, using L_2 results in 21 components, but also a cruder approximation.

A Appendix

Before giving the proof of Theorem 1, we establish some preliminary results in the form of two expected values.

A.1 Expected value of inverse extension

Let X be inverse Wishart distributed $\mathcal{IW}(X; \nu, V)$. Then X^{-1} is Wishart distributed $\mathcal{W}(X^{-1}; \nu - d - 1, V^{-1})$ (Gupta and Nagar, 2000, Theorem 3.4.1). The expected value of X^{-1} is (Gupta and Nagar, 2000, Theorem 3.3.15)

$$\mathbb{E}[X^{-1}] = (\nu - d - 1) V^{-1}. \quad (31)$$

A.2 Expected value of log determinant of extension

Let y be a uni-variate random variable. The moment generating function for y is defined as

$$\mu_y(s) \triangleq \mathbb{E}_y[e^{sy}], \quad (32)$$

and the expected value of y is given in terms of $\mu_y(s)$ as

$$E[y] = \left. \frac{d\mu_y(s)}{ds} \right|_{s=0}. \tag{33}$$

Let $y = \log|X|$, where $X \sim \mathcal{IW}(X; v, V)$. The moment generating function of y is

$$\mu_y(s) = E[|X|^s] = \int |X|^s p(X) dX \tag{34a}$$

$$= \int |X|^s \frac{2^{-\frac{(v-d-1)d}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |X|^{\frac{v}{2}}} \text{etr}\left(-\frac{1}{2} X^{-1} V\right) dX \tag{34b}$$

$$= \int \frac{2^{-\frac{(v-d-1)d}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |X|^{\frac{v-2s}{2}}} \text{etr}\left(-\frac{1}{2} X^{-1} V\right) dX \tag{34c}$$

$$= \frac{\Gamma_d\left(\frac{v-2s-d-1}{2}\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \int \mathcal{IW}(X; v-2s, V) dX \tag{34d}$$

$$= \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s, \tag{34e}$$

where $\Gamma_d(\cdot)$ is the multivariate gamma function. By (Gupta and Nagar, 2000, Theorem 1.4.1), the logarithm of $\Gamma_d(\cdot)$ can be expressed as

$$\log \Gamma_d(a) = d(d-1) \log \pi + \sum_{i=1}^d \log \Gamma\left(a - \frac{i-1}{2}\right). \tag{35}$$

The expected value of y is

$$E[y] = E[\log|X|] \tag{36a}$$

$$= \left. \frac{d}{ds} \left(\frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \right) \right|_{s=0} \tag{36b}$$

$$= \left(\frac{|V|}{2^d}\right)^s \left. \frac{d}{ds} \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \right|_{s=0} + \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left. \frac{d}{ds} \left(\frac{|V|}{2^d}\right)^s \right|_{s=0} \tag{36c}$$

$$= \left(\frac{|V|}{2^d}\right)^s \left. \frac{d}{ds} \log \Gamma_d\left(\frac{v-d-1}{2} - s\right) \right|_{s=0} + \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \left. \log\left(\frac{|V|}{2^d}\right) \right|_{s=0} \tag{36d}$$

$$= - \sum_{j=1}^d \psi_0\left(\frac{v-d-1}{2} - \frac{j-1}{2}\right) + \log\left(\frac{|V|}{2^d}\right) \tag{36e}$$

$$= \log |V| - d \log 2 - \sum_{j=1}^d \psi_0 \left(\frac{v-d-j}{2} \right). \quad (36f)$$

A.3 Proof of Theorem 1

The density $q(\mathbf{x}, X)$ is

$$\begin{aligned} q(\mathbf{x}, X) &\triangleq \arg \min_{q(\mathbf{x}, X)} \text{KL}(p(\mathbf{x}, X) \| q(\mathbf{x}, X)) \\ &= \arg \max_{q(\mathbf{x}, X)} \sum_{i=1}^N w_i \iint \mathcal{N}(\mathbf{x}; m_i, P_i) \\ &\quad \times \mathcal{IW}(X; v_i, V_i) \log(q(\mathbf{x}, X)) \, d\mathbf{x} dX, \end{aligned} \quad (37)$$

where the i :th double integral over \mathbf{x} and X can be rewritten as

$$\begin{aligned} &\iint \mathcal{N}(\mathbf{x}; m_i, P_i) \mathcal{IW}(X; v_i, V_i) \log(q(\mathbf{x}, X)) \, d\mathbf{x} dX \\ &= \log \bar{w} + \int \mathcal{N}(\mathbf{x}; m_i, P_i) \log \mathcal{N}(\mathbf{x}; m, P) \, d\mathbf{x} \\ &\quad + \int \mathcal{IW}(X; v_i, V_i) \log \mathcal{IW}(X; v, V) \, dX. \end{aligned} \quad (38)$$

The integral over \mathbf{x} simplifies to

$$\int \mathcal{N}(\mathbf{x}; m_i, P_i) \log \mathcal{N}(\mathbf{x}; m, P) \, d\mathbf{x} \quad (39a)$$

$$= \int \mathcal{N}(\mathbf{x}; m_i, P_i) \left[-\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |P| - \frac{1}{2} \text{Tr} \left((\mathbf{x} - m)(\mathbf{x} - m)^\top P^{-1} \right) \right] \, d\mathbf{x} \quad (39b)$$

$$= -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |P| - \frac{1}{2} \text{Tr} \left(\mathbb{E}_{P_i} \left[(\mathbf{x} - m)(\mathbf{x} - m)^\top \right] P^{-1} \right) \quad (39c)$$

$$= -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |P| - \frac{1}{2} \text{Tr} \left((P_i + (m_i - m)(m_i - m)^\top) P^{-1} \right) \quad (39d)$$

$$\triangleq f_i(m, P) \quad (39e)$$

and the integral over X simplifies to

$$\int \mathcal{IW}(X; v_i, V_i) \log \mathcal{IW}(X; v, V) \, dX \quad (40a)$$

$$\begin{aligned} &= \int \mathcal{IW}(X; v_i, V_i) \left[-\frac{(v-d-1)d}{2} \log 2 + \frac{v-d-1}{2} \log |V| \right. \\ &\quad \left. - \log \Gamma_d \left(\frac{v-d-1}{2} \right) - \frac{v}{2} \log |X| + \text{Tr} \left(-\frac{1}{2} X^{-1} V \right) \right] \, dX \end{aligned} \quad (40b)$$

$$= -\frac{(v-d-1)d}{2} \log 2 + \frac{v-d-1}{2} \log |V| - \log \Gamma_d \left(\frac{v-d-1}{2} \right)$$

$$-\frac{v}{2} \mathbb{E}_{p_i} [\log |X|] - \frac{1}{2} \text{Tr} \left(\mathbb{E}_{p_i} [X^{-1}] V \right) \quad (40c)$$

$$= -\frac{(v-d-1)d}{2} \log 2 + \frac{v-d-1}{2} \log |V| - \log \Gamma_d \left(\frac{v-d-1}{2} \right)$$

$$-\frac{v}{2} \left(\log |V_i| - d \log 2 - \sum_{j=1}^d \psi_0 \left(\frac{v_i-d-j}{2} \right) \right) - \frac{1}{2} \text{Tr} \left((v_i-d-1) V_i^{-1} V \right) \quad (40d)$$

$$\triangleq g_i(v, V) \quad (40e)$$

where the expected values are derived above. We thus have

$$q(\mathbf{x}, X) \triangleq \arg \min_{q(\mathbf{x}, X)} \text{KL} (p(\mathbf{x}, X) \| q(\mathbf{x}, X)) \quad (41a)$$

$$= \arg \max_{q(\mathbf{x}, X)} \sum_{i=1}^N w_i (\log \bar{w} + f_i(m, P) + g_i(v, V)) \quad (41b)$$

$$= \arg \max_{q(\mathbf{x}, X)} h(m, P, v, V). \quad (41c)$$

Differentiating the objective function $h(\cdot)$ w.r.t. m , setting equal to zero and solving for m gives

$$m = \frac{1}{\bar{w}} \sum_{i=1}^N w_i m_i. \quad (42)$$

Differentiating the objective function $h(\cdot)$ w.r.t. P , setting equal to zero and solving for P gives

$$P = \frac{1}{\bar{w}} \sum_{i=1}^N w_i \left(P_i + (m_i - m)(m_i - m)^T \right). \quad (43)$$

Differentiating the objective function $h(\cdot)$ w.r.t. V , setting equal to zero and solving for V gives

$$V = \bar{w} (v-d-1) \left(\sum_{i=1}^N w_i (v_i-d-1) V_i^{-1} \right)^{-1}. \quad (44)$$

Differentiating the objective function $h(\cdot)$ w.r.t. v , inserting V (44), and setting equal to zero gives

$$0 = \bar{w} d \log (v-d-1) - \bar{w} \sum_{j=1}^d \psi_0 \left(\frac{v-d-j}{2} \right) + \bar{w} d \log \bar{w} \quad (45)$$

$$- \bar{w} \log \left| \sum_{i=1}^N w_i (v_i-d-1) V_i^{-1} \right| + \sum_{i=1}^N \sum_{j=1}^d w_i \psi_0 \left(\frac{v_i-d-j}{2} \right) - \sum_{i=1}^N w_i \log |V_i|.$$

■

Bibliography

- Y. Bar-Shalom and X. Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson. A Look At Gaussian Mixture Reduction Algorithms. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Chicago, IL, USA, July 2011.
- M. Feldmann and D. Fränken. Tracking of Extended Objects and Group Targets using Random Matrices - A New Approach. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cologne, Germany, July 2008.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.
- K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012b.
- K. Granström and U. Orguner. On the Reduction of Gaussian inverse Wishart mixtures. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2162–2169, Singapore, July 2012c.
- K. Granström, C. Lundquist, and U. Orguner. A Gaussian Mixture PHD filter for Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- K. Granström, C. Lundquist, and U. Orguner. Extended Target Tracking using a Gaussian Mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics. Chapman & Hall, 2000. ISBN 9781584880462.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2010.
- M.F. Huber and U.D. Hanebeck. Progressive gaussian mixture reduction. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cologne, Germany, July 2008.

- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, January 2001.
- U. Orguner, C. Lundquist, and K. Granström. Extended Target Tracking with a Cardinalized Probability Hypothesis Density Filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 65–72, Chicago, IL, USA, July 2011.
- G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, May 1964.
- A. R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, July 2007.
- D. J. Salmond. Mixture reduction algorithms for target tracking in clutter. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, pages 434–445, Orlando, FL, USA, January 1990.
- D. Schieferdecker and M. F. Huber. Gaussian Mixture Reduction via Clustering. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Seattle, WA, USA, July 2009.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, second edition, 1993.
- M. Ulmke, O. Erdinc, and P. Willett. Gaussian Mixture Cardinalized PHD Filter for Ground Moving Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 1–8, Quebec City, Canada, July 2007.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- J. L. Williams and P. S. Maybeck. Cost-Function-Based Gaussian Mixture Reduction for Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cairns, Queensland, Australia, July 2003.

Paper G

A New Prediction for Extended Targets with Random Matrices

Authors: Karl Granström, and Umut Orguner

Edited version of the paper:

K. Granström and U. Orguner. A New Prediction Update for Extended Target Tracking with Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 2012b.

A New Prediction for Extended Targets with Random Matrices

Karl Granström*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
karl@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

This paper presents a new prediction update for extended targets whose extensions are modeled as random matrices. The prediction is based on several minimizations of the Kullback-Leibler divergence and allows for a kinematic state dependent transformation of the target extension. The results show that the extension prediction is a significant improvement over the previous work carried out on the topic.

1 Introduction

Extended targets are targets that potentially give rise to more than one measurement per time step, in contrast to standard targets that give rise to at most one measurement per time step, see e.g. (Bar-Shalom and Fortmann, 1987). The multiple measurements per time step raise interesting possibilities to estimate the target's extension, i.e. the shape and size. Several extended target models have been proposed in the literature, see e.g. (Salmond and Parr, 2003; Baum and Hanebeck, 2009; Granström et al., 2011; Lundquist et al., 2011; Zhu et al., 2011; Koch, 2008; Feldmann et al., 2011) and the references therein.

In the extended target model proposed by Koch (2008); Feldmann et al. (2011), the target extension is modeled as an ellipse, and it is represented by a positive definite matrix called extension matrix. The extended target originated measurements are modeled as being (approximately) Gaussian distributed, with covariance related to the extension matrix. Following a Bayesian methodology, the extension matrix is modeled to be a random variable. Hence we refer to the model by Koch (2008); Feldmann et al. (2011) as the the random matrix framework. that is inverse Wishart distributed. The overall extended target state is defined as the combination of the extension matrix and the usual kinematical state vector. The parameters of the kinematical state density, and the extension's inverse Wishart density, are updated in a Bayesian recursion, which consists of prediction (time update) and correction (measurement update).

The focus in this paper is on the prediction update of extended targets within the random matrix framework. In early work, see (Koch, 2008; Feldmann et al., 2011), the extension matrix' prediction was based on simple heuristics which increase the extension's covariance, while keeping the expected value constant. Koch also discusses the use of a Wishart transition density for the extension state (Koch, 2008), see also (Lian et al., 2010; Lan and Rong Li, 2012). In this paper we generalize this idea by including the possibility of a kinematic state dependent transformation of the extension. This would, for example, be useful when the target extension rotates during a coordinated-turn, a situation which appears very frequently in air traffic control applications. In order to derive an Bayesian prediction update for the extension, minimizations of the Kullback-Leibler divergence are used to approximate densities. This methodology enables us to make well-defined approximations when the original density and its approximation have different numbers of parameters.

The rest of the paper is organized as follows. In Section 2 we give a brief introduction to the random matrix framework, and present the approaches to prediction given in (Koch, 2008; Feldmann et al., 2011; Lian et al., 2010; Lan and Rong Li, 2012). Section 3 presents a problem formulation and defines the main aim of the study. In Section 4, we give results that are used in the derivation of the main result, which is a new prediction update presented in Section 5. The merits of the new update are illustrated in simulations, with comparisons to previous methods in Section 6. Concluding remarks are given in Section 7.

2 The random matrix framework

We use the following notation:

- \mathbb{R}^n is the set of real column vectors of length n , \mathbb{S}_{++}^n is the set of symmetric positive definite $n \times n$ matrices, and \mathbb{S}_+^n is the set of symmetric positive semi-definite $n \times n$ matrices.
- $\mathcal{N}(\mathbf{x}; m, P)$ denotes a multi-variate Gaussian probability density function (pdf) defined over the vector $\mathbf{x} \in \mathbb{R}^{n_x}$ with mean vector $m \in \mathbb{R}^{n_x}$, and covariance matrix $P \in \mathbb{S}_+^{n_x}$,

$$\mathcal{N}(\mathbf{x}; m, P) = \frac{e^{-\frac{1}{2}(\mathbf{x}-m)^\top P^{-1}(\mathbf{x}-m)}}{(2\pi)^{\frac{n_x}{2}} |P|^{\frac{1}{2}}}, \quad (1)$$

where $|\cdot|$ is the matrix determinant function.

- $\mathcal{IW}(X; \nu, V)$ denotes an inverse Wishart pdf defined over the matrix $X \in \mathbb{S}_{++}^d$ with scalar degrees of freedom $\nu > 2d$ and parameter matrix $V \in \mathbb{S}_{++}^d$, (Gupta and Nagar, 2000, Definition 3.4.1)

$$\mathcal{IW}(X; \nu, V) = \frac{2^{-\frac{\nu-d-1}{2}} |V|^{\frac{\nu-d-1}{2}}}{\Gamma_d\left(\frac{\nu-d-1}{2}\right) |X|^{\frac{d}{2}}} \text{etr}\left(-\frac{1}{2}X^{-1}V\right), \quad (2)$$

where $\text{etr}(\cdot) = \exp(\text{Tr}(\cdot))$ is exponential of the matrix trace, and $\Gamma_d(\cdot)$ is the multivariate gamma function. The multivariate gamma function can be expressed as a product of ordinary gamma functions, see e.g. (Gupta and Nagar, 2000, Theorem 1.4.1).

- $\mathcal{W}(X; w, W)$ denotes a Wishart pdf defined over the matrix $X \in \mathbb{S}_{++}^d$ with scalar degrees of freedom $w \geq d$ and parameter matrix $W \in \mathbb{S}_{++}^d$, (Gupta and Nagar, 2000, Definition 3.2.1)

$$\mathcal{W}(X; w, W) = \frac{2^{-\frac{wd}{2}} |X|^{\frac{w-d-1}{2}}}{\Gamma_d\left(\frac{w}{2}\right) |W|^{\frac{w}{2}}} \text{etr}\left(-\frac{1}{2} W^{-1} X\right). \quad (3)$$

- $\mathcal{GB}_d^{II}(X; a, b, \Omega, \Psi)$ denotes a generalized matrix-variate beta type-II pdf defined over the matrix $X \in \mathbb{S}_{++}^d$, with scalar parameters $a > \frac{d-1}{2}$, $b > \frac{d-1}{2}$ and matrix parameters $\Omega \in \mathbb{S}_{++}^d$, $\Psi \in \mathbb{S}_+^d$, (Gupta and Nagar, 2000, Definition 5.2.4)

$$\mathcal{GB}_d^{II}(X; a, b, \Omega, \Psi) = \frac{|X - \Psi|^{a - \frac{d+1}{2}} |\Omega + X|^{-(a+b)}}{\beta_d(a, b) |\Omega + \Psi|^{-b}}, \quad (4)$$

where $(X - \Psi) \in \mathbb{S}_{++}^d$ and $(\Omega - \Psi) \in \mathbb{S}_+^d$, and $\beta_d(a, b)$ is the multivariate beta function.

- $\mathbf{0}_d$ is an all zero $d \times d$ matrix, and \mathbf{I}_d is a $d \times d$ identity matrix.

Let ξ_k be the extended target state at time t_k , and let \mathbf{Z}^k denote the set of all measurements up to and including time t_k . The random matrix framework (Koch, 2008; Feldmann et al., 2011) defines the extended target state $\xi_k = (\mathbf{x}_k, X_k)$ as the combination of a kinematical state $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and an extension state $X_k \in \mathbb{S}_{++}^d$. The kinematical state \mathbf{x}_k contains states related to target kinematics, such as position, velocity and heading, while the extension state X_k is a random matrix representing the target extent. The posterior pdf of the extended target state ξ_k , conditioned on \mathbf{Z}^k , is modeled as Gaussian inverse Wishart (GIW) distributed (Feldmann et al., 2011)

$$p(\xi_k | \mathbf{Z}^k) = p(\mathbf{x}_k | X_k, \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (5a)$$

$$\approx p(\mathbf{x}_k | \mathbf{Z}^k) p(X_k | \mathbf{Z}^k) \quad (5b)$$

$$= \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \mathcal{IW}(X_k; \nu_{k|k}, V_{k|k}). \quad (5c)$$

This density approximates the kinematical and extension states as independent, however, as noted in (Feldmann et al., 2011), the measurement update step “provides for the interdependency between kinematics and extension estimation.” The random matrix framework limits the extended targets to be shaped as ellipses, however the ellipse shape is applicable to many real scenarios in which the target and sensor geometry is such that the target measurements resemble a cluster of detections, rather than a geometric structure (or for that matter a single detection).

Note that in a Bayesian state estimation recursion, we (typically) want the predicted pdf $p(\xi_{k+1}|\mathbf{Z}^k)$ to be of the same functional form as the posterior pdf $p(\xi_k|\mathbf{Z}^k)$. For a GIW distributed extended target (5c), this corresponds to obtaining the parameters $m_{k+1|k}$, $P_{k+1|k}$, $v_{k+1|k}$, and $V_{k+1|k}$ of the distribution

$$p(\xi_{k+1}|\mathbf{Z}^k) = \mathcal{N}(\mathbf{x}_{k+1}; m_{k+1|k}, P_{k+1|k}) \mathcal{IW}(X_{k+1}; v_{k+1|k}, V_{k+1|k}). \quad (6)$$

In previous work, see (Koch, 2008; Feldmann et al., 2011), the kinematical state \mathbf{x}_k is predicted using the Kalman filter prediction (Kalman, 1960). The extension state prediction is based on simple heuristics. Under the assumption that “the extension does not tend to change over time” (Feldmann et al., 2011), the inverse Wishart parameters are predicted such that $E[X_{k+1}] = E[X_k]$ and $\text{Cov}(X_{k+1}) > \text{Cov}(X_k)$. The following prediction update is used in (Feldmann et al., 2011),

$$v_{k+1|k} = 2d + 4 + e^{-T/\tau}(v_{k|k} - 2d - 4), \quad (7a)$$

$$V_{k+1|k} = \frac{v_{k+1|k} - 2d - 2}{v_{k|k} - 2d - 2} V_{k|k}, \quad (7b)$$

where T is the prediction time and τ is a design parameter. Note that (7a) is a minor modification of the prediction

$$v_{k+1|k} = e^{-T/\tau} v_{k|k}, \quad (8)$$

which is used in (Koch, 2008). The modification ensures that the expected value and covariance of X_k always are well-defined.

In addition to presenting the prediction update given above, Koch (2008) also discusses using a Wishart extension transition density,

$$p(X_{k+1}|X_k) = \mathcal{W}\left(X_{k+1}; n_{k+1}, \frac{X_k}{n_{k+1}}\right). \quad (9)$$

This transition density is used by Lian et al. (2010). A modified version of (9) is suggested by Lan and Rong Li (2012),

$$p(X_{k+1}|X_k) = \mathcal{W}\left(X_{k+1}; \delta_k, A_k X_k A_k^T\right), \quad (10)$$

where the $d \times d$ matrix A_k describes the extension time evolution, e.g. rotation or scaling of the extension.

The contribution of this paper is a further generalization of the idea to use a Wishart transition density. The presented prediction method allows extension transformations that are functions of the kinematical state.

3 Problem formulation

The state transition pdf $p(\xi_{k+1}|\xi_k)$ describes the time evolution of the extended target state from time t_k to time t_{k+1} . In Bayesian state estimation, the prediction

step consists of solving the integral

$$p(\xi_{k+1}|\mathbf{Z}^k) = \int p(\xi_{k+1}|\xi_k)p(\xi_k|\mathbf{Z}^k)d\xi_k. \quad (11)$$

Assume the following decomposition for the transition density,

$$p(\xi_{k+1}|\xi_k) \triangleq p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(X_{k+1}|\xi_k). \quad (12)$$

The integral (11), with posterior distribution (5b) and transition density (12), is

$$\iint p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(X_{k+1}|\xi_k)p(\mathbf{x}_k|\mathbf{Z}^k)p(X_k|\mathbf{Z}^k)d\mathbf{x}_k dX_k \quad (13a)$$

$$= \int p(\mathbf{x}_{k+1}|\mathbf{x}_k) \int p(X_{k+1}|\xi_k)p(X_k|\mathbf{Z}^k)dX_k p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k \quad (13b)$$

$$= \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(X_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k \quad (13c)$$

$$= p(\mathbf{x}_{k+1}, X_{k+1}|\mathbf{Z}^k). \quad (13d)$$

In general it does not hold that

$$p(\mathbf{x}_{k+1}, X_{k+1}|\mathbf{Z}^k) = p(\mathbf{x}_{k+1}|\mathbf{Z}^k)p(X_{k+1}|\mathbf{Z}^k), \quad (14)$$

and therefore (13c) is here approximated as

$$\begin{aligned} & \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(X_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k \\ & \approx \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k \int p(X_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k. \end{aligned} \quad (15)$$

This is equivalent to solving

$$p(\mathbf{x}_{k+1}|\mathbf{Z}^k) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}^k)d\mathbf{x}_k, \quad (16a)$$

$$p(X_{k+1}|\mathbf{Z}^k) = \int p(X_{k+1}|\xi_k)p(\xi_k|\mathbf{Z}^k)d\xi_k, \quad (16b)$$

instead of solving (11). Approximations are also made in (Koch, 2008; Feldmann et al., 2011; Lan and Rong Li, 2012) to obtain independent prediction integrals for the kinematic and extension state, and it seems that predicting the kinematic state and extension state independently can be justified in many practical scenarios, see e.g. (Koch, 2008; Feldmann et al., 2011; Lan and Rong Li, 2012). Note that the extension prediction (16b) provides for further interdependency between kinematics and extension estimation, in addition to the interdependency provided by the measurement update.

For the kinematical state, the transition density is modeled as

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) \triangleq \mathcal{N}(\mathbf{x}_{k+1}; f(\mathbf{x}_k), Q_{k+1}), \quad (17)$$

where $f(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is a state transition function, and Q_{k+1} is the process

noise covariance for the kinematic state. The function $f(\cdot)$ is generally nonlinear, see (Rong Li and Jilkov, 2003) for a thorough overview of state transition functions. The assumption that the time evolution of the kinematical state (17) is independent of the extension state neglects factors such as wind resistance, which can be modeled as a function of the target size, however the assumption is expected to hold in many practical situations.

Generalizing the Wishart transition densities of (Koch, 2008; Lan and Rong Li, 2012), described in (9) and (10), the extension transition density is modeled as

$$p(X_{k+1}|\xi_k) \triangleq \mathcal{W}\left(X_{k+1}; n_{k+1}, \frac{M_{\mathbf{x}_k} X_k M_{\mathbf{x}_k}^T}{n_{k+1}}\right), \quad (18)$$

where $n_{k+1} > d$ is a scalar design parameter, and the matrix transformation $M_{\mathbf{x}_k} \triangleq M(\mathbf{x}_k) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{d \times d}$ is a non-singular matrix valued function of the kinematic state.

The extension state's time evolution is modeled as being dependent on the kinematical state via a matrix transformation. The main motivation for this specific form is the modeling of rotation of extended targets. However, in general the function $M_{\mathbf{x}_k}$ can be selected arbitrarily, as long as the output is a non-singular $d \times d$ matrix. In terms of, e.g., group target tracking, the extension can grow or shrink over time, corresponding to $M_{\mathbf{x}_k}$ being a scaling of the extension.

The scalar design parameter n_{k+1} in (18) is analogous to the noise covariance Q_{k+1} in (17), i.e. it governs the extension state process noise. Let $\mathbb{M}_k = M_{\mathbf{x}_k} X_k M_{\mathbf{x}_k}^T$ and let $X^{[ij]}$ denote the (i, j) th element of the matrix X . By (Gupta and Nagar, 2000, Theorem 3.3.15) the expected value and variance of the (i, j) th element of $X_{k+1}|\mathbf{x}_k, X_k$ are

$$\mathbb{E}\left[X_{k+1}^{[ij]}|\mathbf{x}_k, X_k\right] = \mathbb{M}_k^{[ij]}, \quad (19a)$$

$$\text{Var}\left(X_{k+1}^{[ij]}|\mathbf{x}_k, X_k\right) = \frac{\left(\mathbb{M}_k^{[ij]}\right)^2}{n_{k+1}} + \frac{\mathbb{M}_k^{[ii]}\mathbb{M}_k^{[jj]}}{n_{k+1}}, \quad (19b)$$

i.e., given \mathbf{x}_k and X_k the variance decreases with increasing n_{k+1} . It can thus be said that a higher n_{k+1} implies less process noise for the extension state. Thus, the shorter the prediction time interval T is, the larger n_{k+1} should be, and in the limit $\lim_{T \rightarrow 0} n_{k+1} = \infty$ should hold. One way to model n_{k+1} as a function of prediction time is (Koch, 2008; Lian et al., 2010),

$$n_{k+1} = n e^{-T/\tau}, \quad (20)$$

with two scalar parameters n and τ . We elaborate further on n_{k+1} after we derive the main result of the paper.

The problem considered in this work is to, given a posterior density (5c) and the transition densities (12), (17), (18), obtain a solution to (16), where the predicted

density $p(\xi_{k+1}|\mathbf{Z}^k)$ is of the same functional form as (5c), i.e.

$$p(\xi_{k+1}|\mathbf{Z}^k) = p(\mathbf{x}_{k+1}|\mathbf{Z}^k)p(X_{k+1}|\mathbf{Z}^k) \quad (21a)$$

$$= \mathcal{N}(\mathbf{x}_{k+1}; m_{k+1|k}, P_{k+1|k}) \mathcal{IW}(X_{k+1}; \nu_{k+1|k}, V_{k+1|k}). \quad (21b)$$

4 Preliminaries

In this section we first give some known results, and then give three theorems, which are all needed in our derivation of a prediction update. For the pdf approximations below, the true densities are approximated by the minimization of the Kullback-Leibler divergence (KL-DIV) (Kullback and Leibler, 1951). The KL-DIV is defined for two pdfs $p(x)$ and $q(x)$ as

$$\text{KL}(p(x)||q(x)) = \int p(x) \log(p(x)/q(x)) dx. \quad (22)$$

Note that, when it comes to approximating distributions in a maximum likelihood sense, the KL-DIV is considered the optimal difference measure (Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009).

4.1 Known results

Let $p(X) = \mathcal{W}(X; v, V)$, and let M be any non-singular $d \times d$ matrix. The random matrix MXM is distributed as (Gupta and Nagar, 2000, Theorem 3.3.1)

$$p(MXM) = \mathcal{W}(MXM; v, MVM). \quad (23)$$

Let $p(X|V) = \mathcal{W}(X; n, V)$ and let $p(V) = \mathcal{IW}(V; \bar{v}, \bar{V})$. The marginal for X is (Gupta and Nagar, 2000, Problem 5.33)

$$p(X) = \mathcal{GB}_d^{II}\left(X; \frac{n}{2}, \frac{\bar{v} - d - 1}{2}, \bar{V}, \mathbf{0}_d\right). \quad (24)$$

4.2 Approximating a \mathcal{GB}_d^{II} with an \mathcal{IW}_d

Theorem 1. Let $p(X) = \mathcal{GB}_d^{II}(X; a, b, \Omega, \mathbf{0}_d)$, and let $q(X) = \mathcal{IW}(X; v, V)$ be the minimizer of the Kullback-Leibler (KL) divergence between $p(X)$ and $q(X)$ among all \mathcal{IW}_d -distributions, i.e.

$$q(X) \triangleq \arg \min_{q(\cdot)=\mathcal{IW}_d(\cdot)} \text{KL}(p(X)||q(X)). \quad (25)$$

Then V is given as

$$V = \frac{(v - d - 1)(2a - d - 1)}{2b} \Omega, \quad (26)$$

and v is the solution to the equation

$$\sum_{i=1}^d \left[\psi_0 \left(\frac{2a+1-i}{2} \right) - \psi_0 \left(\frac{2b+1-i}{2} \right) + \psi_0 \left(\frac{v-d-i}{2} \right) \right] - d \log \left(\frac{(v-d-1)(2a-d-1)}{4b} \right) = 0, \tag{27}$$

where $\psi_0(\cdot)$ is the digamma function (a.k.a. the polygamma function of order 0).
 \square

Proof: Given in Appendix A.3. \square

The equations for V (26) and v (27) in Theorem 1 correspond to matching the expected value of X^{-1} and $\log|X|$,

$$E_q [X^{-1}] = E_p [X^{-1}], \tag{28a}$$

$$E_q [\log|X|] = E_p [\log|X|]. \tag{28b}$$

Notice that in Theorem 1, substituting a value for v into (26) gives the analytical solution for V . The parameter v can be found by applying a numerical root-finding algorithm to (27), see e.g. (Stoer and Bulirsch, 1993, Section 5.1). Examples include Newton-Raphson or modified Newton algorithms, see e.g. (Stoer and Bulirsch, 1993, Section 5.4), for more alternatives see e.g. (Stoer and Bulirsch, 1993, Chapter 5). In the following corollary, we supply an alternative to root-finding to obtain a value for v .

Corollary 1. *A closed form solution for v can be obtained using only (26) together with matching the first order moments. The expected values of the densities $p(\cdot)$ and $q(\cdot)$ are (Gupta and Nagar, 2000, Theorems 5.3.20, 3.4.3)*

$$E_p [X] = \frac{2a}{2b-d-1} \Omega, \tag{29a}$$

$$E_q [X] = \frac{V}{v-2d-2} = \frac{v-d-1}{v-2d-2} \frac{2a-d-1}{2b} \Omega. \tag{29b}$$

Equating the expected values and solving for v gives

$$v = (d+1) \frac{\frac{2a-d-1}{2b} - 2 \frac{2a}{2b-d-1}}{\frac{2a-d-1}{2b} - \frac{2a}{2b-d-1}}. \tag{30}$$

\square

Matching the expected values, as in Corollary 1, can be seen as an approximation of matching the expected values of the log determinant (28b). Indeed, with numerical simulations one can show that the v given by (30) is approximately equal to the solution of (27), the difference is typically on the order of one tenth of a degree of freedom.

References Koch (2008); Gupta and Nagar (2000); Lan and Rong Li (2012) contain

discussions about using moment matching to approximate a \mathcal{GB}_d^{II} -distribution with a \mathcal{IW}_d -distribution. Theorem 1 defines an approximation by minimising the KL divergence, which results in matching the expected values (28). The KL criterion is well-known in the literature for its moment-matching characteristics, see e.g. Bishop (2006); Minka (2001).

4.3 Approximating the density of \mathbb{V}_x with a \mathcal{W}_d

Theorem 2. Let \mathbf{x} be Gaussian distributed with mean m and covariance P , and let $\mathbb{V}_x \triangleq \mathbb{V}(\mathbf{x}) \in \mathbb{S}_{++}^{n_x}$ be a matrix valued function of \mathbf{x} . Let $p(\mathbb{V}_x)$ be the density of \mathbb{V}_x induced by the random variable \mathbf{x} , and let $q(\mathbb{V}_x) = \mathcal{W}(\mathbb{V}_x; s, S)$ be the minimizer of the KL-divergence between $p(\mathbb{V}_x)$ and $q(\mathbb{V}_x)$ among all \mathcal{W} -distributions, i.e.

$$q(\mathbb{V}_x) \triangleq \arg \min_{q(\cdot) \in \mathcal{W}(\cdot)} \text{KL}(p(\mathbb{V}_x) \| q(\mathbb{V}_x)). \quad (31)$$

Then S is given as

$$S = \frac{1}{s} \mathbb{C}_{II} \quad (32)$$

and s is the solution to the equation

$$d \log \left(\frac{s}{2} \right) - \sum_{i=1}^{n_x} \psi_0 \left(\frac{s-i+1}{2} \right) + \mathbb{C}_I - \log |\mathbb{C}_{II}| = 0 \quad (33)$$

where $\mathbb{C}_I \triangleq \mathbb{E}[\log |\mathbb{V}_x|]$ and $\mathbb{C}_{II} \triangleq \mathbb{E}[\mathbb{V}_x]$. \square

Proof: Given in Appendix A.4. \square

Corollary 2. \mathbb{C}_I and \mathbb{C}_{II} can be calculated using a Taylor series expansion of \mathbb{V}_x around $\mathbf{x} = m$. A third order expansion yields

$$\mathbb{C}_I \approx \log |\mathbb{V}_m| + \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \left. \frac{d^2 \log |\mathbb{V}_x|}{d\mathbf{x}_j d\mathbf{x}_i} \right|_{\mathbf{x}=m} P_{ij}, \quad (34a)$$

$$\mathbb{C}_{II} \approx \mathbb{V}_m + \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \left. \frac{d^2 \mathbb{V}_x}{d\mathbf{x}_j d\mathbf{x}_i} \right|_{\mathbf{x}=m} P_{ij}. \quad (34b)$$

In (34) the i :th element of the vector \mathbf{x} and the i, j :th element of the matrix P are \mathbf{x}_i and P_{ij} , respectively. Moreover, the matrix \mathbb{V}_m is the function \mathbb{V}_x evaluated at the mean m of the random variable \mathbf{x} . \square

The equations for S (32) and s (33) in Theorem 2 correspond to matching the expected values of \mathbb{V}_x and $\log |\mathbb{V}_x|$,

$$\mathbb{E}_q[\mathbb{V}_x] = \mathbb{E}_p[\mathbb{V}_x], \quad (35a)$$

$$\mathbb{E}_q[\log |\mathbb{V}_x|] = \mathbb{E}_p[\log |\mathbb{V}_x|]. \quad (35b)$$

Similarly to (27), numerical root-finding can be used to calculate a solution to (33). Note that using a moment matching approach similar to Corollary 1 to find a value for s is not advisable, since this would lead to further approximations (because the true distribution $p(\nabla_{\mathbf{x}})$ is unknown), and would possibly require a more complicated numerical solution.

4.4 Marginalizing $\mathcal{IW}_d(X|V)\mathcal{W}(V)$ over V

This result is similar to the property give in (Gupta and Nagar, 2000, Problem 5.33). If $p(S|\Sigma) = \mathcal{W}(S; n, \Sigma)$ and $p(\Sigma) = \mathcal{IW}(\Sigma; m, \Psi)$ then the marginal density of S is

$$p(S) = \mathcal{GB}_d^{II} \left(X; \frac{n}{2}, \frac{m-d-1}{2}, \Psi, \mathbf{0}_d \right). \quad (36)$$

Theorem 3. Let $p(X|V) = \mathcal{IW}(X; v, V/\gamma)$ and let $p(V) = \mathcal{W}(V; s, S)$. The marginal for X is

$$p(X) = \mathcal{GB}_d^{II} \left(X; \frac{s}{2}, \frac{v-d-1}{2}, \frac{S}{\gamma}, \mathbf{0}_d \right). \quad (37)$$

□

Proof: Given in Appendix A.5. □

5 A new prediction update for the extension

In this section we present the new approach to prediction, first for the kinematical state in Section 5.1 and the for the extension state in Section 5.2.

5.1 Predicting the kinematical state

For the kinematical state we have

$$p(\mathbf{x}_{k+1}|\mathbf{Z}^k) = \int \mathcal{N}(\mathbf{x}_{k+1}; f(\mathbf{x}_k), Q) \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) d\mathbf{x}_k, \quad (38)$$

In case $f(\mathbf{x}_k)$ is a linear function, the solution to the integral (38) is given by the Kalman filter prediction (Kalman, 1960). In general $f(\mathbf{x}_k)$ is non-linear, in which case it is straightforward to solve the integral (38) approximately. Using the extended Kalman filter prediction formulas, see e.g. (Jazwinski, 1970), the predicted mean $m_{k+1|k}$ and covariance $P_{k+1|k}$ are

$$m_{k+1|k} = f(m_{k|k}), \quad P_{k+1|k} = F_{k|k} P_{k|k} F_{k|k}^T + Q \quad (39)$$

where $F_{k|k} \triangleq \nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=m_{k|k}}$ is the gradient of $f(\cdot)$ evaluated at the mean $m_{k|k}$.

5.2 Predicting the extension state

For the extension state we have

$$p(X_{k+1}|\mathbf{Z}^k) = \iint p(X_{k+1}|\mathbf{x}_k, X_k)p(\mathbf{x}_k, X_k|\mathbf{Z}^k)d\mathbf{x}_k dX_k \quad (40a)$$

$$= \iint \mathcal{W}\left(X_{k+1}; n_{k+1}, \frac{M_{\mathbf{x}_k} X_k M_{\mathbf{x}_k}^T}{n_{k+1}}\right) \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \\ \times \mathcal{IW}(X_k; \nu_{k|k}, V_{k|k}) d\mathbf{x}_k dX_k. \quad (40b)$$

Using the properties given in (23) and (24), the integral (40b) becomes

$$p(X_{k+1}|\mathbf{Z}^k) = \int \mathcal{GB}_d^{II}\left(X_{k+1}; \frac{n_{k+1}}{2}, \frac{\nu_{k|k} - d - 1}{2}, \frac{M_{\mathbf{x}_k} V_{k|k} M_{\mathbf{x}_k}^T}{n_{k+1}}, \mathbf{0}_d\right) \\ \times \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) d\mathbf{x}_k. \quad (41)$$

Unfortunately, the integral (41) has no analytical solution, it has to be solved using approximations.

In what follows, we first show how (7) can be heuristically modified to allow for transformations of the extension, and then the prediction method from (Lan and Rong Li, 2012) is briefly described. Lastly the main result of the paper is given, a new prediction update for the extension state.

Heuristic modification of (7)

Note first that the prediction (7) corresponds to the case $M_{\mathbf{x}_k} = \mathbf{I}_d$. The prediction (7) is hereafter called method 1 (M1).

Including a non-identity transformation matrix $M(\cdot)$ in the prediction process can be done heuristically, e.g. by replacing (7b) with

$$V_{k+1|k} = \frac{\nu_{k+1|k} - 2d - 2}{\nu_{k|k} - 2d - 2} M_{m_{k|k}} V_{k|k} M_{m_{k|k}}^T. \quad (42)$$

This prediction for the extension evaluates $M_{\mathbf{x}_k}$ at the last estimated kinematic state $m_{k|k}$, and can thus capture e.g. rotations. However, it neglects the kinematic state uncertainty $P_{k|k}$ completely. The prediction given by (7a) and (42) is hereafter called method 2 (M2).

Prediction method from (Lan and Rong Li, 2012)

An alternative to (42) is to replace $M_{\mathbf{x}_k}$ by $M_{m_{k|k}}$ in (41). In this case the integral (41) has an analytical solution, and Theorem 1 can then be used to approximate the the \mathcal{GB}_d^{II} -density as an \mathcal{IW}_d -distribution. A similar approach is taken in (Lan and Rong Li, 2012), and the extension transition density used in (Lan and Rong Li, 2012) was given in (10). The matrix A_k in (10) is a parameter, and is not dependent on the kinematical state. In (Lan and Rong Li, 2012) the authors use a type of moment matching to approximate the density, instead of minimization of the KL-DIV. The prediction method from (Lan and Rong Li, 2012) is hereafter called method 3 (M3).

Note that if $A_k = M_{m_{k|k}}/\sqrt{\delta_k}$, and if τ and δ_k are chosen correctly, M2 is equivalent to M3. In Section 6.1 we show how τ can be chosen for this equivalence to hold. The transition density (10) is used in a multiple model framework in (Lan and Rong Li, 2012), with m different modes with corresponding parameters $\delta_k^{(m)}$ and $A_k^{(m)}$. The extension modes correspond to, e.g., no rotation, rotation θ rad, and rotation $-\theta$ rad. In the results section it will be clear from context if it is the single mode, or multiple mode, version of M3 that is referred to.

New prediction for the extension state

Using Theorem 1, the \mathcal{GB}_d^{II} -distribution in (41) can be approximated as an \mathcal{IW}_d -distribution,

$$p(X_{k+1}|\mathbf{Z}^k) \approx \int \mathcal{IW}\left(X_{k+1}; v_{k|k}, \frac{M_{\mathbf{x}_k} V_{k|k} M_{\mathbf{x}_k}^T}{\gamma_{k|k}}\right) \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) d\mathbf{x}_k, \quad (43)$$

where $v_{k|k}$ is calculated using Corollary 1 by setting $a = \frac{n_{k+1}}{2}$, $b = \frac{v_{k|k} - d - 1}{2}$, and $\gamma_{k|k} \triangleq \frac{2bn_{k+1}}{(v_{k|k} - d - 1)(2a - d - 1)}$. Using the variable substitution $\mathbb{V}_{\mathbf{x}_k} \triangleq M_{\mathbf{x}_k} V_{k|k} M_{\mathbf{x}_k}^T$, we obtain

$$p(X_{k+1}|\mathbf{Z}^k) \approx \int \mathcal{IW}\left(X_{k+1}; v_{k|k}, \frac{M_{\mathbf{x}_k} V_{k|k} M_{\mathbf{x}_k}^T}{\gamma_{k|k}}\right) \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) d\mathbf{x}_k \quad (44a)$$

$$= \int \mathcal{IW}\left(X_{k+1}; v_{k|k}, \frac{\mathbb{V}_{\mathbf{x}_k}}{\gamma_{k|k}}\right) p(\mathbb{V}_{\mathbf{x}_k}) d\mathbb{V}_{\mathbf{x}_k}. \quad (44b)$$

In (44a) the \mathcal{IW}_d density depends on \mathbf{x}_k only through $M_{\mathbf{x}_k} V_{k|k} M_{\mathbf{x}_k}^T$, and the second equality then follows as a result of the variable substitution and standard probability theory for variable substitutions, see e.g. (Gut, 1995, Theorem 2.1). Note that $\mathbb{V}_{\mathbf{x}_k}$ is a $d \times d$ random matrix, and $p(\mathbb{V}_{\mathbf{x}_k})$ is a matrix variate density. Because exact calculation of $p(\mathbb{V}_{\mathbf{x}_k})$ is prohibitively difficult, we use Theorem 2 to approximate $p(\mathbb{V}_{\mathbf{x}_k})$ by a Wishart distribution. Note that Theorem 2 uses the parameters of $\mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k})$ in order to construct the Wishart approximation. This gives

$$p(X_{k+1}|\mathbf{Z}^k) \approx \int \mathcal{IW}\left(X_{k+1}; v_{k|k}, \frac{\mathbb{V}_{\mathbf{x}_k}}{\gamma_{k|k}}\right) \mathcal{W}(\mathbb{V}_{\mathbf{x}_k}; s_{k|k}, S_{k|k}) d\mathbb{V}_{\mathbf{x}_k}, \quad (45)$$

where $s_{k|k}$ and $S_{k|k}$ are calculated by setting $m = m_{k|k}$ and $P = P_{k|k}$ in Theorem 2, and \mathbb{C}_I and \mathbb{C}_{II} are computed using Corollary 2. Using Theorem 3 the marginal for X_{k+1} , which is the solution to the integral of (45), is given as

$$p(X_{k+1}|\mathbf{Z}^k) \approx \mathcal{GB}_d^{II}\left(X_{k+1}; a_{k+1|k}, b_{k+1|k}, \Omega_{k+1|k}, \mathbf{0}_d\right) \quad (46)$$

where $a_{k+1|k} \triangleq \frac{s_{k|k}}{2}$, $b_{k+1|k} \triangleq \frac{v_{k|k} - d - 1}{2}$ and $\Omega_{k+1|k} \triangleq \frac{S_{k|k}}{\gamma_{k|k}}$. Finally, using Theorem 1 once again we obtain

$$p(X_{k+1}|\mathbf{Z}^k) \approx \mathcal{IW}\left(X_{k+1}; v_{k+1|k}, V_{k+1|k}\right), \quad (47)$$

where the prediction updated parameters $v_{k+1|k}$ and $V_{k+1|k}$ are

$$v_{k+1|k} = (d+1) \frac{\frac{2a_{k+1|k}-d-1}{2b_{k+1|k}} - 2 \frac{2a_{k+1|k}}{2b_{k+1|k}-d-1}}{\frac{2a_{k+1|k}-d-1}{2b_{k+1|k}} - \frac{2a_{k+1|k}}{2b_{k+1|k}-d-1}}, \quad (48a)$$

$$V_{k+1|k} = \frac{(v_{k+1|k} - d - 1)(2a_{k+1|k} - d - 1)}{2b_{k+1|k}} \Omega_{k+1|k}. \quad (48b)$$

Hereafter this prediction update is called method 4 (M4). This method improves upon the prediction updates M2 and M3 by also considering the kinematic state uncertainty.

5.3 Another look at the parameter n_{k+1}

In this section we elaborate on the parameter n_{k+1} in the extension state transition density. Under the assumption $M_x = I_d$ we have

$$p(X_{k+1} | \mathbf{Z}^k) = \mathcal{GB}_d^{II} \left(X_{k+1}; \frac{n_{k+1}}{2}, \frac{v_{k|k} - d - 1}{2}, \frac{V_{k|k}}{n_{k+1}}, \mathbf{0}_d \right), \quad (49)$$

and the expected value and variance of the (i, j) th element $X_{k+1}^{[ij]}$ are

$$\mathbb{E} \left[X_{k+1}^{[ij]} \right] = \mathbb{E} \left[X_k^{[ij]} \right], \quad (50a)$$

$$\text{Var} \left(X_{k+1}^{[ij]} \right) = \underbrace{\left(1 + \frac{v_{k|k} - 2d - 2}{n_{k+1}} \right)}_{\triangleq \eta_{k+1}} \text{Var} \left(X_k^{[ij]} \right). \quad (50b)$$

We see that (50) corresponds to exponential forgetting prediction for the (i, j) th element, see e.g. (Gustafsson et al., 2010). The forgetting factor is $0 < \eta_{k+1}^{-1} < 1$, and the effective window length is

$$w_e = \frac{1}{1 - \eta_{k+1}^{-1}} = 1 + \frac{n_{k+1}}{v_{k|k} - 2d - 2} \quad (51)$$

time steps. Using exponential forgetting prediction with window length w_e approximately means that we only “trust” the information that was contained in the measurements from the last w_e time steps. This gives us a hint as to how a specific value of n_{k+1} could be set, either as a global constant, or dynamically for each individual target at each time step.

An alternative way to interpret n_{k+1} starts with Corollary 1. We can rewrite (30) to obtain

$$v_{k+1|k} = v_{k|k} - \underbrace{\frac{(v_{k|k} - 2d - 2)(v_{k|k} - d - 1)}{n_{k+1} + v_{k|k} - 2d - 2}}_{N(v_{k|k}, n_{k+1})}, \quad (52)$$

where $N(\cdot)$ is a scalar valued function of $v_{k|k}$ and n_{k+1} . This is analogous to the

measurement update (Feldmann et al., 2011)

$$v_{k+1|k+1} = v_{k+1|k} + N_{z,k+1}, \quad (53)$$

where $N_{z,k+1}$ is the number of measurements at time step $k+1$. Thus, we can view the prediction as “removing” degrees of freedom corresponding to $N(v_{k|k}, n_{k+1})$ measurements.

6 Simulations

This section presents results from simulations which compare the new prediction method M4 to the methods M1, M2, and M3. The main focus is on the prediction of the extension state.

In all simulations of M4, Corollary 1 is used to calculate $v_{k|k}$. For computing the quantity $s_{k|k}$, (33) is solved numerically using the iterative Halley’s method (Ortega and Rheinboldt, 1970). This requires the digamma function to be computed, which is performed in MATLAB using the function `psi(·)`. Simulations have shown that $s_{k|k}$ is computed, on average, in just 4 iterations. Note that the only part of M4 that requires a numerical solution is the calculation of $s_{k|k}$. All other quantities required are calculated using their respective closed form expressions.

In the following subsections, first a method to determine the parameter τ in M1 and M2 is given in Section 6.1, then a difference measure for pdfs is presented in Section 6.2. This is followed by simulation results for one dimensional extensions in Section 6.3, and for two dimensional extensions in Section 6.4.

6.1 Determining τ

M1 and M2 contain the parameter τ , which is a “time constant related to the agility with which the object may change its extension over time” (Feldmann et al., 2011). Neither Koch (Koch, 2008) nor Feldmann *et al* (Feldmann et al., 2011) elaborate on how τ is best determined. To make as fair comparison as possible, here Theorem 1 is used to determine τ . By Theorem 1, the following holds,

$$\int \mathcal{IW}\left(X_+; n, \frac{X}{n}\right) \mathcal{W}(X; v, V) dX \approx \mathcal{IW}(X_+; v_+, V_+). \quad (54)$$

By setting v_+ equal to (7a), τ can be determined for any combination of T , v , and n . With this choice of τ , all prediction methods yield the same result when $M_x = \mathbf{I}_d$.

6.2 Difference measure for probability density functions

In order to measure the algorithms’ prediction performances, a distance measure between two pdfs $p(x)$ and $q(x)$ is needed. Here the L_2 -norm is used,

$$\Delta_{\text{pdf}}(p(x), q(x)) \triangleq \int |p(x) - q(x)|^2 dx. \quad (55)$$

In order to calculate the integral numerically, a uniform discretization is made over the union of the supports of $p(x)$ and $q(x)$.

6.3 Results in one dimension

This section presents results for a one dimensional ($d = 1$) extension X . The kinematic state \mathbf{x}_k is also selected as one dimensional, i.e. $n_x = d = 1$. The transformation function $M(\mathbf{x}_k)$ is given as

$$M(\mathbf{x}_k) = 1 + \mathbf{x}_k^2. \quad (56)$$

The integral in (55) is computed with a discretization over the interval $[0, 1000]$ with a discretization interval of length 0.1.

Accuracy of Theorem 1

The accuracy of Theorem 1, i.e. of the approximation

$$\mathcal{GB}_d^{II} \left(X_+; \frac{n}{2}, \frac{\nu - d - 1}{2}, \frac{V}{n}, \mathbf{0}_d \right) \approx \mathcal{IW}(X_+; \nu_+, V_+), \quad (57)$$

is evaluated for different values of the parameters n and ν by computing

$$\Delta_{\text{pdf}}(\mathcal{GB}_d^{II}, \mathcal{IW}_d) \quad (58)$$

for each combination of n and ν . The results, see Figure 1a, show that the approximation is least accurate when n is small. A small n corresponds to a very short effective window w_e , see (51) and Section 5.3.

Accuracy of Theorem 2

Let $\mathbb{V}_x = M_x V_0 M_x$, where M_x is given in (56) and $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; m, \sigma)$. The accuracy of Theorem 2, i.e. approximation of the pdf of \mathbb{V}_x with a Wishart distribution, is evaluated for different values of the parameter σ , when $m = 2$ and $V_0 = 1$. For each σ , an empirical pdf $p(\mathbb{V}_x)$ is computed using 10^7 samples from $\mathcal{N}(\mathbf{x}; m, \sigma)$. The results, see Figure 1b, show that, as expected, the approximation becomes less accurate as σ becomes larger. While the result in Figure 1b is specific for the transformation (56), the observation that $\Delta_{\text{pdf}}(p(\mathbb{V}_x), \mathcal{W}_d)$ increases with σ can be expected to hold for other transformation functions as well.

Accuracy of the new prediction

The following parameter settings are used for the distribution (5c),

$$\begin{aligned} \nu_{k|k} &= 50, & V_0 &= 1, & V_{k|k} &= (\nu_{k|k} - 2d - 2) V_0, \\ n_{k+1} &= 50, & m_{k|k} &= 2, & P_{k|k} &= \sigma_{k|k} \in [0.01, 1]. \end{aligned}$$

For each $\sigma_{k|k}$ value, a total of 10^7 samples were generated from (5c) and each sample is predicted by sampling from (18). The resulting empirical pdf (emp) over X_{k+1} is compared to the pdfs obtained by M1, M2 and M4. Remember that when τ is computed as in Section 6.1 and $A_k = M_{m_{k|k}} / \sqrt{\delta_k}$, M2 is equivalent to the single mode version of M3. For another choice of A_k , the error for M3 would be larger than for M2.

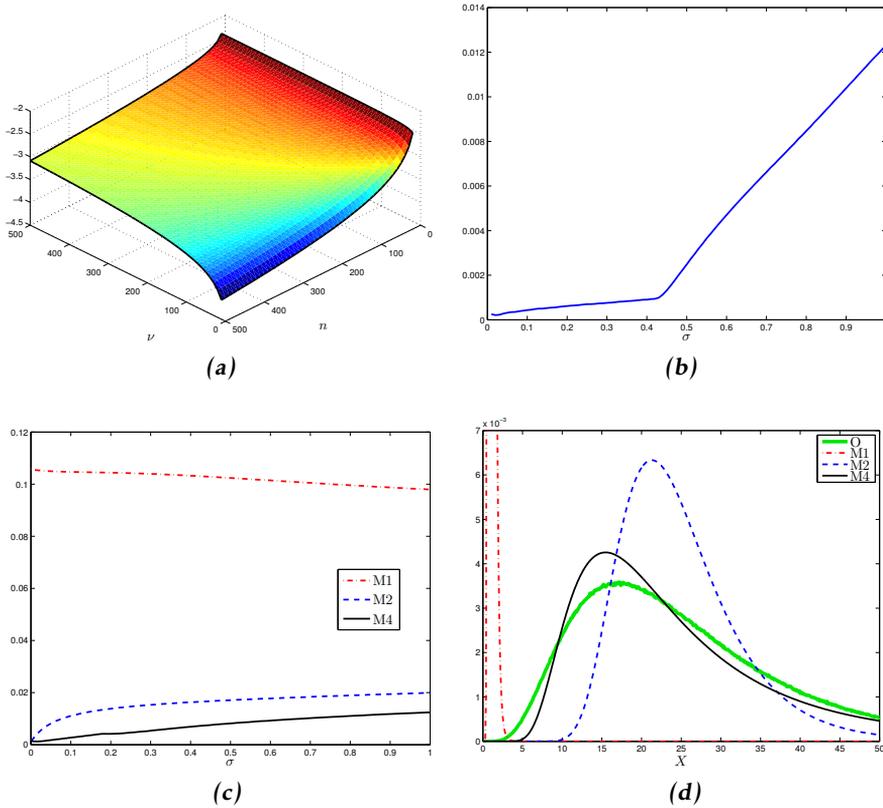


Figure 1: Results for one dimensional extensions. (a) The log-distances $\log_{10}(\Delta_{pdf}(p(x), q(x)))$ when a \mathcal{GB}_d^I is approximated as an \mathcal{IW}_d . The approximation is least accurate when the parameter n (cf. (18)) is small. (b). The distances $\Delta_{pdf}(p(\mathbb{V}_x), \mathcal{W})$ when \mathbf{x} is normal distributed and the distribution over \mathbb{V}_x is approximated as a \mathcal{W}_d . As expected, the accuracy of the approximation decreases when the uncertainty of \mathbf{x} increases. (c) The distances $\Delta_{pdf}(\cdot, \cdot)$ between the empirical distribution and the three prediction methods for different values of σ . The same transformation function M_x and same values of σ as were used in (b) are used here. (d) Comparison of the empirical distribution and the three different predicted distributions for $\sigma = 0.3$, the legend refers to the empirical distribution (O) and the three methods M1, M2 and M4. The suggested new prediction outperforms the other methods.

Figure 1c shows $\Delta_{pdf}(\text{emp}, M_i)$ for different values of the parameter σ . For all values of σ , M4 outperforms the other two methods. Figure 1d shows the pdfs for the case $\sigma = 0.3$. Again it is evident that M4 is the best approximation of the empirical distribution.

6.4 Results in two dimensions

This section presents results for a two dimensional ($d = 2$) extension. A constant turn-rate (CT) model with polar velocity (Rong Li and Jilkov, 2003) is considered. The kinematic state $\mathbf{x}_k = [x_k, y_k, v_k, \phi_k, \omega_k]^T$ contains the (x_k, y_k) -position in Cartesian coordinates, the speed v_k , the heading ϕ_k and the turn-rate ω_k . With this kinematic state, it is intuitive to let the transformation function be a rotation matrix

$$M(\mathbf{x}_k) = \begin{bmatrix} \cos(T\omega_k) & -\sin(T\omega_k) \\ \sin(T\omega_k) & \cos(T\omega_k) \end{bmatrix}. \quad (59)$$

Performance evaluation

For single step prediction, the predicted expected values $E[X_{k+1}]$ and covariances $\text{Cov}(X_{k+1})$ are compared. Because the covariance of the extension matrix is a $d^2 \times d^2$ matrix (Gupta and Nagar, 2000, Definition 1.2.6), we are going to constrain ourselves to illustrate only the $d \times d$ covariance matrix of the diagonal entries of the predicted extension matrix.

For single and multiple maneuvering targets, the predicted root mean square errors (RMSE) are computed over N_s Monte Carlo runs. The predicted kinematical state position RMSE, and the extension state RMSE, are computed as follows,

$$\text{RMSE}_k^{\mathbf{x}} = \left(\frac{1}{N_s} \sum_{i=1}^{N_s} \left(\hat{x}_{k|k-1}^{(i)} - x_k \right)^2 + \left(\hat{y}_{k|k-1}^{(i)} - y_k \right)^2 \right)^{\frac{1}{2}}, \quad (60a)$$

$$\text{RMSE}_k^X = \left(\frac{1}{N_s} \sum_{i=1}^{N_s} \text{Tr} \left(\left(\hat{X}_{k|k-1}^{(i)} - X_k \right)^2 \right) \right)^{\frac{1}{2}}, \quad (60b)$$

where x_k, y_k and X_k are the true position and extension, and $\hat{x}_{k|k-1}^{(i)}, \hat{y}_{k|k-1}^{(i)}$ and $\hat{X}_{k|k-1}^{(i)}$ are the predicted position and extension from the i th Monte Carlo run.

Single step prediction

The following parameter settings are used for the distribution (5c).

$$\begin{aligned} v_{k|k} &= 50, & V_{k|k} &= (v_{k|k} - 2d - 2) V_0, \\ V_0 &= \text{diag}([5, 2]), & n_{k+1} &= 50, \\ \omega_{k|k} &= 0 \text{ or } 45 [\text{deg}], & P_\omega &= 1 \text{ or } 20 [\text{deg}], \end{aligned}$$

where P_ω is the standard deviation for $\omega_{k|k}$. For each of the four parameter combinations, a total of 10^5 samples were generated, and each sample was then predicted by sampling from (18). The resulting sample mean X_{k+1}^O (of the extent matrix) and sample covariance C_{k+1}^O (of the diagonal elements of the extent matrix) are compared to the expected value and covariance given by M1, M2 and M4. Note again that M2 and single mode M3 gives equivalent results.

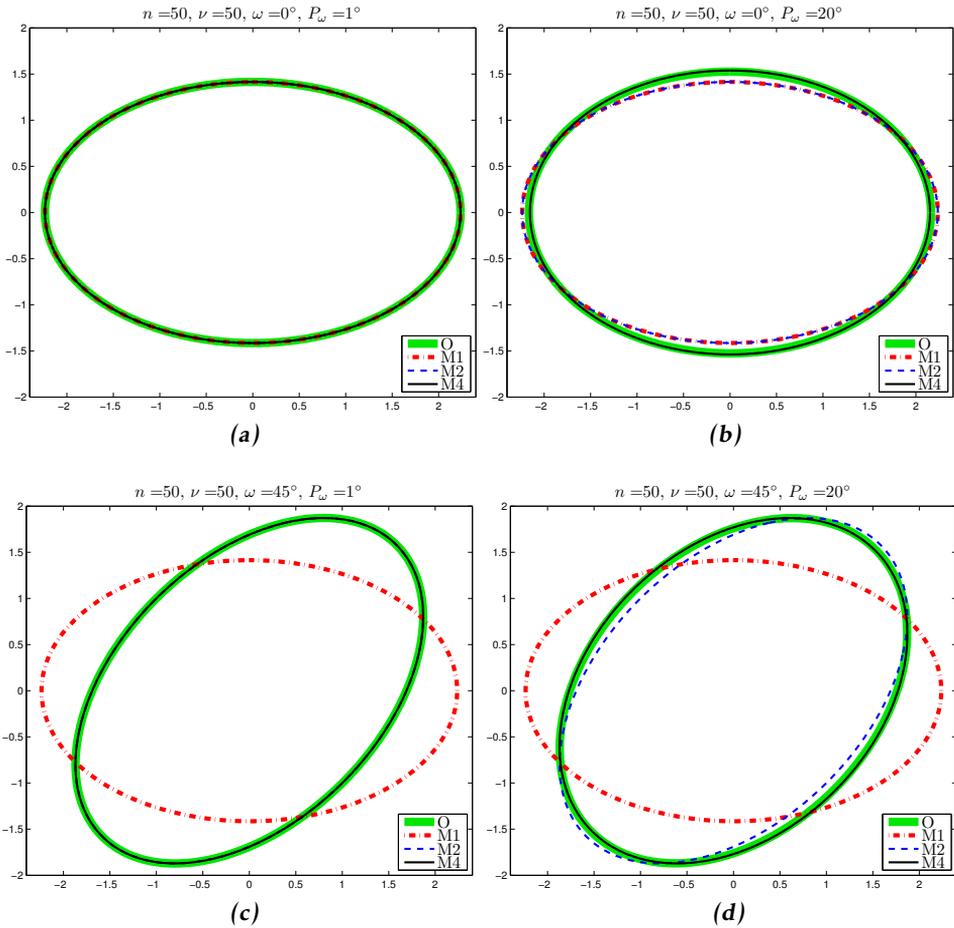


Figure 2: Results for two dimensional extensions, showing the expected value of the predicted extension. The legend refers to the empirical distribution (O) and the three methods $M1$, $M2$ and $M3$.

The results are shown in Figure 2 and Figure 3. It is evident that $M1$ has the worst performance of all three methods, because it does not take the rotation of the extension into account. $M2$ performs identically to $M4$ when P_ω is small, however for larger P_ω the sample mean X_{k+1}^O is slightly distorted, which $M2$ does not capture, and $M2$'s covariance is underestimated compared to C_{k+1}^O . $M4$, in comparison, captures the distorted shape of the sample mean, and $M4$'s covariance is not underestimated, rather it is slightly overestimated compared to C_{k+1}^O . Overestimation of the covariance is in general seen as more benign than underestimation, which can cause instability. Moreover, the increase of the covariance over the correct one C_{k+1}^O can be interpreted as a compensation for the approxi-

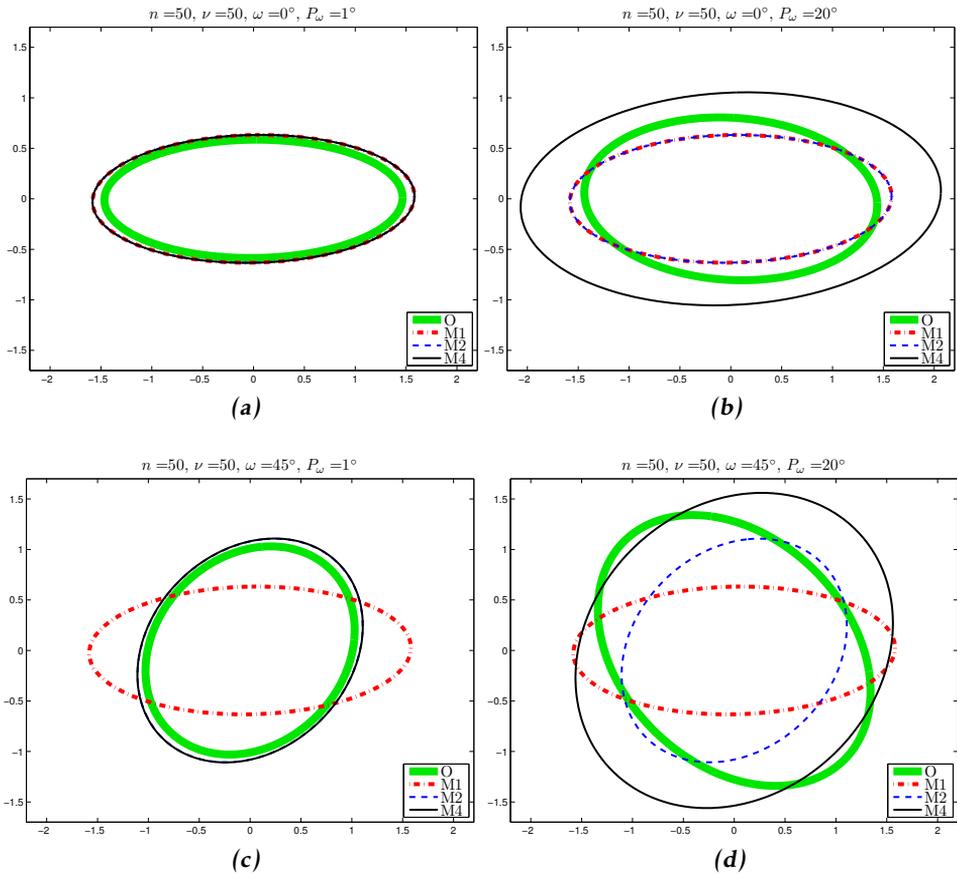


Figure 3: Results for two dimensional extensions, showing the covariance matrices corresponding to the diagonal elements of the predicted extension. The legend refers to the empirical distribution (O) and the three methods M1, M2 and M3.

mations made during the calculation. As a result, M4 outperforms M1 and M2 in terms of both the first and second order moments of the predicted pdf over X_{k+1} .

Single maneuvering target

Two single maneuvering target scenarios were simulated. In Figure 4a the target moves forward with constant velocity (CV) for 25 time steps, and then maneuvers with constant turn rate (CT) $\omega_k = 10$ deg per time step for 35 time steps. In Figure 4b the target moves forward with constant velocity for 25 time steps, and then maneuvers with a variable turn rate ω_k for 50 time steps. The turn rate first increased from 0 to 20 deg per time step, and then decreased to 0 deg per time step. The last five time steps is CV motion. In both scenarios, the target generated

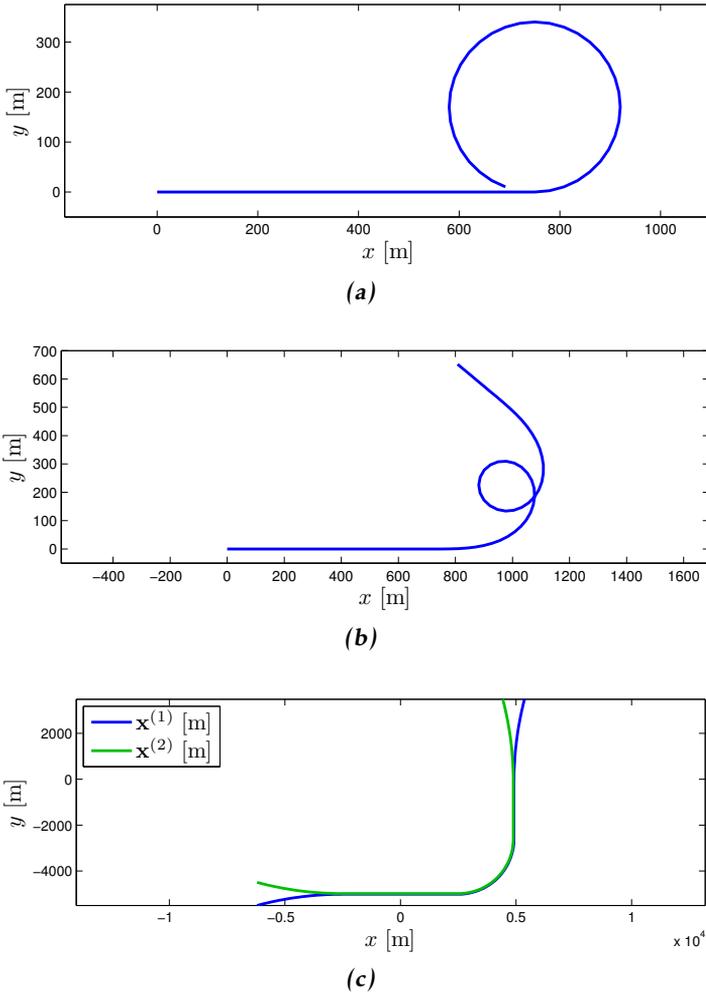


Figure 4: True target tracks used in simulations. In (a) and (b) the target starts at the origin. In (c) the two targets start in the bottom left.

10 measurements in each time step, and there were no clutter measurements.

For these two scenarios the multiple mode version of M3 was implemented, refer to (Lan and Rong Li, 2012) for details. The three extension evolution modes correspond to (1) no change with small process noise, (2) rotation θ deg with large process noise, and (3) rotation $-\theta$ deg with large process noise, where θ is a manually set model parameter. In each mode, the kinematical state is predicted according to a CV model (Lan and Rong Li, 2012).

The multiple mode version of M3 is compared to the CT version of M4 outlined above. Note that (Lan and Rong Li, 2012) also includes a new model for the

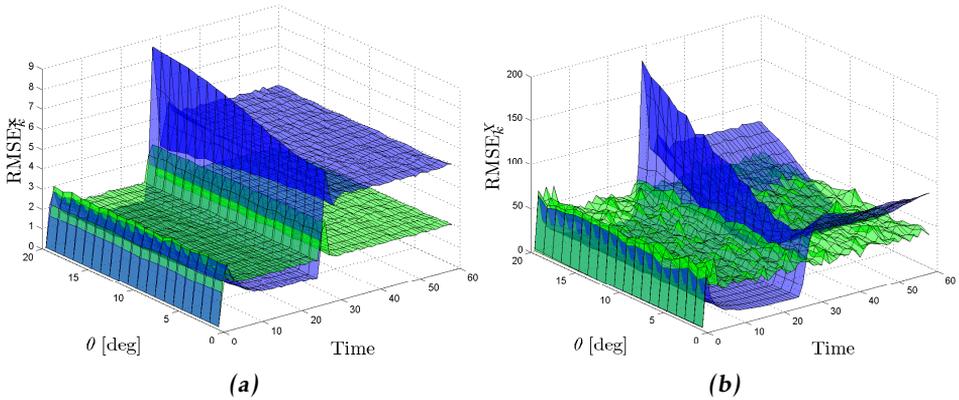


Figure 5: Results for single target tracking for the true track in Fig. 4a, prediction M3 in blue and prediction M4 in green. (a) and (b) show the position and extension RMSE, respectively. From time 25 to time 60 the true turn rate is 10 deg. For M3, the position RMSE increases during the maneuver, but is independent of the rotation parameter θ . However the extension RMSE increases with increasing parameter error. Note that for small parameter errors, M3 has lower extension RMSE than M4.

measurement update, however in this paper we study only the prediction update and therefore use the standard measurement update (Koch, 2008; Feldmann et al., 2011).

Both scenarios were simulated $N_s = 1000$ times. Figures 5 and 6 show a comparison of the results from M3 and M4 for the true target tracks in Figures 4a and 4b, respectively. To test M3's sensitivity to the parameter θ , M4 is compared to M3 using $\theta \in [1, 20]$ deg.

M3 has lower prediction error when the target moves according to a CV model, because M3's CV model for the kinematics is better than M4's CT model for this type of motion. When the target moves according to a CT model, M3 has lower extension error if it holds that $|\theta - \omega_k T| \leq 3$ deg, where T is the sampling time, see Figure 5b. For larger parameter error, M4 is better because it estimates the turn-rate online. As the parameter error grows larger, M3's performance degrades more and more.

When the target maneuvers with variable turn-rate, in terms of position error M4 is significantly better than M3, especially when the turn rate is higher, see Figure 6a. As above, the reason is that a CT model is better than a CV model for this type of motion. In terms of the extension error, M3 performs better than M4 during the time steps that correspond to a small parameter error. However, for the time steps where the parameter error is larger, M4 has significantly better performance than M3, see Figure 6b.

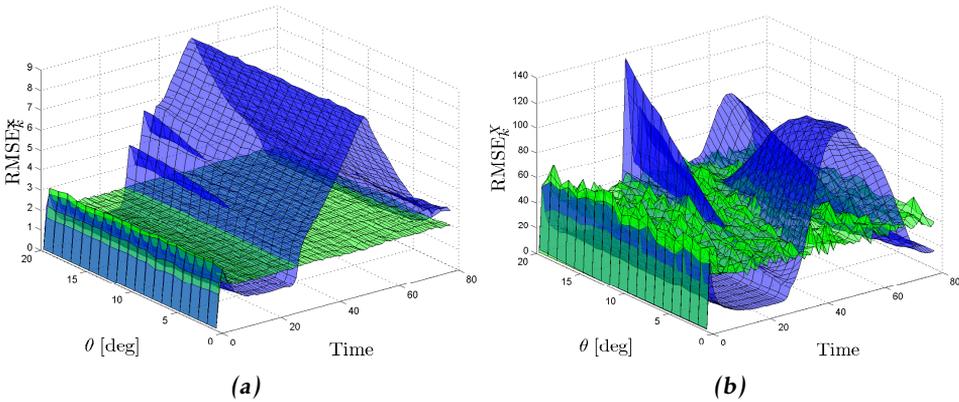


Figure 6: Results for single target tracking for the true track in Fig. 4b, prediction M3 in blue and prediction M4 in green. (a) and (b) show the position and extension RMSE, respectively. From time 25 to time 75 the true turn rate goes from 0 deg to 24 deg, and then back to 0 deg again. For prediction M4 the position and extension RMSE are constant for the whole trajectory, with the exception of small increases immediately after mode changes. For M3 the position and extension RMSE are largest when the true turn rate is highest. During the maneuver, M4 has significantly smaller position and extension RMSEs than M3.

In comparison, M3 and M4 are quite similar in that the extension transition density is a Wishart density that allows for, among other things, rotations of the extension. However, M3 requires a rotation parameter to be set, which can be difficult, especially in the multiple target case where the targets can maneuver with individual time varying turn-rates. The results show that the RMSEs increase when the parameter is set incorrectly.

An important issue to stress here is that the multiple model framework M3 can be used more flexibly than what is proposed originally in (Lan and Rong Li, 2012). A straightforward improvement would be to add additional rotation modes that have different probable θ values. Using a larger discrete set of parameter values, M3 would cover the unknown parameter space more efficiently. With M4, the measurements are used to estimate an individual turn-rate for each target, and in a sense M4 can be considered as a continuous parameter version of M3. However, note that in order to match M4, M3 might require a considerable number of modes, with a corresponding increase in computational demands. In this sense, M4 is more efficient than M3, because it handles a variable turn-rate using a single mode.

A final issue that must be mentioned for a fair comparison of M3 and M4 is that M3 uses Koch's random matrix model (Koch, 2008), while M4 uses Feldmann et al.'s random matrix model (Feldmann et al., 2011). Due to this, it is not possible

to make use of a CT model for M3. The ability to use such a CT model for M3 would enable one to obtain a good turn-rate estimate, which can be substituted into the multiple model framework of M3 to reduce its errors. This is essentially the idea that was used for model M2.

Multiple maneuvering targets

In (Granström and Orguner, 2012a; Granström and Orguner, 2012) a GIW version of the extended target PHD filter (Mahler, 2009) is given. For prediction the standard method M1 is used. In the results section of (Granström and Orguner, 2012a) it is noted that multiple targets that move according to a CV model are easy to track with a CV motion model. However, targets that maneuver according to a CT model, while simultaneously being spatially close, are difficult to track. One problem is the simple CV prediction (7), which is insufficient to describe the target motion during maneuvers (Granström and Orguner, 2012a). A result is that the filter cannot keep spatially close targets resolved during the maneuvers, resulting in cardinality being underestimated.

The presented prediction method M4 was used in the GIW-PHD filter (Granström and Orguner, 2012a), and tested on a scenario with two targets. This scenario was also used in (Granström and Orguner, 2012a), and the true target tracks are shown in Figure 4c. While moving in parallel, the targets' extents were separated by a distance d . In (Granström and Orguner, 2012a) it was shown that the targets needed to be separated by $d \geq 21\text{m}$ for the cardinality to be estimated correctly during the CT maneuver. At closer distance, the GIW-PHD filter could not keep the two targets resolved.

For this paper the scenario was simulated for separation $d = 0, 0.5, 1, \dots, 10$ [m]. For each separation d , the scenario was simulated $N_s = 100$ times. The mean estimated cardinality is shown in Figure 7. From the results we can make two observations. The first is that cardinality is estimated correctly for separation $d \geq 6\text{m}$, which is an improvement over (Granström and Orguner, 2012a) where $d \geq 21\text{m}$ was needed. This performance improvement is a direct result of using a prediction that allows for kinematic state dependent rotations of the extension estimate.

The second observation is that at separation $d \leq 4\text{m}$, the cardinality is underestimated during CV motion (from time 40 to time 75), because the filter cannot keep the targets resolved. This is actually worse performance than (Granström and Orguner, 2012a), where the cardinality was estimated correctly during CV motion at separation $d = 0\text{m}$. The explanation is that the kinematic state motion model that is used in (Granström and Orguner, 2012a) is a better model for CV motion than the CT model used in this paper.

Summary of 2D results

The results show that when the turn rate is known with high accuracy, the prediction methods M2, M3 and M4 perform similarly. However, when the turn rate is uncertain, M4 performs better because it estimates the turn rate directly from

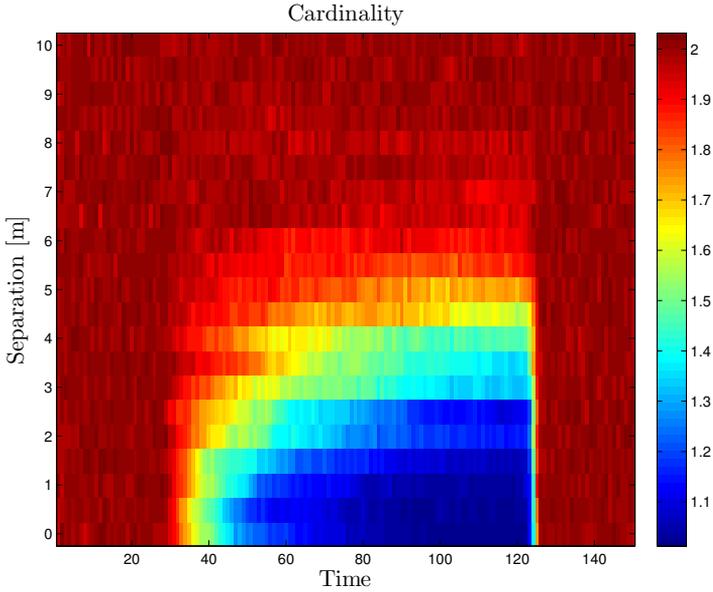


Figure 7: Results for the multiple target scenario in Fig. 4c. At separation distances $d \geq 6\text{m}$ the cardinality is estimated correctly.

the measurement data, and it also considers the effects of turn rate uncertainty on the extension estimate. The scenario with two maneuvering targets shows that including rotation of the extension can significantly improve performance for multiple maneuvering target tracking.

7 Concluding remarks

This paper presented a new prediction update for extended targets whose extensions are modeled as random matrices. The new prediction allows for transformation of the target extension during target maneuvers, and the main tools for deriving the prediction are presented in terms of three different theorems. Two of the theorems show how matrix variate probability density functions can be approximated, the third theorem shows how a conditional matrix variate distribution can be marginalized.

Results from simulating a single prediction step show that the presented prediction method outperforms related work in terms of the expected value, and covariance of the predicted extension. In two simulations with a single maneuvering target, it is shown that the presented prediction method is more general than related work, because it can estimate the turn rate online and does not rely on setting a parameter. In a simulation with two targets it was shown that the presented prediction method can improve performance significantly when the targets maneuver while being spatially close.

In future work, we plan to include the presented prediction method in a multiple model framework. This could include motion modes for, e.g., constant velocity motion, constant turn rate motion with rotation of the extension, and scaling of the extension. It would also be interesting to see how a scaling of the extension matrix could be made dependent on the kinematic state, possibly using a kinematic state that corresponds to the scaling rate. As noted in (Koch, 2008; Feldmann et al., 2011; Lan and Rong Li, 2012), scaling of the extension matrix has important applications for group target tracking.

This work has not considered the measurement update within the random matrix framework, see e.g. (Orguner, 2012; Lan and Rong Li, 2012) for some recent work on this topic. Coupling the extension measurement update to the turn rate could possibly improve estimation of the turn rate. Finally, in this work, we have used the random matrix model of Koch (2008) and Feldmann et al. (2011) which uses inverse Wishart densities to represent the target extents. A drawback of this methodology could be that in higher dimensions, a single parameter might not be sufficient to represent the uncertainty of the extent matrix. Hence, the consideration of more general matrix-variate densities, with many parameters to represent the uncertainty, might be necessary for high dimensions.

A Appendix

A.1 Preliminaries

This appendix gives some known results that are needed for the proofs of the theorems.

Multivariate gamma function

For $a > \frac{d-1}{2}$, the multivariate gamma function, and its logarithm, can be expressed in terms of the ordinary gamma function as (Gupta and Nagar, 2000, Theorem 1.4.1)

$$\Gamma_d(a) = \pi^{d(d-1)} \prod_{i=1}^d \Gamma(a - (i-1)/2), \quad (61a)$$

$$\log \Gamma_d(a) = d(d-1) \log \pi + \sum_{i=1}^d \log \Gamma(a - (i-1)/2). \quad (61b)$$

Multivariate beta function

For $a > \frac{d-1}{2}$ and $b > \frac{d-1}{2}$, the multivariate beta function is expressed in terms of the multivariate gamma function as (Gupta and Nagar, 2000, Theorem 1.4.2)

$$\beta_d(a, b) = \frac{\Gamma_d(a)\Gamma_d(b)}{\Gamma_d(a+b)}. \quad (62)$$

Expected value and covariance of some matrix variate distributions

Let A_{ij} denote the i, j :th element of a matrix A . Let X have pdf $\mathcal{W}(X; n, N)$. The expected value and covariance of X are (Gupta and Nagar, 2000, Theorem 3.3.15)

$$E[X_{ij}] = nN_{ij}, \quad (63a)$$

$$\text{Cov}(X_{ij}, X_{kl}) = n(N_{ik}N_{jl} + N_{il}N_{jk}). \quad (63b)$$

Let X have pdf $\mathcal{IW}(X; v, V)$. The expected value and covariance of X are (Gupta and Nagar, 2000, Theorem 3.4.3)

$$E[X_{ij}] = \frac{V_{ij}}{v - 2d - 2}, \quad v - 2d - 2 > 0, \quad (64a)$$

$$\text{Cov}(X_{ij}, X_{kl}) = \frac{2(v - 2d - 2)^{-1} V_{ij}V_{kl} + V_{ik}V_{jl} + V_{il}V_{jk}}{(v - 2d - 1)(v - 2d - 2)(v - 2d - 4)}, \quad v - 2d - 4 > 0. \quad (64b)$$

Let X have pdf $\mathcal{GB}_d^{II}(X; a, b, \Omega, \Psi)$. If $\Psi = \mathbf{0}_d$, the first and second order moments of X are (Gupta and Nagar, 2000, Theorem 5.3.20)

$$E[X_{ij}] = \frac{2a}{2b - d - 1} \Omega_{ij} \quad (65a)$$

$$E[X_{ij}X_{kl}] = \frac{2a}{(2b - d)(2b - d - 1)(2b - d - 3)} \left[\{2a(2b - d - 2) + 2\} \Omega_{ij}\Omega_{kl} + (2a + 2b - d - 1)(\Omega_{jl}\Omega_{ik} + \Omega_{il}\Omega_{kj}) \right], \quad 2b - d - 3 > 0. \quad (65b)$$

The covariance of X is easily derived using the first and second order moments.

A.2 Expected values of the \mathcal{GB}_d^{II} -distribution

This appendix derives some expected values for the matrix variate generalized beta type-II distribution.

Expected value of the inverse

Let U be matrix variate beta type-II distributed with pdf (Gupta and Nagar, 2000, Definition 5.2.2)

$$p(U) = \mathcal{B}_d^{II}(U; a, b) \quad (66a)$$

$$= \frac{|U|^{a - \frac{d+1}{2}} |\mathbf{I}_d + U|^{-(a+b)}}{\beta_d(a, b)} \quad (66b)$$

where $a > \frac{d-1}{2}$, $b > \frac{d-1}{2}$, and \mathbf{I}_d is a $d \times d$ identity matrix. Then U^{-1} has pdf (Gupta and Nagar, 2000, Theorem 5.3.6)

$$p(U^{-1}) = \mathcal{B}_d^{II}(U^{-1}; b, a). \quad (67)$$

Let $X = \Omega^{1/2}U\Omega^{1/2}$ where $\Omega \in \mathbb{S}_{++}^d$. The pdf of (X) is (Gupta and Nagar, 2000, Theorem 5.2.2)

$$p(X) = \mathcal{GB}_d^{II}(X; a, b, \Omega, \mathbf{0}_d) \tag{68}$$

and subsequently the pdf of $X^{-1} = \Omega^{-1/2}U^{-1}\Omega^{-1/2}$ is

$$p(X^{-1}) = \mathcal{GB}_d^{II}(X^{-1}; b, a, \Omega^{-1}, \mathbf{0}_d) \tag{69}$$

The expected value of X^{-1} is (Gupta and Nagar, 2000, Theorem 5.3.20)

$$E[X^{-1}] = \frac{2b}{2a - d - 1}\Omega^{-1}. \tag{70}$$

Expected value of the log-determinant

Let y be a univariate random variable. The moment generating function of y is defined as

$$\mu_y(s) \triangleq E_y[e^{sy}], \tag{71}$$

and the expected value of y is given in terms of $\mu_y(s)$ as

$$E[y] = \left. \frac{d\mu_y(s)}{ds} \right|_{s=0}. \tag{72}$$

Let $y = \log |X|$, where $p(X) = \mathcal{B}_d^{II}(X; a, b)$. The moment generating function of y is

$$\mu_y(s) = E[|X|^s] = \int |X|^s p(X) dX \tag{73a}$$

$$= \int |X|^s \beta_d^{-1}(a, b) |X|^{a-\frac{1}{2}(d+1)} |\mathbf{I}_d + X|^{-(a+b)} dX \tag{73b}$$

$$= \beta_d^{-1}(a, b) \beta_d(a + s, b - s) \times \int \beta_d^{-1}(a + s, b - s) |X|^{a+s-\frac{1}{2}(d+1)} |\mathbf{I}_d + X|^{-(a+s+b-s)} dX \tag{73c}$$

$$= \frac{\beta_d(a + s, b - s)}{\beta_d(a, b)} \int \mathcal{B}_d^{II}(X; a + S, b - s) dX \tag{73d}$$

$$= \frac{\beta_d(a + s, b - s)}{\beta_d(a, b)} = \frac{\Gamma_d(a + s)\Gamma_d(b - s)}{\Gamma_d(a + s + b - s)} \frac{\Gamma_d(a + b)}{\Gamma_d(a)\Gamma_d(b)} \tag{73e}$$

$$= \frac{\Gamma_d(a + s)\Gamma_d(b - s)}{\Gamma_d(a)\Gamma_d(b)}, \tag{73f}$$

The expected value of y is

$$E[y] = E[\log |X|] \tag{74a}$$

$$= \left. \frac{d(\Gamma_d(a + s)\Gamma_d(b - s))}{ds} \right|_{s=0} \frac{1}{\Gamma_d(a)\Gamma_d(b)} \tag{74b}$$

$$= \left(\frac{d\Gamma_d(a+s)}{ds} + \frac{d\Gamma_d(b-s)}{ds} \right) \Big|_{s=0} \quad (74c)$$

$$= \left(\frac{d \log \Gamma_d(a+s)}{ds} + \frac{d \log \Gamma_d(b-s)}{ds} \right) \Big|_{s=0} \quad (74d)$$

$$= \left(\sum_{i=1}^d \frac{d \log \Gamma(a+s-(i-1)/2)}{ds} + \frac{d \log \Gamma(b-s-(i-1)/2)}{ds} \right) \Big|_{s=0} \quad (74e)$$

$$= \sum_{i=1}^d \psi_0(a-(i-1)/2) - \psi_0(b-(i-1)/2), \quad (74f)$$

where $\psi_0(\cdot)$ is the digamma function, also called the polygamma function of order zero. If $p(Y) = \mathcal{GB}_d^{II}(Y; a, b, \Omega, \mathbf{0}_d)$, then $Z = \Omega^{-1/2}Y\Omega^{-1/2}$ has pdf $\mathcal{B}_d^{II}(Z; a, b)$ (Gupta and Nagar, 2000, Theorem 5.2.2). It then follows that

$$E[\log |Y|] = E[\log |\Omega^{1/2}\Omega^{-1/2}Y\Omega^{-1/2}\Omega^{1/2}|] \quad (75a)$$

$$= E[\log |\Omega^{1/2}| + \log |\Omega^{-1/2}Y\Omega^{-1/2}| + \log |\Omega^{1/2}|] \quad (75b)$$

$$= E[\log |\Omega| + \log |\Omega^{-1/2}Y\Omega^{-1/2}|] \quad (75c)$$

$$= \log |\Omega| + E[\log |Z|] \quad (75d)$$

$$= \log |\Omega| + \sum_{i=1}^d \left[\psi_0(a-(i-1)/2) - \psi_0(b-(i-1)/2) \right]. \quad (75e)$$

A.3 Proof of Theorem 1

The density $q(X)$ is given as

$$q(X) \triangleq \arg \min_{q(X)} \text{KL}(p(X) \| q(X)) \quad (76a)$$

$$= \arg \max_{q(X)} \int p(X) \log(q(X)) dX \quad (76b)$$

$$= \arg \max_{q(X)} \int p(X) \left[-\frac{(v-d-1)d}{2} \log 2 + \frac{v-d-1}{2} \log |V| \right. \\ \left. - \log \Gamma_d\left(\frac{v-d-1}{2}\right) - \frac{v}{2} \log |X| + \text{Tr}\left(-\frac{1}{2}X^{-1}V\right) \right] dX \quad (76c)$$

$$= \arg \max_{q(X)} -\frac{(v-d-1)d}{2} \log 2 + \frac{v-d-1}{2} \log |V| \\ - \log \Gamma_d\left(\frac{v-d-1}{2}\right) - \frac{v}{2} E_p[\log |X|] + \text{Tr}\left(-\frac{1}{2}E_p[X^{-1}]V\right) \quad (76d)$$

$$= \arg \max_{q(X)} f(v, V) \quad (76e)$$

Differentiating the objective function $f(v, V)$ with respect to V gives

$$\frac{df(v, V)}{dV} = \frac{v-d-1}{2}V^{-1} - \frac{1}{2}E_p[X^{-1}]. \quad (77)$$

Setting to zero and solving for V gives

$$V = (v-d-1)E_p[X^{-1}]^{-1} = \frac{(v-d-1)(2a-d-1)}{2b}\Omega \quad (78)$$

where the expected value is calculated based on a result derived in Appendix A.2. Differentiating the objective function with respect to v gives

$$\frac{df(v, V)}{dv} = -\frac{d}{2}\log 2 + \frac{1}{2}\log|V| - \frac{d\log\Gamma_d\left(\frac{v-d-1}{2}\right)}{dv} - \frac{1}{2}E_p[\log|X|] \quad (79a)$$

$$= -\frac{d}{2}\log 2 + \frac{1}{2}\log|V| - \frac{1}{2}\sum_{i=1}^d\psi_0\left(\frac{v-d-i}{2}\right) - \frac{1}{2}E_p[\log|X|]. \quad (79b)$$

Setting the result equal to zero gives

$$0 = \log|V| - d\log 2 - \sum_{i=1}^d\psi_0\left(\frac{v-d-i}{2}\right) - E_p[\log|X|] \quad (80a)$$

$$\begin{aligned} &= \log|V| - d\log 2 - \sum_{i=1}^d\psi_0\left(\frac{v-d-i}{2}\right) - \log|\Omega| \\ &\quad - \sum_{i=1}^d\left[\psi_0\left(a - \frac{1}{2}(i-1)\right) - \psi_0\left(b - \frac{1}{2}(i-1)\right)\right] \end{aligned} \quad (80b)$$

where the expected value of $\log|X|$ is derived in Appendix A.2. Inserting V from (78) gives

$$\begin{aligned} 0 &= \log|\Omega| + d\log\left(\frac{(v-d-1)(2a-d-1)}{2b}\right) - d\log 2 - \sum_{i=1}^d\psi_0\left(\frac{v-d-i}{2}\right) \\ &\quad - \log|\Omega| - \sum_{i=1}^d\left[\psi_0\left(\frac{2a+1-i}{2}\right) - \psi_0\left(\frac{2b+1-i}{2}\right)\right] \end{aligned} \quad (81a)$$

$$\begin{aligned} &= d\log\left(\frac{(v-d-1)(2a-d-1)}{4b}\right) \\ &\quad - \sum_{i=1}^d\left[\psi_0\left(\frac{2a+1-i}{2}\right) - \psi_0\left(\frac{2b+1-i}{2}\right) + \psi_0\left(\frac{v-d-i}{2}\right)\right] \end{aligned} \quad (81b)$$

which is the equation for v in the theorem.

A.4 Proof of Theorem 2

The density $q(\mathbb{V}_x)$ is

$$q(\mathbb{V}_x) = \arg \min_{q(\mathbb{V}_x)} \text{KL}(p(\mathbb{V}_x) \| q(\mathbb{V}_x)) \quad (82a)$$

$$= \arg \max_{q(\mathbb{V}_x)} \int p(\mathbb{V}_x) \log(q(\mathbb{V}_x)) d\mathbb{V}_x \quad (82b)$$

$$= \arg \max_{q(\mathbb{V}_x)} \int p(\mathbb{V}_x) \left[-\frac{sd}{2} \log 2 - \log \Gamma_d\left(\frac{s}{2}\right) - \frac{s}{2} \log |S| + \frac{s-d-1}{2} \log |\mathbb{V}_x| + \text{Tr}\left(-\frac{1}{2} S^{-1} \mathbb{V}_x\right) \right] d\mathbb{V}_x \quad (82c)$$

$$= \arg \max_{q(\mathbb{V}_x)} \int p(\mathbf{x}) \left[-\frac{sd}{2} \log 2 - \log \Gamma_d\left(\frac{s}{2}\right) - \frac{s}{2} \log |S| + \frac{s-d-1}{2} \log |\mathbb{V}_x| + \text{Tr}\left(-\frac{1}{2} S^{-1} \mathbb{V}_x\right) \right] d\mathbf{x} \quad (82d)$$

$$= \arg \max_{q(\mathbb{V}_x)} \mathbb{E}_x \left[-\frac{sd}{2} \log 2 - \log \Gamma_d\left(\frac{s}{2}\right) - \frac{s}{2} \log |S| + \frac{s-d-1}{2} \log |\mathbb{V}_x| + \text{Tr}\left(-\frac{1}{2} S^{-1} \mathbb{V}_x\right) \right] \quad (82e)$$

$$= -\frac{sd}{2} \log 2 - \log \Gamma_d\left(\frac{s}{2}\right) - \frac{s}{2} \log |S| + \frac{s-d-1}{2} \mathbb{E}_x [\log |\mathbb{V}_x|] + \text{Tr}\left(-\frac{1}{2} S^{-1} \mathbb{E}_x [\mathbb{V}_x]\right). \quad (82f)$$

Let $\mathbb{C}_I = \mathbb{E}_x [\log |\mathbb{V}_x|]$ and $\mathbb{C}_{II} = \mathbb{E}_x [\mathbb{V}_x]$. This results in

$$q(\mathbb{V}_x) = \arg \max_{q(\mathbb{V}_x)} -\frac{sd}{2} \log 2 - \log \Gamma_d\left(\frac{s}{2}\right) - \frac{s}{2} \log |S| + \frac{s-d-1}{2} \mathbb{C}_I + \text{Tr}\left(-\frac{1}{2} S^{-1} \mathbb{C}_{II}\right) = \arg \max_{q(\mathbb{V}_x)} f(s, S). \quad (83)$$

Differentiating the objective function $f(s, S)$ with respect to S , setting the result equal to zero and multiplying both sides by 2 gives

$$-sS^{-1} + S^{-1} \mathbb{C}_{II} S^{-1} = 0 \quad \Leftrightarrow \quad S = \frac{1}{s} \mathbb{C}_{II}. \quad (84)$$

Note that the expected value for \mathbb{V}_x under the Wishart distribution $q(\cdot)$ is $sS = \mathbb{C}_{II}$. Thus the expected value under $q(\cdot)$ is correct regardless of the parameter s . Differentiating the objective function $f(s, S)$ in (83) with respect to s gives

$$\frac{df(s, S)}{ds} = -\frac{d}{2} \log 2 - \frac{1}{2} \sum_{i=1}^d \psi_0\left(\frac{s-i+1}{2}\right) - \frac{1}{2} \log |S| + \frac{1}{2} \mathbb{C}_I \quad (85a)$$

$$= \frac{d}{2} \log \frac{s}{2} - \frac{1}{2} \sum_{i=1}^d \psi_0 \left(\frac{s-i+1}{2} \right) - \frac{1}{2} \log |\mathbb{C}_{II}| + \frac{1}{2} \mathbb{C}_I \quad (85b)$$

where we substituted S with (84) to obtain (85b). Equating the result to zero and multiplying both sides by 2 gives (33) in Theorem 2.

A.5 Proof of Theorem 3

We have $p(X)$ given as

$$p(X) = \int p(X|V)p(V)dV \quad (86a)$$

$$= \int \mathcal{I} \mathcal{W} \left(X; v, \frac{V}{\gamma} \right) \mathcal{W} (V; s, S) dV \quad (86b)$$

$$= \int \left\{ 2^{\frac{(v-d-1)d}{2}} \Gamma_d \left(\frac{v-d-1}{2} \right) |X|^{\frac{v}{2}} \right\}^{-1} \left| \frac{V}{\gamma} \right|^{(v-d-1)/2} \text{etr} \left(-0.5 X^{-1} \frac{V}{\gamma} \right) \\ \times \left\{ 2^{\frac{sd}{2}} \Gamma_d \left(\frac{s}{2} \right) |S|^{\frac{s}{2}} \right\}^{-1} |V|^{\frac{s-d-1}{2}} \text{etr}(-0.5 S^{-1} V) dV \quad (86c)$$

$$= \left\{ \Gamma_d \left(\frac{v-d-1}{2} \right) \Gamma_d \left(\frac{s}{2} \right) |X|^{\frac{v}{2}} |S|^{\frac{s}{2}} \right\}^{-1} 2^{-\frac{(v+s-d-1)d}{2}} \gamma^{-\frac{(v-d-1)d}{2}} \\ \times \int |V|^{\frac{v+s-2d-2}{2}} \text{etr} \left(-0.5 \left(\frac{X^{-1}}{\gamma} + S^{-1} \right) V \right) dV \quad (86d)$$

$$= \left\{ \Gamma_d \left(\frac{v-d-1}{2} \right) \Gamma_d \left(\frac{s}{2} \right) |X|^{\frac{v}{2}} |S|^{\frac{s}{2}} \right\}^{-1} 2^{-\frac{(v+s-d-1)d}{2}} \gamma^{-\frac{(v-d-1)d}{2}} \\ \times 2^{\frac{sd}{2}} \Gamma_d \left(\frac{v+s-d-1}{2} \right) \left| \left(\frac{X^{-1}}{\gamma} + S^{-1} \right) \right|^{\frac{v+s-d-1}{2}} \\ \times \int \mathcal{W} \left(V; v+s-d-1, \frac{X^{-1}}{\gamma} + S^{-1} \right) dV \quad (86e)$$

$$= \left\{ \Gamma_d \left(\frac{v-d-1}{2} \right) \Gamma_d \left(\frac{s}{2} \right) |X|^{\frac{v}{2}} |S|^{\frac{s}{2}} \right\}^{-1} 2^{-\frac{(v+s-d-1)d}{2}} \gamma^{-\frac{(v-d-1)d}{2}} \\ \times 2^{\frac{(v+s-d-1)d}{2}} \Gamma_d \left(\frac{v+s-d-1}{2} \right) \left| (\gamma X)^{-1} + S^{-1} \right|^{-\frac{v+s-d-1}{2}} \quad (86f)$$

$$= \left\{ \Gamma_d \left(\frac{v-d-1}{2} \right) \Gamma_d \left(\frac{s}{2} \right) |X|^{\frac{v}{2}} |S|^{\frac{s}{2}} \right\}^{-1} \gamma^{-\frac{(v-d-1)d}{2}} \\ \times \Gamma_d \left(\frac{v+s-d-1}{2} \right) \left| X^{-1} \left(\frac{S}{\gamma} + X \right) S^{-1} \right|^{-\frac{v+s-d-1}{2}} \quad (86g)$$

$$= \frac{\Gamma_d \left(\frac{s+v-d-1}{2} \right)}{\Gamma_d \left(\frac{v-d-1}{2} \right) \Gamma_d \left(\frac{s}{2} \right)} \gamma^{-\frac{(v-d-1)d}{2}} \frac{\left(|X^{-1}| \left| \frac{S}{\gamma} + X \right| |S^{-1}| \right)^{-\frac{v+s-d-1}{2}}}{|X|^{\frac{v}{2}} |S|^{\frac{s}{2}}} \quad (86h)$$

$$= \frac{1}{\beta_d \left(\frac{s}{2}, \frac{v-d-1}{2} \right)} \gamma^{-\frac{(v-d-1)d}{2}} \frac{\left| \frac{S}{\gamma} + X \right|^{-\frac{v+s-d-1}{2}}}{|X|^{\frac{s-d-1}{2}} |S|^{\frac{v-d-1}{2}}} \quad (86i)$$

$$= \frac{|X|^{\frac{s-d-1}{2}} \left| X + \frac{S}{\gamma} \right|^{-\frac{s+v-d-1}{2}}}{\beta_d \left(\frac{s}{2}, \frac{v-d-1}{2} \right) \left| \frac{S}{\gamma} \right|^{\frac{v-d-1}{2}}} \quad (86j)$$

which, by (Gupta and Nagar, 2000, Theorem 5.2.2), is the probability density function for

$$\mathcal{GB}_d^{II} \left(X; \frac{s}{2}, \frac{v-d-1}{2}, \frac{S}{\gamma}, \mathbf{0}_d \right). \quad (87)$$

Bibliography

- Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.
- K. Granström and U. Orguner. A New Prediction Update for Extended Target Tracking with Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 2012b.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström and U. Orguner. Implementation of the GIW-PHD filter. Technical Report LiTH-ISY-R-3046, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, March 2012. URL <http://www.control.isy.liu.se/publications/doc?id=2508>.
- A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics. Chapman & Hall, 2000. ISBN 9781584880462.
- F. Gustafsson, L. Ljung, and M. Millnert. *Signal Processing*. Studentlitteratur, 2010.
- A. Gut. *An Intermediate Course in Probability*. Springer, New York, NY, USA, 1 edition, 1995. ISBN 0-387-94507-5.
- A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, March 1960.

- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- J. Lan and X. Rong Li. Tracking of extended object or target group using random matrix – part I: New model and approach. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2177–2184, Singapore, July 2012.
- F. Lian, C.-Z. Han, W.-F. Liu, X.-X. Yan, and H.-Y. Zhou. Sequential Monte Carlo implementation and state extraction of the group probability hypothesis density filter for partly unresolvable group targets-tracking problem. *IET Radar, Sonar and Navigation*, 4(5):685–702, October 2010.
- C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, January 2001.
- U. Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, July 2012.
- J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.
- X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking: Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, October 2003.
- A. R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, July 2007.
- D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):389–395, December 2003.
- D. Schieferdecker and M. F. Huber. Gaussian Mixture Reduction via Clustering. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Seattle, WA, USA, July 2009.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, second edition, 1993.

- J. L. Williams and P. S. Maybeck. Cost-Function-Based Gaussian Mixture Reduction for Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cairns, Queensland, Australia, July 2003.
- H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 73–78, Chicago, IL, USA, July 2011.

Paper H

On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices

Authors: Karl Granström, and Umut Orguner

Edited version of the paper:

K. Granström and U. Orguner. On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices. *IEEE Transactions on Signal Processing*, 2012e.

On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices

Karl Granström*, and Umut Orguner**

*Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
karl@isy.liu.se

**Dept. of Electrical and Electronics
Engineering
Middle East Technical University
Ankara, Turkey
umut@eee.metu.edu.tr

Abstract

In extended/group target tracking, where the extensions of the targets are estimated, target spawning and combination events might have significant implications on the extensions. This paper investigates target spawning and combination events for the case that the target extensions are modeled in a random matrix framework. The paper proposes functions that should be provided by the tracking filter in such a scenario. The results, which are obtained by a gamma Gaussian inverse Wishart implementation of an extended target probability hypothesis density filter, confirms that the proposed functions improve the performance of the tracking filter for spawning and combination events.

1 Introduction

Multiple target tracking can be defined as the processing of multiple measurements obtained from multiple targets in order to maintain estimates of the targets' current states, see e.g. (Bar-Shalom and Fortmann, 1987). In this context, a point target is defined as a target which is assumed to give rise to at most one measurement per time step, and an extended target is defined as a target that potentially gives rise to more than one measurement per time step. Closely related to extended target is group target, defined as a cluster of point targets which can not be tracked individually, but has to be treated as a single object.

In a target tracking scenario, multiple targets may maneuver such that they become spatially close and cannot be resolved, i.e. they appear at the sensor as one target (sometimes called group target) and must be treated as such. Conversely, the individual targets in a group of unresolved targets may maneuver such that they become resolved, i.e. they no longer have to be treated as a group. In this

paper, we refer to the former as the target combination problem, and to the latter as the target spawning problem.

Target spawning, also referred to as splitting targets, is the event that a new target appears very close to an existing target, or the event that a single target separates into two, or more, targets. Spawning occurs e.g. when a platform launches another platform, or an unresolved group of targets resolve into multiple closely spaced targets, see e.g. (Bar-Shalom, 1992; Mahler, 2007). An interactive multiple model joint probabilistic data association filter for tracking a single point target that spawns one point target is given in (Bar-Shalom, 1992, Chapter 4).

Target combination, also referred to as target merge, is the event that multiple single targets form a group of targets. In certain scenarios target combination can efficiently be seen, and modeled, as target death (when a target disappears from sensor view). In other scenarios it may be computationally efficient to combine resolved single targets into a group, see e.g. (Lau et al., 2010).

While tracking point targets, target spawning and combination events can be handled by additional target births around the main target and spontaneous target deaths, respectively, in the tracking filter. On the other hand, in extended or group target tracking where the target or group size should also be estimated by the tracker, a target spawning event might potentially cause a reduction in the size of the extent of the main target. Likewise, in the case of target combination, the size of the combined target logically can become the sum of the sizes of the individual targets. This interesting phenomenon that can be observed in extended and group target tracking, but not in conventional point target tracking, has to be modeled and taken care of in the tracking filter.

In this paper we consider combination and spawning for extended targets. An extended target's size and shape can be modeled in different ways, see e.g. (Salmond and Parr, 2003; Baum and Hanebeck, 2009; Baum et al., 2010; Granström et al., 2011; Lundquist et al., 2011; Zhu et al., 2011; Baum and Hanebeck, 2011), here we use Koch's random matrix model (Koch, 2008). We limit the discussion to considering combination of two targets, and spawning of one new target, or equivalently splitting into two targets.

To the best of the authors' knowledge, there is no previous work on extended or group target combination, and the only work that mentions extended or group target spawning is (Lian et al., 2010). The work (Lian et al., 2010), which also uses the random matrix model (Koch, 2008), proposes a spawning model that corresponds to a spawned target whose state's expected value is identical to the expected value of the state of the target from which it spawned. This includes the spawned target's extension, which also keeps the expected value of the original target's extension.

This very simple model cannot be expected to be valid in all scenarios, especially not when the original target extension is large and the spawned target's extension is small, which is a quite common case. The spawning model presented in this paper uses a multiple hypothesis structure that considers reasonable alternatives

about the spawned target. The spawning model in (Lian et al., 2010) has a single hypothesis in which the expected kinematic and extension states are equal to the original target. Therefore the model in (Lian et al., 2010) can be considered to be a special case of the presented model.

The rest of the paper is organized as follows. In Section 2 we present the extended target tracking framework, and give a problem formulation. Section 3 contains results on the approximation of probability density functions, in the form of four theorems that will be used in the subsequent parts of the paper. In Sections 4 and 5 we present the proposed combination and spawning methodologies, respectively, for the two target case. A discussion about how the presented methodologies could be used if another extension model is used is presented in Section 6. A simulation study is presented in Section 8, using an example multiple extended target tracking filter which is briefly described in Section 7. The paper is finished with concluding remarks in Section 9.

2 Extended target framework and problem formulation

We use the following notation:

- \mathbb{R}^n is the set of real column vectors of length n , \mathbb{S}_{++}^n is the set of symmetric positive definite $n \times n$ matrices, and \mathbb{S}_+^n is the set of symmetric positive semi-definite $n \times n$ matrices.
- $\mathcal{GAM}(\gamma; \alpha, \beta)$ denotes a gamma probability density function (pdf) defined over the scalar $\gamma > 0$ with scalar shape parameter $\alpha > 0$ and scalar inverse scale parameter $\beta > 0$,

$$\mathcal{GAM}(\gamma; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \gamma^{\alpha-1} e^{-\beta\gamma}, \quad (1)$$

where $\Gamma(\cdot)$ is the gamma function.

- $\mathcal{N}(\mathbf{x}; m, P)$ denotes a multi-variate Gaussian pdf defined over the vector $\mathbf{x} \in \mathbb{R}^{n_x}$ with mean vector $m \in \mathbb{R}^{n_x}$, and covariance matrix $P \in \mathbb{S}_+^{n_x}$,

$$\mathcal{N}(\mathbf{x}; m, P) = \frac{e^{-\frac{1}{2}(\mathbf{x}-m)^\top P^{-1}(\mathbf{x}-m)}}{(2\pi)^{\frac{n_x}{2}} |P|^{\frac{1}{2}}}. \quad (2)$$

where $|\cdot|$ is the matrix determinant function.

- $\mathcal{IW}(X; v, V)$ denotes an inverse Wishart pdf defined over the matrix $X \in \mathbb{S}_{++}^d$ with scalar degrees of freedom $v > 2d$ and parameter matrix $V \in \mathbb{S}_{++}^d$, (Gupta and Nagar, 2000, Definition 3.4.1)

$$\mathcal{IW}(X; v, V) = \frac{2^{-\frac{v-d-1}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |X|^{\frac{v}{2}}} \text{etr}\left(-\frac{1}{2} X^{-1} V\right), \quad (3)$$

where $\text{etr}(\cdot) = \exp(\text{Tr}(\cdot))$ is exponential of the matrix trace, and $\Gamma_d(\cdot)$ is the multivariate gamma function. The multivariate gamma function can be expressed as a product of ordinary gamma functions, see (83) in Appendix A.4.

- $\mathcal{W}(X; w, W)$ denotes a Wishart pdf defined over the matrix $X \in \mathbb{S}_{++}^d$ with scalar degrees of freedom $w \geq d$ and parameter matrix $W \in \mathbb{S}_{++}^d$, (Gupta and Nagar, 2000, Definition 3.2.1)

$$\mathcal{W}(X; w, W) = \frac{2^{-\frac{wd}{2}} |X|^{\frac{w-d-1}{2}}}{\Gamma_d\left(\frac{w}{2}\right) |W|^{\frac{d}{2}}} \text{etr}\left(-\frac{1}{2} W^{-1} X\right). \quad (4)$$

- $\mathcal{BE}(\bar{\gamma}; a, b)$ denotes a beta pdf defined over the scalar $0 < \bar{\gamma} < 1$ with scalar shape parameters $a > 0$ and $b > 0$,

$$\mathcal{BE}(\bar{\gamma}; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \bar{\gamma}^{a-1} (1-\bar{\gamma})^{b-1}. \quad (5)$$

Let ξ_k be the extended target state at time t_k . In this paper we define the extended target state as the combination of a scalar Poisson rate $\gamma_k > 0$, a kinematical state vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and an extension state matrix $X_k \in \mathbb{S}_{++}^d$, i.e. the extended target state is a triple $\xi_k \triangleq (\gamma_k, \mathbf{x}_k, X_k)$. The kinematical state \mathbf{x}_k contains states related to target kinematics, such as position, velocity and heading, while the extension state X_k is a random matrix representing the size and shape of the target. At time t_k , each extended target generates a set of sensor measurements

$$\mathbf{Z}_k = \left\{ \mathbf{z}_k^{(j)} \right\}_{j=1}^{N_{z,k}}, \quad (6)$$

where the measurement noise covariance is related to the extension X_k . In this paper we use the following measurement model from (Koch, 2008),

$$p\left(\mathbf{z}_k^{(j)} \mid \mathbf{x}_k, X_k\right) = \mathcal{N}\left(\mathbf{z}_k^{(j)}; H_k \mathbf{x}_k, X_k\right). \quad (7)$$

The measurement set cardinality $N_{z,k}$ is a random draw from a Poisson distribution whose unknown rate is γ_k .

Let $\mathbf{Z}^k = \{\mathbf{Z}_1, \dots, \mathbf{Z}_k\}$ denote all measurement sets up to and including time t_k . The state estimate, conditioned on \mathbf{Z}^k , is assumed to be gamma Gaussian inverse Wishart (GGIW) distributed,

$$p\left(\xi_k \mid \mathbf{Z}^k\right) = p\left(\gamma_k \mid \mathbf{Z}^k\right) p\left(\mathbf{x}_k \mid X_k, \mathbf{Z}^k\right) p\left(X_k \mid \mathbf{Z}^k\right) \quad (8a)$$

$$= \mathcal{GAM}\left(\gamma_k; \alpha_{k|k}, \beta_{k|k}\right) \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, P_{k|k} \otimes X_k\right) \\ \times \mathcal{IW}\left(X_k; v_{k|k}, V_{k|k}\right) \quad (8b)$$

$$= \mathcal{GGIW}\left(\xi_k; \zeta_{k|k}\right), \quad (8c)$$

where $A \otimes B$ is the Kronecker product between matrices A and B , and $\zeta_{k|k} = (\alpha_{k|k}, \beta_{k|k}, m_{k|k}, P_{k|k}, v_{k|k}, V_{k|k})$ is the set of GGIW density parameters. The Gaussian covariance is $(P_{k|k} \otimes X_k) \in \mathbb{S}_+^{n_x}$, where $P_{k|k} \in \mathbb{S}_+^s$, and we thus have $n_x = ds$ (refer to (Koch, 2008) for further details).

Decomposing the target kinematics and extension into a Gaussian distributed random vector \mathbf{x}_k and an inverse Wishart distributed random matrix X_k was proposed by Koch (Koch, 2008), see also (Feldmann et al., 2011). As in (Granström and Orguner, 2012b), the Poisson rate is modeled as gamma distributed because the gamma distribution is the conjugate prior for the Poisson rate, see e.g. (Gelman et al., 2004).

The model (8) assumes the Poisson rate γ_k to be conditionally independent of \mathbf{x}_k and X_k . In many applications the number of measurements depends on the distance between the sensor and the target, i.e. on the kinematical position, and also depends on the size of the target, i.e. on the size of the extension. This assumption neglects such dependencies, however the probability density over the number of measurements, conditioned on the target kinematics and extension, is unknown in many applications, and we believe that this assumption is valid in many cases. Furthermore, the assumption also facilitates further analysis. Modeling the extension as a random matrix limits the extended targets to be shaped as ellipses, however the ellipse shape is applicable to many real scenarios in which the target and sensor geometry is such that the target measurements resemble a cluster of detections, rather than a geometric structure (or for that matter a single detection). Finally, it is also assumed that multiple targets evolve independently over time, and generate measurements independently. This assumption is typical in multiple target tracking, see e.g. (Bar-Shalom and Fortmann, 1987).

The first problem considered in this paper is two target combination, i.e. finding the GGIW distribution that corresponds to a group of two independent GGIW distributed extended target estimates. The second problem is target spawning, i.e. finding two GGIW distributions that corresponds to either the splitting of a GGIW distributed extended target estimate, or the appearance of a new GGIW distributed extended target estimate next to an existing estimate.

3 Preliminary results on probability density approximations

In this section we present four probability density approximations, that are all needed in the derivation of the main result. The true densities are approximated by analytical minimization of the Kullback-Leibler divergence (KL-DIV) (Kullback and Leibler, 1951), defined for two pdfs $p(x)$ and $q(x)$ as

$$\text{KL}(p(x)||q(x)) = \int p(x) \log(p(x)/q(x)) dx. \quad (9)$$

Note that, when it comes to approximating distributions in a maximum likelihood sense, the KL-DIV is considered the optimal difference measure (Williams and Maybeck, 2003; Runnalls, 2007; Schieferdecker and Huber, 2009).

3.1 Approximating the distribution of functions of gamma distributed random variables

Let γ_1 and γ_2 be two gamma distributed random variables,

$$p(\gamma_1) = \mathcal{GAM}(\gamma_1; \alpha_1, \beta_1), \quad (10a)$$

$$p(\gamma_2) = \mathcal{GAM}(\gamma_2; \alpha_2, \beta_2). \quad (10b)$$

It is here of interest to approximate the distributions over $\gamma = \gamma_1 + \gamma_2$ and $\bar{\gamma}_1 = \frac{\gamma_1}{\gamma_1 + \gamma_2}$. There are some convenient properties for these quantities which we summarize in (66) and (67) in Appendix A.1. However, for these properties to hold, the inverse scale parameters β_1 and β_2 must be equal. We investigate below the general case where β_1 and β_2 need not be equal, i.e. $\beta_1 \neq \beta_2$.

Approximate distribution of γ

Theorem 1. *Let γ_1 and γ_2 be distributed as in (10), and let $p(\gamma)$ be the true distribution of $\gamma = \gamma_1 + \gamma_2$. Let $q(\gamma) = \mathcal{GAM}(\gamma; \alpha, \beta)$ be the gamma distribution, among all gamma distributions, that minimizes the KL-DIV between $p(\gamma)$ and $q(\gamma)$,*

$$q(\gamma) = \arg \min_{q(\cdot) \in \mathcal{GAM}(\cdot)} \text{KL}(p(\gamma) \| q(\gamma)). \quad (11)$$

Then the shape parameter α is the solution to

$$\log(\alpha) - \psi_0(\alpha) + E_p[\log(\gamma)] - \log(E_p[\gamma]) = 0, \quad (12)$$

where $\psi_0(\cdot)$ is the digamma function (a.k.a. the polygamma function of order 0), and the inverse scale parameter β is given by

$$\beta = \frac{\alpha}{E_p[\gamma]}. \quad (13)$$

■

Proof: Given in Appendix A.5. □

Remark: The expressions for the shape parameter (12) and the inverse scale parameter (13) correspond to equating the expected values of $\log(\gamma)$ and γ , respectively, under both distributions,

$$E_p[\log(\gamma)] = E_q[\log(\gamma)], \quad (14a)$$

$$E_p[\gamma] = E_q[\gamma]. \quad (14b)$$

Approximate distribution of $\bar{\gamma}_1$

Theorem 2. Let γ_1 and γ_2 be distributed as in (10), and let $p(\bar{\gamma}_1)$ be the true distribution of $\bar{\gamma}_1 = \frac{\gamma_1}{\gamma_1 + \gamma_2}$. Let $q(\bar{\gamma}_1) = \mathcal{BE}(\bar{\gamma}_1; a, b)$ be the beta distribution, among all beta distributions, that minimizes the KL-DIV between $p(\bar{\gamma}_1)$ and $q(\bar{\gamma}_1)$,

$$q(\bar{\gamma}_1) = \arg \min_{q(\cdot) \in \mathcal{BE}(\cdot)} \text{KL}(p(\bar{\gamma}_1) \| q(\bar{\gamma}_1)). \quad (15)$$

Then the shape parameters a and b are the solution to the system of equations

$$\begin{cases} \psi_0(a+b) - \psi_0(a) + E_p[\log(\bar{\gamma}_1)] = 0 \\ \psi_0(a+b) - \psi_0(b) + E_p[\log(\bar{\gamma}_2)] = 0 \end{cases} \quad (16)$$

where $\bar{\gamma}_2 = \frac{\gamma_2}{\gamma_1 + \gamma_2} = 1 - \bar{\gamma}_1$. ■

Proof: Given in Appendix A.6. □

Remark: The system of equations (16) correspond to equating the expected values of $\log(\bar{\gamma}_1)$ and $\log(1 - \bar{\gamma}_1) = \log(\bar{\gamma}_2)$ under both distributions,

$$E_p[\log(\bar{\gamma}_1)] = E_q[\log(\bar{\gamma}_1)], \quad (17a)$$

$$E_p[\log(\bar{\gamma}_2)] = E_q[\log(\bar{\gamma}_2)]. \quad (17b)$$

3.2 Approximating matrix variate densities

We present below results on how to approximate matrix variate densities with Wishart and inverse-Wishart densities.

Approximation with a \mathcal{W} -distribution

Theorem 3. Let $p(X)$ be a probability density function defined over $X \in \mathbb{S}_{++}^d$. Suppose that $q(X) \triangleq \mathcal{W}(X; v, V)$ is the minimizer of $\text{KL}(p \| q)$ among all Wishart densities. Then V is given as

$$V = \frac{1}{v} E_p[X] \quad (18)$$

and v is the solution to

$$\sum_{i=1}^d \psi_0((v-i+1)/2) + d \log(v/2) - E_p[\log|X|] + \log|E_p[X]| = 0. \quad (19)$$

■

Proof: Given in Appendix A.7. □

Remark: The expressions for the scale matrix V (18) and degrees of freedom v (19) correspond to equating the expected values of X and $\log|X|$ under both

distributions,

$$E_p[X] = E_q[X], \quad (20a)$$

$$E_p[\log |X|] = E_q[\log |X|]. \quad (20b)$$

Approximation with an \mathcal{IW} -distribution

Theorem 4. Let $p(X)$ be a probability density function defined over $X \in \mathbb{S}_{++}^d$. Suppose that $q(X) \triangleq \mathcal{IW}(X; v, V)$ is the minimizer of $\text{KL}(p||q)$ among all inverse Wishart distributions. Then V is given as

$$V = (v - d - 1) \left[E_p(X^{-1}) \right]^{-1} \quad (21)$$

and v is the solution to

$$\sum_{i=1}^d \psi_0((v - d - i)/2) - d \log((v - d - 1)/2) + E_p(\log |X|) + \log |E_p(X^{-1})| = 0. \quad (22)$$

■

Proof: Given in Appendix A.8. □

Remark: The expressions for the inverse scale matrix V (21) and degrees of freedom v (22) correspond to equating the expected values of X^{-1} and $\log |X|$ under both distributions,

$$E_p[X^{-1}] = E_q[X^{-1}], \quad (23a)$$

$$E_p[\log |X|] = E_q[\log |X|]. \quad (23b)$$

3.3 Numerical root-finding

The equations (12), (16), (19), and (22) each have one unique solution, and can be solved using numerical root-finding, see e.g. (Stoer and Bulirsch, 1993, Section 5.1). Examples include Newton-Raphson or modified Newton algorithms, see e.g. (Stoer and Bulirsch, 1993, Section 5.4), for more alternatives see e.g. (Stoer and Bulirsch, 1993, Chapter 5).

4 Target combination

In this section we address the problem of combination of two extended targets, and describe a methodology that should be applied by a random matrix based Bayesian extended target tracking filter in the case of target combination. In Section 4.1 we give a model for extended target combination, and in Section 4.2 we show how the combined distribution can be computed, given the combination model and two extended target estimates. In Section 4.3 we give a criterion that can be used to determine whether or not two extended target estimates should be combined.

4.1 Combination model

The combination of two extended targets $\xi_k^{(1)} = (\gamma_k^{(1)}, \mathbf{x}_k^{(1)}, X_k^{(1)})$ and $\xi_k^{(2)} = (\gamma_k^{(2)}, \mathbf{x}_k^{(2)}, X_k^{(2)})$, yielding independent sets of measurements \mathbf{Z}_1 and \mathbf{Z}_2 , can be seen as the problem of finding the extended target $\xi_k = (\gamma_k, \mathbf{x}_k, X_k)$ that would yield a set of measurements $\mathbf{Z} = \mathbf{Z}_1 \cup \mathbf{Z}_2$, i.e. the union of both measurement sets.

Let $\mathbf{Z}_1 = \{\mathbf{z}_1^{(j)}\}_{j=1}^{n_1}$ and $\mathbf{Z}_2 = \{\mathbf{z}_2^{(j)}\}_{j=1}^{n_2}$ be two sets of measurements, where $\mathbf{z}_i^{(j)} \in \mathbb{R}^d$ for all i, j . The corresponding sample means and sample covariances are given as

$$\bar{\mathbf{z}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{z}_i^{(j)}, \quad (24a)$$

$$Z_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (\mathbf{z}_i^{(j)} - \bar{\mathbf{z}}_i) (\mathbf{z}_i^{(j)} - \bar{\mathbf{z}}_i)^\top, \quad (24b)$$

for $i = 1, 2$, respectively. Straightforward calculations will give the following sample mean and sample covariance for \mathbf{Z} ,

$$\bar{\mathbf{z}} = \frac{n_1}{n_1 + n_2} \bar{\mathbf{z}}_1 + \frac{n_2}{n_1 + n_2} \bar{\mathbf{z}}_2, \quad (25a)$$

$$Z = \frac{n_1}{n_1 + n_2} Z_1 + \frac{n_2}{n_1 + n_2} Z_2 + \frac{n_1 n_2}{(n_1 + n_2)^2} (\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2) (\bar{\mathbf{z}}_1 - \bar{\mathbf{z}}_2)^\top. \quad (25b)$$

Considering that, under the measurement model (7), $\bar{\mathbf{z}}_i$ and Z_i are the maximum likelihood estimates of $H\mathbf{x}^{(i)}$ and $X^{(i)}$, an intuitive two target combination model for the kinematical and extension states can be based on (25) as follows,

$$\mathbf{x}_k = \frac{\gamma_k^{(1)}}{\gamma_k^{(1)} + \gamma_k^{(2)}} \mathbf{x}_k^{(1)} + \frac{\gamma_k^{(2)}}{\gamma_k^{(1)} + \gamma_k^{(2)}} \mathbf{x}_k^{(2)}, \quad (26a)$$

$$X_k = \frac{\gamma_k^{(1)}}{\gamma_k^{(1)} + \gamma_k^{(2)}} X_k^{(1)} + \frac{\gamma_k^{(2)}}{\gamma_k^{(1)} + \gamma_k^{(2)}} X_k^{(2)} + \frac{\gamma_k^{(1)} \gamma_k^{(2)}}{(\gamma_k^{(1)} + \gamma_k^{(2)})^2} H (\mathbf{x}_k^{(1)} - \mathbf{x}_k^{(2)}) (\mathbf{x}_k^{(1)} - \mathbf{x}_k^{(2)})^\top H^\top. \quad (26b)$$

For the Poisson rate, the sum of two Poisson distributed variables with rates $\gamma_k^{(1)}$ and $\gamma_k^{(2)}$ is Poisson distributed with rate $\gamma_k^{(1)} + \gamma_k^{(2)}$. Thus, for the Poisson rate we have the following model,

$$\gamma_k = \gamma_k^{(1)} + \gamma_k^{(2)}. \quad (26c)$$

4.2 Combined distribution for two extended targets

Let the states $\xi_k^{(1)}$ and $\xi_k^{(2)}$ of the two extended targets to be combined be distributed as follows,

$$p\left(\xi_k^{(1)} \mid \mathbf{Z}^k\right) = \mathcal{GGIW}\left(\xi_k^{(1)}; \zeta_{k|k}^{(1)}\right), \quad (27a)$$

$$p\left(\xi_k^{(2)} \mid \mathbf{Z}^k\right) = \mathcal{GGIW}\left(\xi_k^{(2)}; \zeta_{k|k}^{(2)}\right). \quad (27b)$$

We wish to find the parameter $\zeta_{k|k}$ of the distribution

$$p\left(\xi_k \mid \mathbf{Z}^k\right) = \mathcal{GGIW}\left(\gamma_k; \zeta_{k|k}\right), \quad (28)$$

where $\xi_k = (\gamma_k, \mathbf{x}_k, X_k)$ is the state of the combined extended target and is given by the model (26). In what follows, we use the quantities $\bar{\gamma}_k^1$ and $\bar{\gamma}_k^2$ given as

$$\bar{\gamma}_k^{(1)} = \frac{\gamma_k^{(1)}}{\gamma_k^{(1)} + \gamma_k^{(2)}}, \quad (29a)$$

$$\bar{\gamma}_k^{(2)} = \frac{\gamma_k^{(2)}}{\gamma_k^{(1)} + \gamma_k^{(2)}} = 1 - \bar{\gamma}_k^{(1)}, \quad (29b)$$

which are distributed with beta distributions obtained in Theorem 2.

Poisson rate

A gamma distribution for $\gamma_k = \gamma_k^{(1)} + \gamma_k^{(2)}$ is obtained using Theorem 1.

Marginal distribution of kinematical state

Let $m_{k|k}^{(i)}$ and $\hat{P}_{k|k}^{(i)}$ be the mean vector and covariance matrix of the Gaussian marginal distribution of $\mathbf{x}_k^{(i)}$, $i = 1, 2$, see Appendix A.3. Straightforward calculations show that $p\left(\mathbf{x}_k \mid \mathbf{Z}^k\right) = \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, \hat{P}_{k|k}\right)$, where $m_{k|k}$ and $\hat{P}_{k|k}$ are given as

$$m_{k|k} = \mathbb{E}\left[\bar{\gamma}_k^{(1)}\right] m_{k|k}^{(1)} + \mathbb{E}\left[\bar{\gamma}_k^{(2)}\right] m_{k|k}^{(2)}, \quad (30a)$$

$$\hat{P}_{k|k} = \mathbb{E}\left[\left(\bar{\gamma}_k^{(1)}\right)^2\right] \hat{P}_{k|k}^{(1)} + \mathbb{E}\left[\left(\bar{\gamma}_k^{(2)}\right)^2\right] \hat{P}_{k|k}^{(2)}. \quad (30b)$$

The expected values are given in Appendix A.1.

Extension state

Rewrite (26b) as

$$X_k = \bar{\gamma}_k^{(1)} X_k^{(1)} + \bar{\gamma}_k^{(2)} X_k^{(2)} + \bar{\gamma}_k^{(1)} \bar{\gamma}_k^{(2)} X_k^{(12)}, \quad (31)$$

where

$$X_k^{(12)} = H\left(\mathbf{x}_k^{(1)} - \mathbf{x}_k^{(2)}\right)\left(\mathbf{x}_k^{(1)} - \mathbf{x}_k^{(1)}\right)^T H^T, \quad (32)$$

with expected value and covariance given in Appendix A.2 for the marginal distributions of $\mathbf{x}_k^{(i)}$.

Below we are going to find an approximate inverse Wishart density for X_k as follows:

1. Approximate the true density of X_k with a Wishart distribution. This requires the expected values of X_k and $\log|X_k|$. The latter expected value does not have an analytical solution, and must be approximated.
2. Approximate the Wishart distribution with an inverse Wishart distribution. This requires the expected values of X_k^{-1} and $\log|X_k|$, which both have analytical solutions under the Wishart distribution obtained in step 1.

The reason that we do not approximate the true density of X_k with an inverse Wishart density directly is that this would require us to approximate also the expected value of X_k^{-1} . With the two step approach outlined above, only one expected value approximation is needed, which, we have empirically found, gives better results.

Using Theorem 3, the distribution over X_k can be approximated with a Wishart distribution

$$p\left(X_k \mid \mathbf{Z}^k\right) \approx \mathcal{W}\left(X_k; w_{k|k}, W_{k|k}\right). \quad (33)$$

Theorem 3 requires the expected value of $\log|X_k|$, which does not have an analytical solution. It is approximated using a second order Taylor expansion around $E[X_k]$. The required first and second order moments of X_k are

$$E[X_k] = E\left[\bar{\gamma}_k^{(1)}\right] E\left[X_k^{(1)}\right] + E\left[\bar{\gamma}_k^{(2)}\right] E\left[X_k^{(2)}\right] + E\left[\bar{\gamma}_k^{(1)} \bar{\gamma}_k^{(2)}\right] E\left[X_k^{(12)}\right], \quad (34)$$

$$\begin{aligned} E\left[X_{k,ij} X_{k,mn}\right] &= E\left[\bar{\gamma}_1^2\right] \text{Cov}\left(X_k^{(1)}\right)_{ijmn} + E\left[\bar{\gamma}_2^2\right] \text{Cov}\left(X_k^{(2)}\right)_{ijmn} \\ &+ E\left[\bar{\gamma}_k^{(1)} \bar{\gamma}_k^{(2)}\right] \text{Cov}\left(X_k^{(12)}\right)_{ijmn} + E\left[X_{k,ij}\right] E\left[X_{k,mn}\right], \end{aligned} \quad (35)$$

where $X_{k,ij}$ denotes the i, j th element of X_k (i th row and j th column), and $\text{Cov}\left(X_k\right)_{ijmn}$ denotes the covariance between $X_{k,ij}$ and $X_{k,mn}$. Using Theorem 4, the Wishart distribution (33) is approximated with an inverse Wishart distribution,

$$p\left(X_k \mid \mathbf{Z}^k\right) \approx \mathcal{IW}\left(X_k; v_{k|k}, V_{k|k}\right). \quad (36)$$

The required expected values of X_k^{-1} and $\log|X_k|$, under the Wishart distribution (33), are given in Appendix A.4.

Conditional distribution of kinematical state

The conditional distribution for \mathbf{x}_k is

$$p\left(\mathbf{x}_k \mid X_k, \mathbf{Z}^k\right) = \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, P_{k|k} \otimes X_k\right), \quad (37)$$

where $m_{k|k}$ is given in (30). Given $\hat{P}_{k|k}$ in (30), and $v_{k|k}$ and $V_{k|k}$ in (36), $P_{k|k}$ is obtained as the least squares solution to

$$\hat{P}_{k|k} = \frac{P_{k|k} \otimes V_{k|k}}{v_{k|k} + s - sd - 2}. \quad (38)$$

Due to the symmetry of all three matrices, this least squares problem has $s(s+1)/2$ unknown variables and $n_x(n_x + 1)/2 = sd(sd + 1)/2$ equations, thus the problem is overdetermined.

4.3 Target combination criterion

Two extended targets should be combined into one larger target if (and only if) they are located close to each other, and have similar velocity vectors. We decompose this requirement into two separate criteria, one for the spatial closeness, and one for the velocity vectors.

Spatial closeness

Spatial closeness is defined as whether or not the two targets' extensions overlap. Let $\hat{\mathbf{x}}_{k|k}^{(i)} = E[\mathbf{x}_k^{(i)} | \mathbf{Z}^k]$ and $\hat{X}_{k|k}^{(i)} = E[X_k^{(i)} | \mathbf{Z}^k]$. A point $\mathbf{p} \in \mathbb{R}^{n_x}$ lies within $v > 0$ standard deviations of $\hat{\mathbf{x}}_{k|k}^{(i)}$ if the following holds,

$$\left(\mathbf{p} - H\hat{\mathbf{x}}_{k|k}^{(i)} \right)^T \left(v^2 \hat{X}_{k|k}^{(i)} \right)^{-1} \left(\mathbf{p} - H\hat{\mathbf{x}}_{k|k}^{(i)} \right) < 1. \quad (39)$$

Let \mathbf{P}_i be the set of points \mathbf{p} that satisfy (39).

Overlap of the target extensions $X_k^{(i)}$ and $X_k^{(j)}$ is here simplified to whether or not the intersection $\mathbf{P}_{ij} = \mathbf{P}_i \cap \mathbf{P}_j$ is non-empty. This corresponds to the non-existence of a hyperplane that separates the two ellipsoids $\left(\hat{\mathbf{x}}_{k|k}^{(i)}, v^2 \hat{X}_{k|k}^{(i)} \right)$ and $\left(\hat{\mathbf{x}}_{k|k}^{(j)}, v^2 \hat{X}_{k|k}^{(j)} \right)$, which can be posed as a second order cone program (SOCP) feasibility problem, see e.g. (Boyd and Vandenberghe, 2004, Problem 4.25). An SOCP feasibility problem is a type of convex optimization problem, and it can be readily solved using standard MATLAB interfaces such as YALMIP (Löfberg, 2004, 2012) or CVX (Grant and Boyd, 2011, 2008).

Velocity vectors

Two extended targets have similar velocity vectors if the following holds,

$$c_{ij}^v = \left(m_{k|k}^{(i)} - m_{k|k}^{(j)} \right)^T I_v^T \Lambda_{k|k}^{(ij)} I_v \left(m_{k|k}^{(i)} - m_{k|k}^{(j)} \right) < u_v, \quad (40)$$

where $u_v > 0$ is a threshold, $\Lambda_{k|k}^{(ij)} = \left(\hat{P}_{k|k}^{(i)} \right)^{-1} + \left(\hat{P}_{k|k}^{(j)} \right)^{-1}$ and I_v is an $n_x \times n_x$ matrix with identities on the velocity states (all other elements are zero). This is a modified version of a criterion that was used to group single measurement targets (Clark and Godsill, 2007).

Combination criterion

In order to not combine targets that are close but moving in different directions, or combine targets moving at similar velocity in different parts of the surveillance space, two extended targets are combined if (and only if) the following holds,

$$\text{comb} \left(\xi_{k|k}^{(i)}, \xi_{k|k}^{(j)} \right) \triangleq \left(\mathbf{P}_i \cap \mathbf{P}_j \neq \emptyset \right) \& \left(c_{ij}^v < u_v \right) \quad (41)$$

where $\&$ is the logical and operator.

5 Target spawning

This section addresses the problem of extended target spawning and describes a methodology that should be applied by a random matrix based Bayesian extended target tracking filter in the case of target spawning. Here we will only consider two target spawning, and we will assume that the spawning event occurs in between measurement generation, i.e. during the prediction step of extended target tracking filtering.

Because there might be many different spawning pairs which, when combined, would give the same original extended target, we will adopt a multiple hypotheses framework where each hypothesis represents an alternative spawning event.

5.1 Spawning model

Let the target distribution be

$$p \left(\xi_{k-1} | \mathbf{Z}^{k-1} \right) = \mathcal{GGIW} \left(\xi_{k-1}; \zeta_{k-1|k-1} \right). \quad (42)$$

By means of a prediction update, see (Koch, 2008; Feldmann et al., 2011; Granström and Orguner, 2012a), a predicted target distribution

$$p \left(\xi_k | \mathbf{Z}^{k-1} \right) = \mathcal{GGIW} \left(\xi_k; \zeta_{k|k-1} \right), \quad (43)$$

can be obtained. In case a spawning event takes place during the prediction phase, we would instead have two targets

$$p \left(\xi_k^{(1)} | \mathbf{Z}^{k-1} \right) = \mathcal{GGIW} \left(\xi_k^{(1)}; \zeta_{k|k-1}^{(1)} \right), \quad (44a)$$

$$p \left(\xi_k^{(2)} | \mathbf{Z}^{k-1} \right) = \mathcal{GGIW} \left(\xi_k^{(2)}; \zeta_{k|k-1}^{(2)} \right). \quad (44b)$$

Assume that the Poisson rates relate to each other as follows,

$$\gamma_k^{(1)} = \kappa \gamma_k, \quad (45a)$$

$$\gamma_k^{(2)} = (1 - \kappa) \gamma_k, \quad (45b)$$

where $0 < \kappa < 1$. Further, assume that the two spawned targets' extensions relate to each other as follows,

$$X_k^{(1)} = \kappa X_k^{(1/2)}, \quad (46a)$$

$$X_k^{(2)} = (1 - \kappa) X_k^{(1/2)}, \quad (46b)$$

i.e. the extensions have the same shape but different size. The matrix $X_k^{(1/2)} \in \mathbb{S}_{++}^d$ is introduced to simplify the notation below. Note that (45) and (46) can be interpreted as meaning that a larger target (i.e. larger extension) will cause more measurements (i.e. have a higher Poisson rate).

If the two spawned targets (44) were to immediately combine into one target, the resulting combined target is assumed to be equal to the prediction (43). Under this assumption, inserting (45) and (46) into the target combination model (26) gives

$$\mathbf{x}_k = \kappa \mathbf{x}_k^{(1)} + (1 - \kappa) \mathbf{x}_k^{(2)}, \quad (47a)$$

$$X_k = (1 + 2\kappa(\kappa - 1)) X_k^{(1/2)} + \kappa(1 - \kappa) X_k^{(12)}, \quad (47b)$$

$$\gamma_k = \gamma_k^{(1)} + \gamma_k^{(2)}, \quad (47c)$$

where $X_k^{(12)}$ is defined as in (32). For a given κ , (47) is the suggested spawning model.

The assumption that both spawned targets have the same shape, cf. (46), is limiting, however it is necessary because we have two unknown variables, $X_k^{(1)}$ and $X_k^{(2)}$, and only one equation (26b). Furthermore, the assumption is not very critical because it is made in the prediction step, and the subsequent correction step(s) would correct the shapes.

5.2 Spawning hypotheses

Given a prior target distribution (42), the prediction method from (Koch, 2008; Granström and Orguner, 2012b) is used to obtain the predicted target distribution (43). Note that, for a given predicted target distribution (43), there exists an infinite number of spawning pairs (44) whose combination is identical to the predicted single target.

We generate multiple spawning hypotheses as follows. For each κ value, and each dimension ℓ of the extension, one spawned estimate pair is generated, with parameters $\zeta_{k|k-1}^{(1,\ell,\kappa)}$ and $\zeta_{k|k-1}^{(2,\ell,\kappa)}$.

Poisson rates

It follows from the definition of $\mathcal{GAM}(\cdot)$ that $\gamma_k^{(1)}$ and $\gamma_k^{(2)}$ are gamma distributed with parameters

$$\alpha_{k|k-1}^{(i,\ell,\kappa)} = \alpha_{k|k-1} \text{ for } i = 1, 2, \quad (48a)$$

$$\beta_{k|k-1}^{(1,\ell,\kappa)} = \frac{\beta_{k|k-1}}{\kappa}, \quad (48b)$$

$$\beta_{k|k-1}^{(2,\ell,\kappa)} = \frac{\beta_{k|k-1}}{1-\kappa}. \quad (48c)$$

Kinematical states

Let $\hat{X}_{k|k-1} = \mathbb{E}[X_k | \mathbf{Z}^{k-1}]$ under the pdf (43), and let e_ℓ and \mathbf{v}_ℓ be the ℓ :th eigenvalue and eigenvector of $\hat{X}_{k|k-1}$. We set the parameters of the spawned kinematical states to

$$m_{k|k-1}^{(1,\ell,\kappa)} = m_{k|k-1} + (1-\kappa)\sqrt{e_\ell}H^\top \mathbf{v}_\ell, \quad (49a)$$

$$m_{k|k-1}^{(2,\ell,\kappa)} = m_{k|k-1} - \kappa\sqrt{e_\ell}H^\top \mathbf{v}_\ell, \quad (49b)$$

$$P_{k|k-1}^{(i,\ell,\kappa)} = P_{k|k-1} \text{ for } i = 1, 2. \quad (49c)$$

Note that other ways are possible, however, empirically we have found that (49) gives good results.

Extension states

Rewriting (47b), we have

$$X_k^{(1/2)} = \underbrace{\frac{1}{1+2\kappa(\kappa-1)}}_{\triangleq \kappa_1} X_k - \underbrace{\frac{\kappa(1-\kappa)}{1+2\kappa(\kappa-1)}}_{\triangleq \kappa_2} X_k^{(12)}. \quad (50)$$

Similarly to Section 4.2, we first approximate the true distribution over $X_k^{(1/2)}$ with a Wishart distribution, and subsequently approximate the Wishart distribution with an inverse Wishart distribution.

Using Theorem 3 the distribution over $X_k^{(1/2)}$ is approximated with a Wishart distribution

$$p\left(X_k^{(1/2)} | \mathbf{Z}^{k-1}\right) \approx \mathcal{W}\left(X_k^{(1/2)}; w_{k|k-1}^{(\ell,\kappa)}, W_{k|k-1}^{(\ell,\kappa)}\right). \quad (51)$$

This requires the expected value of $\log |X_k^{(1/2)}|$, for which there is no analytical solution. As in Section 4.2, the expected value is approximated using a second order Taylor expansion around $\mathbb{E}[X_k^{(1/2)}]$. The necessary first and second order moments of $X_k^{(1/2)}$ are

$$\mathbb{E}[X_k^{1/2}] = \kappa_1 \mathbb{E}[X_k] - \kappa_2 \mathbb{E}[X_k^{(12)}], \quad (52a)$$

$$\mathbb{E}[X_{k,ij}^{(1/2)} X_{k,mn}^{(1/2)}] = \kappa_1^2 \text{Cov}(X_k)_{ijmn} + \kappa_2^2 \text{Cov}(X_k^{(12)})_{ijmn} + \mathbb{E}[X_{k,ij}^{(1/2)}] \mathbb{E}[X_{k,mn}^{(1/2)}]. \quad (52b)$$

The distribution (51) is subsequently approximated with an inverse Wishart dis-

tribution using Theorem 4,

$$p\left(X_k^{(1/2)} \mid \mathbf{Z}^{k-1}\right) \approx \mathcal{IW}\left(X_k^{(1/2)}; v_{k|k-1}^{(\ell, \kappa)}, V_{k|k-1}^{(\ell, \kappa)}\right). \quad (53)$$

Finally, by (Gupta and Nagar, 2000, Theorems 3.3.11 and 3.4.1) we have

$$p\left(X_k^{(1)} \mid \mathbf{Z}^{k-1}\right) \approx \mathcal{IW}\left(X_k^{(1)}; v_{k|k-1}^{(\ell, \kappa)}, \kappa V_{k|k-1}^{(\ell, \kappa)}\right), \quad (54a)$$

$$p\left(X_k^{(2)} \mid \mathbf{Z}^{k-1}\right) \approx \mathcal{IW}\left(X_k^{(2)}; v_{k|k-1}^{(\ell, \kappa)}, (1 - \kappa)V_{k|k-1}^{(\ell, \kappa)}\right). \quad (54b)$$

Summary

To summarize, for each dimension ℓ of the extension and each κ value, a spawned estimate pair is generated with the following parameters

$$\zeta_{k|k-1}^{(1, \ell, \kappa)} = \left(\alpha_{k|k-1}, \frac{\beta_{k|k-1}}{\kappa}, m_{k|k-1}^{(1, \ell, \kappa)}, p_{k|k-1}^{(1, \ell, \kappa)}, v_{k|k-1}^{(\ell, \kappa)}, \kappa V_{k|k-1}^{(\ell, \kappa)}\right), \quad (55a)$$

$$\zeta_{k|k-1}^{(2, \ell, \kappa)} = \left(\alpha_{k|k-1}, \frac{\beta_{k|k-1}}{1 - \kappa}, m_{k|k-1}^{(2, \ell, \kappa)}, p_{k|k-1}^{(2, \ell, \kappa)}, v_{k|k-1}^{(\ell, \kappa)}, (1 - \kappa)V_{k|k-1}^{(\ell, \kappa)}\right). \quad (55b)$$

If a set \mathbf{K} of K different κ values are used, in total dK spawned estimate pairs are generated, or $2dK$ GGIW components.

6 On the use of other spatial distributions

By using positive definite matrices to represent the target extensions our work implicitly assumes that the target extent is an ellipsoid. Moreover, the spatial distribution of the measurements in our work is a Gaussian density. One potential extension of the presented work is thus to relax the Gaussian and/or the ellipsoidal assumption. This would allow different types of spatial distributions for the target measurements, see e.g. (Gilholm et al., 2005).

The methodology presented here gives hints on what type of approach can be used in a general setting, e.g. when parametric densities from the exponential family are used. The proposed combination model is based on representing the set of measurements, generated by the individual target's spatial densities, with a single spatial density of the same functional form as those of the individual targets. With a different parametric spatial density, one would need to write the formulae for the combined density parameters in terms of the formulae that connect the parameters of the spatial density of each target to the corresponding measurements. This is what is performed in (24) and (25).

The multiple hypothesis methodology for spawning could also be useful if other spatial distributions are used. In this work a single ellipsoid is simply divided into alternative possible ellipsoids, if other distributions from the exponential family are used, similar division methods must be devised. If the spatial distribution is multi-modal, the different modes of the spatial density might provide intuitive alternative divisions. Note that, in the spawning case and without using the subsequent measurements, one can never arrive at a unique solution for how

a single target can be divided into multiple targets. Therefore, an uncertainty margin must always be left for the forthcoming measurements to resolve.

7 Multiple target tracking framework

To demonstrate the merits of the presented methodologies for target combination and target spawning, the methodologies must be integrated into a multiple extended target tracking framework. In this section we will briefly describe the framework that we have worked in, we show how combination and spawning fits into the framework, and we also discuss target extraction and performance metrics.

7.1 The GGIW-PHD filter

We have used a modified version of the Gaussian inverse Wishart (GIW) implementation (Granström and Orguner, 2012a; Granström and Orguner, 2012) of the extended target probability hypothesis density (PHD) filter (Mahler, 2009). In the GIWPHD filter the extended target state is composed only of the kinematical and extension states (i.e. there are no Poisson rates), and the PHD of the target set is approximated as a mixture of GIW densities as follows (Granström and Orguner, 2012a)

$$D_{k|k}(\xi_k) \approx \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k) \mathcal{IW}(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}), \quad (56)$$

where $J_{k|k}$ is the number of mixture components, and the scalars $w_{k|k}^{(j)} > 0$ are the components weights.

In the modified version of the GIWPHD filter that we use in the current work, called the GGIW-PHD filter, the extended target state also includes the Poisson rates. The PHD of the target set is approximated as a mixture of GGIW densities as follows

$$D_{k|k}(\xi_k) \approx \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{GAM}(\gamma_k; \alpha_{k|k}, \beta_{k|k}) \mathcal{N}(\mathbf{x}_k; m_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k) \\ \times \mathcal{IW}(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}) \quad (57a)$$

$$= \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{GGIW}(\xi_k; \zeta_{k|k}^{(j)}) \quad (57b)$$

In both PHD filter implementations, the parameters of the PHDs are predicted and updated recursively with the measurements. For details on the implementations, please refer to (Granström and Orguner, 2012a; Granström and Orguner, 2012; Granström and Orguner, 2012b).

7.2 Combination in the GGIW-PHD filter

Target combination in the GGIW-PHD filter is performed after the correction step (measurement update). An algorithm for target combination is given in Table 1. In the algorithm, all GGIW components with a weight less than 0.5 are left unaltered. The components with weight larger than 0.5 are checked for combination in a pairwise manner, starting with the highest weights. Note that any component is combined with at most one other component.

Table 1: GGIW-PHD filter target combination

1:	require: Combination criterion parameters ν and u_ν , and PHD intensity
	$D_{k k}(\xi_k) = \sum_{j=1}^{\tilde{J}_{k k}} w_{k k}^{(j)} \mathcal{GGIW}(\xi_k; \zeta_{k k}^{(j)}).$
2:	initialize: $I = \left\{ i \mid w_{k k}^{(i)} \geq 0.5 \right\}$,
3:	$\tilde{D}_{k k}(\xi_k) = \sum_{j \in I} w_{k k}^{(j)} \mathcal{GGIW}(\xi_k; \zeta_{k k}^{(j)}),$
4:	$\ell = I^c .$
5:	repeat
6:	$\ell = \ell + 1$
7:	$j = \arg \max_{i \in I} w_{k k}^{(i)}$
8:	$I_j = \left\{ i \in I \setminus j \mid \text{comb}(\xi_{k k}^{(i)}, \xi_{k k}^{(j)}) = 1 \right\}$
9:	if $I_j \neq \emptyset$ then
10:	$n = \arg \max_{i \in I_j} w_{k k}^{(i)}$
11:	Combine components j and n as presented in Section 4.2, let $\tilde{\zeta}_{k k}^{(\ell)}$ denote the corresponding GGIW distribution parameters.
12:	$\tilde{w}_{k k}^{(\ell)} = \mathbb{E} \left[\tilde{\gamma}_k^{(j)} \right] w_{k k}^{(j)} + \mathbb{E} \left[\tilde{\gamma}_k^{(n)} \right] w_{k k}^{(j)}$
13:	$I = I \setminus \{j, n\}$
14:	else
15:	$\tilde{\zeta}_{k k}^{(\ell)} = \zeta_{k k}^{(j)}$
16:	$\tilde{w}_{k k}^{(\ell)} = w_{k k}^{(j)}$
17:	$I = I \setminus j$
18:	end if
19:	until $I = \emptyset$
20:	output: Combined PHD intensity, where $\tilde{J}_{k k} \leq J_{k k}$,
	$\tilde{D}_{k k}(\xi_k) = \sum_{j=1}^{\tilde{J}_{k k}} \tilde{w}_{k k}^{(j)} \mathcal{GGIW}(\xi_k; \tilde{\zeta}_{k k}^{(j)}).$

7.3 Spawning in the GGIW-PHD filter

Generation of spawning estimate pairs in the GGIW-PHD filter is performed in the prediction step (time update). An algorithm for target prediction with spawning is given in Table 2. The spawning weight parameter $w_{\text{sp}} > 0$ can be understood as follows. If the PHD has $\hat{N}_{x,k}$ GGIW components, all with weight ≈ 1 , the total sum of weights for the spawning components is approximately

$$\hat{N}_{x,k} \times 2dK \times w_{\text{sp}} = N_{\text{sp}}. \quad (58)$$

The quantity N_{sp} approximates the mean number of spawned targets. Thus, the more likely spawning events are thought to be, the larger the spawning weight parameter should be set.

In the algorithm, for each component with weight greater than 0.5, K additional component pairs (which have negligible weights compared to the corresponding component, because typically $w_{\text{sp}} \ll 1$) are added to the predicted PHD. These added components correspond to a heuristic modification of the extended target PHD filter to include spawning hypotheses. The procedure of adding component pairs is analogous to the Gaussian Mixture PHD-filter for point targets (Vo and Ma, 2006), in which a single spawned Gaussian component is added for each existing component.

7.4 Performance Evaluation

Let the true target set at time t_k be

$$\mathbf{X}_k = \left\{ \xi_k^{(i)} \right\}_{i=1}^{N_{x,k}}, \quad (59)$$

where the true target cardinality $N_{x,k}$, and each true target state $\xi_k^{(i)}$, are unknown. Estimates of the target states $\hat{\xi}_{k|k}^{(j)}$ are obtained by extracting the GGIW components whose weights are larger than or equal to a threshold, e.g. 0.5, see (Vo and Ma, 2006). Let the set of extracted targets be denoted

$$\hat{\mathbf{X}}_{k|k} = \left\{ \hat{\xi}_{k|k}^{(i)} \right\}_{i=1}^{\hat{N}_{x,k}}, \quad (60a)$$

$$\hat{\xi}_{k|k}^{(i)} = \left(\hat{\gamma}_{k|k}^{(i)}, \hat{\mathbf{x}}_{k|k}^{(i)}, \hat{X}_{k|k}^{(i)} \right), \quad (60b)$$

$$\hat{\gamma}_{k|k}^{(i)} = \mathbb{E}[\gamma_k], \quad (60c)$$

$$\hat{\mathbf{x}}_{k|k}^{(i)} = \mathbb{E}[\mathbf{x}_k], \quad (60d)$$

$$\hat{X}_{k|k}^{(i)} = \mathbb{E}[X_k], \quad (60e)$$

where the expected values are taken with respect to the i :th GGIW distribution.

An assignment $\bar{\pi}$ between the true target states $\xi_k^{(j)}$ and the extracted states $\hat{\xi}_{k|k}^{(i)}$ is computed using the optimal sub-pattern assignment (OSPA) metric (Schuhmacher et al., 2008).

Table 2: GGIW-PHD filter prediction with spawning

1: **require:** Spawning weight w_{sp} , set \mathbf{K} of κ values, and PHD intensity

$$D_{k|k}(\xi_k) = \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{GGIW}(\xi_k; \zeta_{k|k}^{(j)}).$$

2: **initialize:** $J_{\text{aux}} = J_{k|k}$

3: **for** $j = 1, \dots, J_{k|k}$ **do**

4: Predict j :th component as outlined in (Granström and Orguner, 2012a; Granström and Orguner, 2012).

5: **if** $w_{k|k}^{(j)} > 0.5$ **then**

6: **for** $\kappa \in \mathbf{K}$ **do**

7: **for** $\ell = 1, \dots, d$ **do**

8: Compute $\zeta_{k|k-1}^{(1,\ell,\kappa)}$ and $\zeta_{k|k-1}^{(2,\ell,\kappa)}$ as presented in Section 5.2.

9: For $i = 1, 2$, set

$$\begin{aligned} w_{k+1|k}^{(J_{\text{aux}}+i)} &= w_{\text{sp}} w_{k|k}^{(j)}, \\ \zeta_{k+1|k}^{(J_{\text{aux}}+i)} &= \zeta_{k+1|k}^{(i,\ell,\kappa)}. \end{aligned}$$

10: $J_{\text{aux}} = J_{\text{aux}} + 2$

11: **end for**

12: **end for**

13: **end if**

14: **end for**

15: **output:** Predicted PHD intensity with spawned estimate pairs, where $J_{k+1|k} \geq J_{k|k}$,

$$D_{k+1|k}(\xi_{k+1}) = \sum_{j=1}^{J_{k+1|k}} w_{k+1|k}^{(j)} \mathcal{GGIW}(\xi_{k+1}; \zeta_{k+1|k}^{(j)}),$$

The tracking results are evaluated in terms of the following quantities,

$$d^{(\gamma)} = \sum_j \left| \gamma_k^{(j)} - \hat{\gamma}_{k|k}^{(\bar{\pi}(j))} \right|, \quad (61a)$$

$$d^{(\mathbf{x})} = \sum_j \left\| \mathbf{x}_k^{(j)} - \hat{\mathbf{x}}_{k|k}^{(\bar{\pi}(j))} \right\|_2, \quad (61b)$$

$$d^{(X)} = \sum_j \left\| X_k^{(j)} - \hat{X}_{k|k}^{(\bar{\pi}(j))} \right\|_F, \quad (61c)$$

where $|\cdot|$ is the absolute value, $\|\cdot\|_2$ is the Euclidean norm, and $\|\cdot\|_F$ is the Frobenius norm. An estimate of the target cardinality is given by the sum of weights (Mahler, 2007), $\hat{N}_{k|k} = \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$.

8 Simulation study

This section presents a simulation study conducted for testing the proposed target combination and spawning functions.

8.1 Multiple target tracking setup

Four scenarios were simulated. The kinematical state contains $2D$ position, velocity and acceleration, the extension is two dimensional (i.e. $d = 2$, $n_x = 6$ and thus $s = 3$). In each scenario, the i :th target's true extension is

$$X_k^{(i)} = R_k^{(i)} \text{diag}([\bar{a}_i^2 \ a_i^2]) \left(R_k^{(i)}\right)^T, \quad (62)$$

where \bar{a}_i and a_i are the major and minor axes, and $R_k^{(i)}$ is a rotation matrix applied such that either \bar{a}_i or a_i is aligned with the direction of motion. The motion model used in the filter is described in detail in (Koch, 2008), as in (Granström and Orguner, 2012a) the motion model parameters were set to $t_s = 1\text{s}$, $\theta = 1\text{s}$, $\Sigma = 0.1\text{m/s}^2$ and $\tau = 5\text{s}$.

The true target motions were not generated using a specific motion model. This choice may seem simplistic, however the main focus of this paper is not on motion modeling, but on spawning and combination. The generated true tracks are sufficiently realistic to test the presented spawning and combination functions.

In each scenario a Poisson distributed number of clutter measurements were distributed uniformly in the surveillance space, with Poisson rate 10 per scan.

8.2 True target tracks

Target combination

In the first scenario two targets maneuver such that they move in parallel and give rise to unresolved sets of measurements, see the true tracks in Figure 1a. The scenario is meant to simulate a real world scenario such as a radar tracking two airplanes that begin to fly in a close formation. It has 24 time steps, starting at time step 12 the targets move in parallel at equal speeds, with their 2σ ellipses touching (this corresponds to $v = 2$ in (39)). True target measurements were generated with $\gamma_k^{(i)} = 20$, $\bar{a}_i = 10$ and $a_i = 5$ for $i = 1, 2$.

Target split

In the second scenario an extended target splits in half into two smaller extended targets, see the true tracks in Figure 1b. The scenario is meant to simulate a real world scenario such as a radar tracking two airplanes flying in close formation before separating. It has 15 time steps, the spawning occurs between time steps 5 and 6. True target measurements were generated with $\gamma_k = 40$, $\bar{a} = 10$ and $a = 10$ before spawning, and $\gamma_k^{(i)} = 20$, $\bar{a}_i = 10$ and $a_i = 5$, for $i = 1, 2$, after spawning.

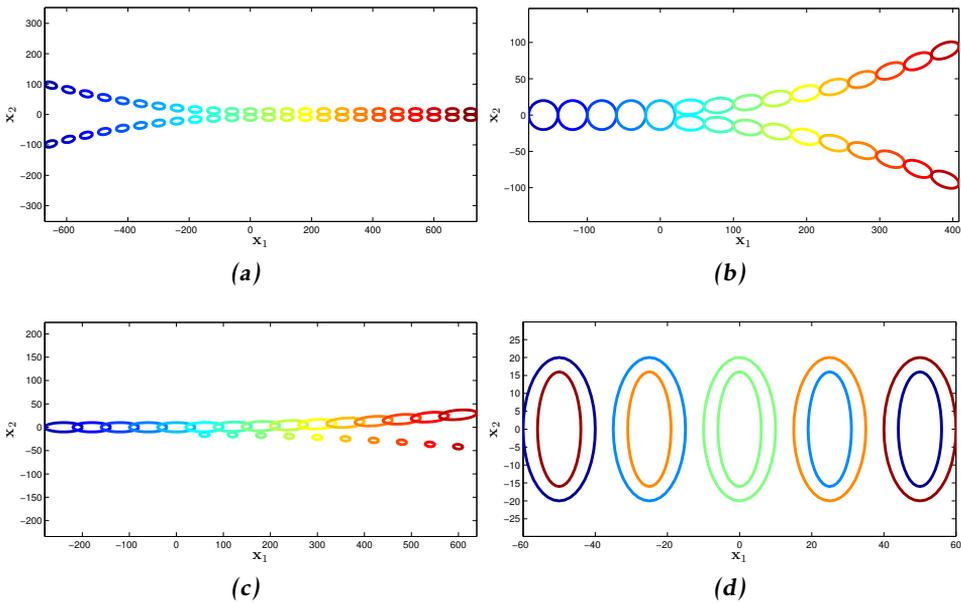


Figure 1: True tracks for the simulation scenarios. Colors are used to show time steps, dark blue and dark red are the first and last time steps. (a) (b), (c) and (d) show the true target positions for target combination, target split, new target appearance, and target occlusion, respectively. For the target occlusion scenario, only every 25:th time step is shown for increased clarity.

New target appearance

In the third scenario a new smaller target appears next to an existing target, see the true tracks in Figure 1c. The scenario has 15 time steps, the spawning occurs between time steps 5 and 6. This scenario is meant to simulate a real world scenario such as a radar tracking an airplane that launches a weapon. True target measurements were generated with $\gamma_k^{(1)} = 40$, $\bar{a}_1 = 20$ and $a_1 = 5$ for the larger target, and $\gamma_k^{(2)} = 10$, $\bar{a}_2 = 6.67$ and $a_2 = 1$ for the smaller spawned target.

Target occlusion

In the fourth scenario two targets of different size move in opposite direction towards each other, and as the targets pass each other the smaller target is occluded by the larger target, i.e. it is not seen by the sensor and thus does not produce any measurements. The scenario has 101 time steps, and the true kinematic positions were generated such that $\mathbf{x}_{51}^{(1)} = \mathbf{x}_{51}^{(2)}$, i.e. the targets are at the same position at the 51:st time step. The respective initial positions vary with the simulated constant speed $c^{(i)}$ of the targets. In Figure 1d the true tracks are shown for $c^{(i)} = 1$. Only every 25:th time step is shown for increased clarity.

The spawning event occurs when the smaller target becomes visible to the sensor again. Because this happens gradually, it is not possible to give a definitive time for when the spawning happens. The scenario is meant to simulate a real world scenario such as a camera that is used to track two persons moving across the field of view, in opposite directions, and at different distances from the sensor. For the detections in the image plane, this would appear as two different sized targets that move “through” each other.

True target measurements were generated with $\gamma_k^{(1)} = 30$, $\bar{a}_1 = 10$ and $a_1 = 5$ for the larger target, and $\gamma_k^{(2)} = 15$, $\bar{a}_2 = 8$ and $a_2 = 3$ for the smaller target. At each time step measurements were simulated for both targets, however for the second target the measurements that fell inside the 3σ ellipse of the first target were removed to simulate the occlusion.

8.3 Combination results

For the spatial closeness criterion we set $v = 2$, and for the velocity vectors we set $u_v = 50$. The results are shown in Figure 2. When the targets are sufficiently close, moving in the same direction, they are combined into just one target. For $v = 2$, the true targets fulfill the combination criterion between time steps $k = 12$ and $k = 24$. Over 10^3 Monte Carlo simulations, for 60% of the cases the two target estimates are combined at time step $k = 13$, i.e. with a delay of one time step. The delay is typically caused by the fact that the velocity vector estimates must converge to similar values first.

8.4 Spawning results

Three GGIW-PHD filters were run in parallel: one filter with spawning hypotheses computed using the model presented in Section 5 (denoted F1), one filter without spawning model (denoted F2), and one filter with a single spawning hypothesis as in (Lian et al., 2010) (denoted F3). Neither filter used the target combination outlined in Section 4. In F1 and F3 the spawning weight was $w_{sp} = 0.05$. In F1 spawning hypotheses were generated for

$$\kappa \in \mathbf{K} = \left\{ \frac{1}{4}, \frac{1}{2}, \frac{3}{4} \right\}. \quad (63)$$

The parameters of F3 were set such that the expected value was constant for the extended target state, and the variance was increased. The variance of the measurement rate was increased by 50%, a matrix $\text{diag}([1, 0, 0])$ was added to the kinematical state covariance, and the degrees of freedom of the extension state was decreased by 25. These parameters were chosen such that the best possible performance was obtained.

Target split and new target appearance

The second and third scenarios were simulated 10^3 times each, Figures 3 and 4 show the results. The mean sum of weights, and the performance metrics (61), are shown for different distances between the kinematical positions. When the

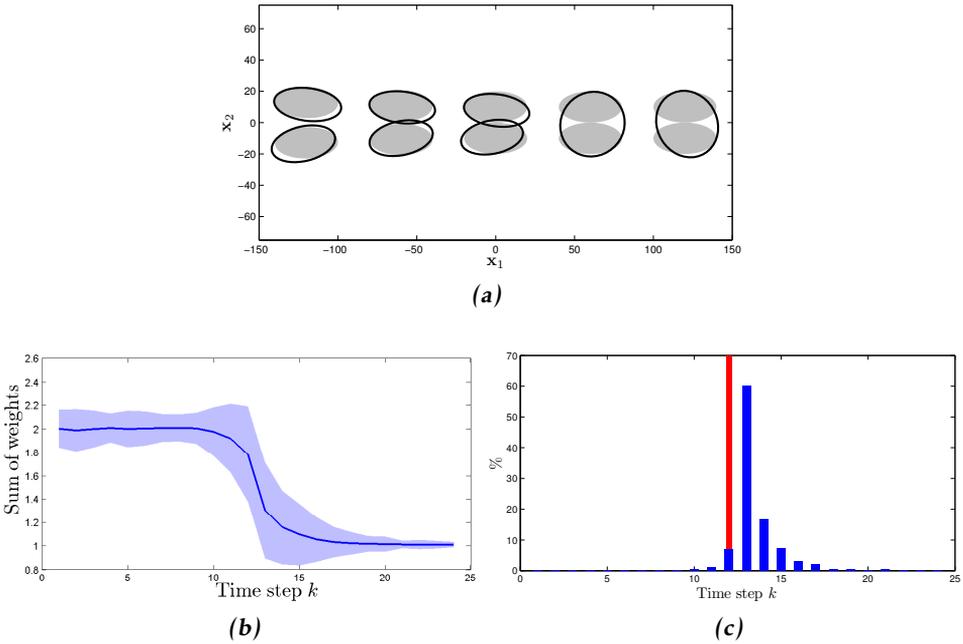


Figure 2: Combination of two targets. (a) Example result from a single simulation run. The true targets are shown as the light gray filled ellipses, the target estimates are shown as black ellipses. When the targets are sufficiently close, and have similar velocity vectors, they are combined into one target. (b) Sum of weights (i.e. estimated cardinality), averaged over 10^3 runs, shown in blue. Mean \pm one standard deviation is shown in light blue. (c) Histogram showing for which time step the two targets were combined. The two targets move in parallel starting at time step 12 (red line), in 60% of the 10^3 simulations the targets estimates were combined after measurement updating in time step 13.

extended targets are still very close, no filter is able to detect the spawning event. However, when the targets start to separate, F1 detects the event at a shorter distance, or equivalently at an earlier time step, than F3. The worst performance is obtained with F2, i.e. the filter without any spawning model.

There is also a significant difference between the three filters with respect to the performance metrics (61), with F1 clearly having the best performance. After the spawning event is detected by F1 and F3, the measurement rate and kinematical state starts to converge towards the correct value. The extension state has a small positive error, however this is expected. As the two targets turn away from each other, their corresponding extensions rotate, and the simple extension prediction used, see (Koch, 2008), does not account for rotations. As noted in previous work (Granström and Orguner, 2012a), during maneuvers the extension estimation error is always larger than during straight line motion.

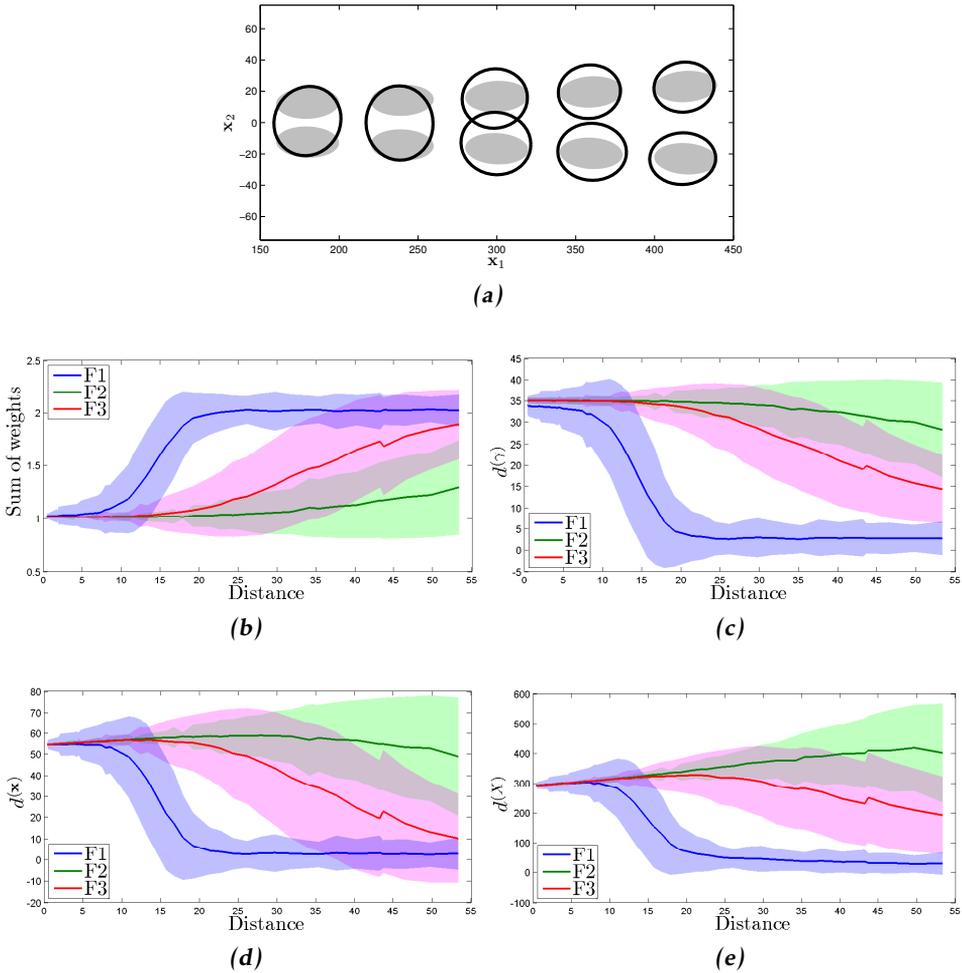


Figure 3: Spawning results for the true tracks in Figure 1b. (a) Example result from a single simulation run. The true targets are shown as the light gray filled ellipses, the target estimates are shown as black ellipses. (b) Estimated cardinality (true cardinality is two). (c) Measurement rate estimation error. (d) Kinematic state estimation error. (e) Extension state estimation error. The results in (b) to (e) are averaged over 10^3 Monte Carlo runs, and are shown for different separation distances. While the GGIW-PHD with spawning can detect the spawning events, adding spawned estimate pairs allows the filter to detect the spawning at a closer distance.

Target occlusion

The fourth scenario was simulated with different target speeds,

$$c^{(i)} = [0.5, 0.51, \dots, 1.0], \quad i = 1, 2. \quad (64)$$

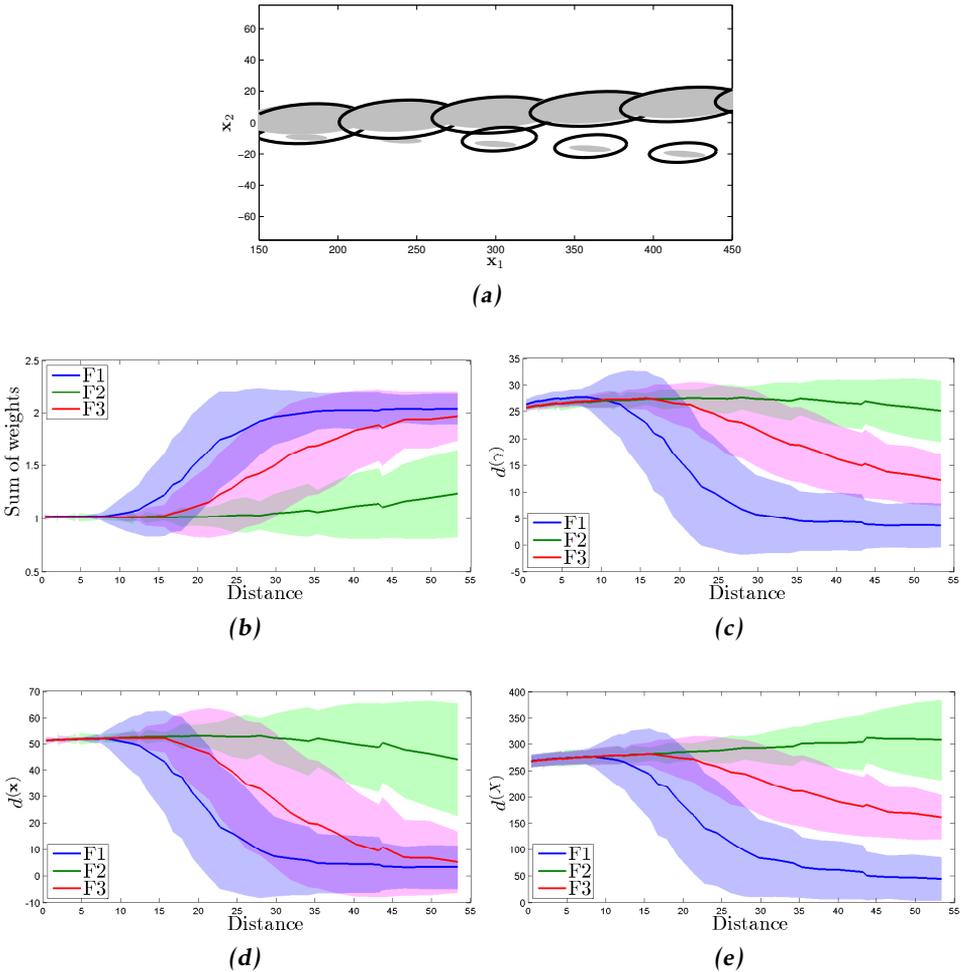


Figure 4: Spawning results for the true tracks in Figure 1c. (a) Example result from a single simulation run. The true targets are shown as the light gray filled ellipses, the target estimates are shown as black ellipses. (b) Estimated cardinality (true cardinality is two). (c) Measurement rate estimation error. (d) Kinematic state estimation error. (e) Extension state estimation error. The results in (b) to (e) are averaged over 10^3 Monte Carlo runs, and are shown for different separation distances. While the GGIW-PHD with spawning can detect the spawning events, adding spawned estimate pairs allows the filter to detect the spawning at a closer distance.

For each speed, the scenario was simulated 10^2 times. The mean estimated cardinalities of all three filters are shown for different target speeds and target distances in Figures 5a, 5b, and 5c, respectively. Figure 5d illustrates the contour plots for Figures 5a, 5b, and 5c, superposed onto each other.

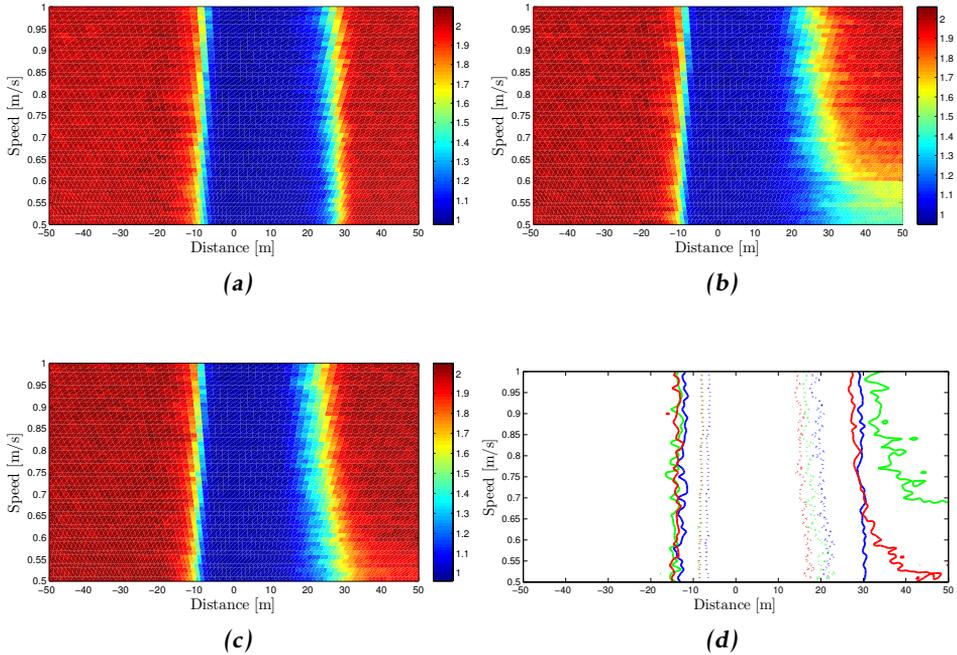


Figure 5: Spawning results for the true tracks in Figure 1d. (a), (b) and (c) show the mean estimated cardinality for F1, F2 and F3, respectively. Dashed lines corresponds to cardinality 1.1, solid lines correspond to cardinality 1.9. (d) Contour plot of the estimated cardinality for F1 (blue), F2 (green), and F3 (red), respectively. Distance is computed as $x_k^{(1)} - x_k^{(2)}$, i.e. the difference in x -position, explaining why there are negative distances.

As the two targets approach each other, all three filters can track both targets until the point where the targets' respective 1σ ellipses are touching. After this point, all three filters estimate cardinality to one target, which is expected. As the targets move away from each other, F1 correctly estimates the cardinality as two around a point which corresponds to when the 4σ ellipses of the targets are touching, regardless of the target speed $c_k^{(i)}$. The filter F2 corrects the cardinality estimate at a much later point, especially at lower speeds, and the performance of F3 is in between F1 and F2.

This strange dependence of the spawning performance on the target speeds observed in F2, and to a lesser degree also in F3, deserves an explanation. When the second target is occluded, all three filters estimate a single target. Hence when the targets start to separate after the occlusion event, F2 predicts and expects a single target in the next sampling instant. On the other hand, F1 and F3 also expect a single target with large probability, however with small probability, F1

and F3 also expect two targets thanks to the spawning hypotheses their PHDs contains. As the targets separate further, one of the spawning hypotheses gains weight and eventually dominates the single target hypothesis easily when the targets are sufficiently separated. This happens earlier for F1 than F3.

The filter F2 always expects a single target. For obtaining the correct cardinality, it has to initiate/give birth to a new extended target. When the targets move/separate fast, the size of the single extended target predicted by the filter cannot catch up with the swiftly enlarged size of the measurement cluster (due to the target separation). Since the predicted target size remains small while the size of the measurement cluster becomes large, a new target is initialized/born easily under this estimate-measurement mismatch. Hence F2 can compensate the lack of spawning hypotheses by initiating a new extended target when the targets separate fast.

However, when the speeds of the targets are small (i.e. when the targets separate slowly), the predicted target size can easily match the overall measurement cluster size, and the incentive to initiate a new target is greatly reduced. Only when the targets are very far can F2 realize that a single elliptical target extent is too poor an explanation for the separated measurement clusters, and initiate a new target.

Hence when the targets separate slowly, new target initiation in F2 is delayed too much, and the lack of the spawning hypotheses becomes really critical.

Summary

To summarize, it is possible for the GGIW-PHD filter to detect spawned targets when spawning hypotheses are not used, however it becomes increasingly difficult as the separation speed decreases. The GGIW-PHD filter with spawning hypotheses detects the spawned targets at the same distance, independent of the separation speed.

Further, used in the GGIW-PHD filter and run on the scenarios in this paper, the presented spawning method clearly outperforms the spawning method presented in (Lian et al., 2010).

8.5 Cycle times

Adding spawning hypotheses increases the number of GGIW components in the filter, and as a consequence the computational complexity increases. Conversely, using the combination functionality decreases the complexity. Mean cycle times for the scenarios in Figures 1a and 1b are given in Tables 3 and 4, respectively. As expected, the mean cycle time increases when spawning hypotheses are used, and it decreases when target combination is used. Note that one should not compare the cycle times for the filter without spawning and the filter without combination, because, while the filters are identically implemented, they are run on different scenarios.

Table 3: Cycle times [s] for the scenario in Figure 1a

Filter	Mean	Median	St.dev.
w comb	0.11	0.06	0.17
w/o comb	0.25	0.12	0.34

Table 4: Cycle times [s] for the scenario in Figure 1b

Filter	Mean	Median	St.dev.
F1	0.82	0.66	0.73
F2	0.11	0.09	0.07
F3	0.14	0.12	0.09

9 Concluding remarks

This paper presented models for combination and spawning of extended targets modeled with random matrices. These models were then used in order to propose functions for multiple extended target tracking filters similar to those used in multiple point target tracking filters. Results show that with an appropriate combination criterion, two extended targets can be combined into one larger target when they are spatially close, and moving in the same direction, while at the same time taking care of their extensions. For spawning, the results show that by including spawning hypotheses the spawning events can be detected earlier than the case when the spawning hypotheses are not used. The results also show that the presented extended target spawning method outperforms earlier work on the topic.

The simulation study clearly shows that adding spawning hypotheses enables earlier detection of spawned targets, however this comes at the price of increased complexity. In the present implementation, spawning hypotheses are added in each time step for GGIW components with weights $w > 0.5$. As an alternative, the measurement sets could be used to determine when it is appropriate to add spawning hypotheses.

The analysis in the paper is limited to the two target case. The results can be directly applicable to combination and spawning events with more than two targets, if the combination/spawning involves two (groups of) targets at a time. The analysis of target combination can be generalized to more than two targets combining at the same time with a considerable amount of work. The more challenging scenarios where more than two (groups of) targets are spawned from an extended target is left as an interesting topic of future work. The combination and spawning functions could also be tested on experimental data, e.g. from a laser range sensor, a radar sensor, or a camera.

A Appendix

A.1 Properties of Gamma distributed random variables

Let γ_1 and γ_2 be independent and Gamma distributed with equal inverse scale parameters,

$$p(\gamma_1) = \mathcal{GAM}(\gamma_1; \alpha_1, \beta), \quad (65a)$$

$$p(\gamma_2) = \mathcal{GAM}(\gamma_2; \alpha_2, \beta). \quad (65b)$$

Then $\gamma = \gamma_1 + \gamma_2$ is Gamma distributed (Jambunathan, 1954)

$$p(\gamma) = \mathcal{GAM}(\gamma; \alpha_1 + \alpha_2, \beta) \quad (66)$$

and $\bar{\gamma}_1 = \frac{\gamma_1}{\gamma_1 + \gamma_2}$ is Beta distributed (Jambunathan, 1954)

$$p(\bar{\gamma}_1) = \mathcal{BE}(\bar{\gamma}_1; \alpha_1, \alpha_2). \quad (67)$$

Let $\bar{\gamma}_2 = \frac{\gamma_2}{\gamma_1 + \gamma_2} = 1 - \bar{\gamma}_1$. It follows immediately from the definition of the beta distribution that $\bar{\gamma}_2$ is beta distributed,

$$p(\bar{\gamma}_2) = \mathcal{BE}(\bar{\gamma}_2; \alpha_2, \alpha_1). \quad (68)$$

The first and second order moments of $\bar{\gamma}_1$ are

$$E[\bar{\gamma}_1] = \frac{\alpha_1}{\alpha_1 + \alpha_2}, \quad (69a)$$

$$E[\bar{\gamma}_1^2] = \frac{\alpha_1(\alpha_1 + 1)}{(\alpha_1 + \alpha_2)(\alpha_1 + \alpha_2 + 1)}, \quad (69b)$$

and consequently the expected value of $\bar{\gamma}_1 \bar{\gamma}_2 = \bar{\gamma}_1(1 - \bar{\gamma}_1)$ is straightforward to compute.

A.2 Matrix product of sum of Gaussians

Let $\mathbf{x}_1 \in \mathbb{R}^{n_x}$ and $\mathbf{x}_2 \in \mathbb{R}^{n_x}$ be two independent Gaussian distributed random vectors with mean vectors $m_1 \in \mathbb{R}^{n_x}$ and $m_2 \in \mathbb{R}^{n_x}$ and covariance matrices $P_1 \in \mathbb{S}_+^{n_x}$ and $P_2 \in \mathbb{S}_+^{n_x}$, and let H be a $d \times n_x$ matrix. Then the quantity $\mathbf{x}_{12} = H(\mathbf{x}_1 - \mathbf{x}_2) \in \mathbb{R}^d$ is Gaussian distributed,

$$p(\mathbf{x}_{12}) = \mathcal{N}(\mathbf{x}_{12}; m_{12}, P_{12}), \quad (70a)$$

$$m_{12} = H(m_1 - m_2), \quad (70b)$$

$$P_{12} = H(P_1 + P_2)H^T. \quad (70c)$$

Let $M_{12} = m_{12}m_{12}^T$. The expected value and covariance of the $d \times d$ matrix $X_{12} = \mathbf{x}_{12}\mathbf{x}_{12}^T$ are given by (Gupta and Nagar, 2000)

$$E[X_{12}]_{ij} = P_{12,ij} + M_{12,ij}, \quad (71a)$$

$$\begin{aligned} \text{Cov}(X_{12})_{ijkl} = & P_{12,ik}P_{12,jl} + P_{12,il}P_{12,jk} + P_{12,jl}M_{12,ik} \\ & + P_{12,il}M_{12,jk} + P_{12,jk}M_{12,il} + P_{12,ik}M_{12,jl}, \end{aligned} \quad (71b)$$

where $E[X_{12}]_{ij}$ is the expected value of the i, j :th element of X_{12} , and $\text{Cov}(X_{12})_{ijkl}$ is the covariance of the i, j :th and k, l :th elements of X_{12} . The expected value (71a) is derived using the first and second order moments of \mathbf{x}_{12} , deriving the covariance (71b) requires tedious calculations involving the first to fourth order moments of \mathbf{x}_{12} , see (Gupta and Nagar, 2000).

A.3 Marginal distribution of kinematical state

The marginal distribution $p(\mathbf{x}_k | \mathbf{Z}^k)$ is a multivariate student-t distribution (Koch, 2008; Feldmann and Koch, 2012), with expected value and covariance (Koch, 2008)

$$E[\mathbf{x}_k] = m_{k|k}, \quad (72a)$$

$$\text{Cov}(\mathbf{x}_k) = \frac{P_{k|k} \otimes V_{k|k}}{v_{k|k} + s - sd - 2} \triangleq \hat{P}_{k|k}, \quad (72b)$$

for $v_{k|k} > sd + 2 - s$. The multivariate student-t distribution can be approximated with a multivariate Gaussian distribution by analytical minimization of the KL-DIV. This gives the following marginal distribution,

$$p(\mathbf{x}_k | \mathbf{Z}^k) \approx \mathcal{N}(\mathbf{x}_k; m_{k|k}, \hat{P}_{k|k}). \quad (73)$$

A.4 Expected values

Gamma distributed random variables

Let γ_1 and γ_2 be independent and gamma distributed

$$p(\gamma_1) = \mathcal{GAM}(\gamma_1; \alpha_1, \beta_1), \quad (74a)$$

$$p(\gamma_2) = \mathcal{GAM}(\gamma_2; \alpha_2, \beta_2), \quad (74b)$$

with $\beta_1 \neq \beta_2$. The expected value of $\gamma = \gamma_1 + \gamma_2$ is

$$E[\gamma] = E[\gamma_1 + \gamma_2] \quad (75a)$$

$$= E[\gamma_1] + E[\gamma_2] \quad (75b)$$

$$= \frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2}. \quad (75c)$$

Let $\bar{\gamma}_1 = \frac{\gamma_1}{\gamma_1 + \gamma_2}$. The expected value of $\log \bar{\gamma}_1$ can be rewritten as

$$E[\log \bar{\gamma}_1] = E[\log \gamma_1] - E[\log(\gamma_1 + \gamma_2)] \quad (76a)$$

$$= \psi_0(\alpha_1) - \log(\beta_1) - E[\log(\gamma_1 + \gamma_2)]. \quad (76b)$$

There is no analytical solution to $E[\log(\gamma_1 + \gamma_2)]$, however it can be computed after Taylor expanding the function $\log(\gamma_1 + \gamma_2)$ around the point $\gamma_1^0 = E[\gamma_1]$ and $\gamma_2^0 = E[\gamma_2]$, which gives

$$E[\log(\gamma_1 + \gamma_2)] \approx \log\left(\frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2}\right) - \frac{1}{2} \frac{\frac{\alpha_1}{\beta_1^2} + \frac{\alpha_2}{\beta_2^2}}{\left(\frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2}\right)^2}. \quad (77)$$

Inverse random matrix – inverse Wishart

Let X be inverse Wishart distributed $p(X) = \mathcal{IW}(X; v, V)$. Then X^{-1} is Wishart distributed $p(X^{-1}) = \mathcal{W}(X^{-1}; v - d - 1, V^{-1})$ (Gupta and Nagar, 2000, Theorem 3.4.1). The expected value of X^{-1} is (Gupta and Nagar, 2000, Theorem 3.3.15)

$$E[X^{-1}] = (v - d - 1) V^{-1}. \quad (78)$$

Inverse random matrix – Wishart

Let X be Wishart distributed $p(X) = \mathcal{W}(X; v, V)$. Then X^{-1} is inverse Wishart distributed $p(X^{-1}) = \mathcal{IW}(X^{-1}; v + d + 1, V^{-1})$ (Gupta and Nagar, 2000, Theorem 3.4.1). The expected value of X^{-1} is (Gupta and Nagar, 2000, Theorem 3.4.3)

$$E[X^{-1}] = \frac{V^{-1}}{(v - d - 1)}. \quad (79)$$

Log-determinant of random matrix – inverse Wishart

Let y be a uni-variate random variable. The moment generating function for y is defined as

$$\mu_y(s) \triangleq E[e^{sy}], \quad (80)$$

and the expected value of y is given in terms of $\mu_y(s)$ as

$$E[y] = \left. \frac{d\mu_y(s)}{ds} \right|_{s=0}. \quad (81)$$

Let $y = \log |X|$, and $p(X) = \mathcal{IW}(X; v, V)$. The moment generating function of y is

$$\mu_y(s) = E[|X|^s] = \int |X|^s p(X) dX \quad (82a)$$

$$= \int |X|^s \frac{2^{-\frac{(v-d-1)d}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |X|^{\frac{v}{2}}} \text{etr}\left(-\frac{1}{2} X^{-1} V\right) dX \quad (82b)$$

$$= \int \frac{2^{-\frac{(v-d-1)d}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |X|^{\frac{v-2s}{2}}} \text{etr}\left(-\frac{1}{2} X^{-1} V\right) dX \quad (82c)$$

$$= \frac{\Gamma_d\left(\frac{v-2s-d-1}{2}\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \int \mathcal{IW}(X; v - 2s, V) dX \quad (82d)$$

$$= \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s. \quad (82e)$$

By (Gupta and Nagar, 2000, Theorem 1.4.1), $\Gamma_d(\cdot)$ and its logarithm is

$$\Gamma_d(a) = \pi^{\frac{1}{4}d(d-1)} \prod_{i=1}^d \Gamma\left(a - \frac{i-1}{2}\right), \tag{83a}$$

$$\log \Gamma_d(a) = \frac{1}{4}d(d-1) \log \pi + \sum_{i=1}^d \log \Gamma\left(a - \frac{i-1}{2}\right). \tag{83b}$$

The expected value of y is

$$E[y] = E[\log |X|] \tag{84a}$$

$$= \frac{d}{ds} \left(\frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \right) \Bigg|_{s=0} \tag{84b}$$

$$= \left(\frac{|V|}{2^d}\right)^s \frac{d}{ds} \Gamma_d\left(\frac{v-d-1}{2} - s\right) \Bigg|_{s=0} + \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \frac{d}{ds} \left(\frac{|V|}{2^d}\right)^s \Bigg|_{s=0} \tag{84c}$$

$$= \left(\frac{|V|}{2^d}\right)^s \frac{d}{ds} \log \Gamma_d\left(\frac{v-d-1}{2} - s\right) \Bigg|_{s=0} + \frac{\Gamma_d\left(\frac{v-d-1}{2} - s\right)}{\Gamma_d\left(\frac{v-d-1}{2}\right)} \left(\frac{|V|}{2^d}\right)^s \log\left(\frac{|V|}{2^d}\right) \Bigg|_{s=0} \tag{84d}$$

$$= - \sum_{j=1}^d \psi_0\left(\frac{v-d-1}{2} - \frac{j-1}{2}\right) + \log\left(\frac{|V|}{2^d}\right) \tag{84e}$$

$$= \log |V| - d \log 2 - \sum_{j=1}^d \psi_0\left(\frac{v-d-j}{2}\right). \tag{84f}$$

Log-determinant of random matrix – Wishart

Let $y = \log |X|$, and $p(X) = \mathcal{W}(X; v, V)$. The moment generating function of y is

$$\mu_y(s) = E[|X|^s] = \int |X|^s p(X) dX \tag{85a}$$

$$= \int |X|^s \frac{2^{-\frac{nd}{2}} |X|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v}{2}\right) |V|^{\frac{v}{2}}} \text{etr}\left(-\frac{1}{2} V^{-1} X\right) dX \tag{85b}$$

$$= \int \frac{2^{-\frac{nd}{2}} |X|^{\frac{v+2s-d-1}{2}}}{\Gamma_d\left(\frac{v}{2}\right) |V|^{\frac{v}{2}}} \text{etr}\left(-\frac{1}{2} V^{-1} X\right) dX \tag{85c}$$

$$= \frac{\Gamma_d\left(\frac{v+2s}{2}\right)}{\Gamma_d\left(\frac{v}{2}\right)} (2^d |V|)^s \int \mathcal{W}(X; v+2s, V) dX \tag{85d}$$

$$= \frac{\Gamma_d\left(\frac{v}{2} + s\right)}{\Gamma_d\left(\frac{v}{2}\right)} (2^d |V|)^s, \tag{85e}$$

and the expected value of y is

$$E[y] = E[\log |X|] \quad (86a)$$

$$= \frac{d}{ds} \left(\frac{\Gamma_d\left(\frac{v}{2} + s\right)}{\Gamma_d\left(\frac{v}{2}\right)} (2^d |V|)^s \right) \Bigg|_{s=0} \quad (86b)$$

$$= (2^d |V|)^s \frac{d}{ds} \frac{\Gamma_d\left(\frac{v}{2} + s\right)}{\Gamma_d\left(\frac{v}{2}\right)} \Bigg|_{s=0} + \frac{\Gamma_d\left(\frac{v}{2} + s\right)}{\Gamma_d\left(\frac{v}{2}\right)} \frac{d}{ds} (2^d |V|)^s \Bigg|_{s=0} \quad (86c)$$

$$= (2^d |V|)^s \frac{d}{ds} \log \Gamma_d\left(\frac{v}{2} + s\right) \Bigg|_{s=0} + \frac{\Gamma_d\left(\frac{v}{2} + s\right)}{\Gamma_d\left(\frac{v}{2}\right)} (2^d |V|)^s \log(2^d |V|) \Bigg|_{s=0} \quad (86d)$$

$$= \sum_{j=1}^d \psi_0\left(\frac{v}{2} - \frac{j-1}{2}\right) + \log(2^d |V|) \quad (86e)$$

$$= \log |V| + d \log 2 + \sum_{j=1}^d \psi_0\left(\frac{v-j+1}{2}\right). \quad (86f)$$

A.5 Proof of Theorem 1

Proof: We have $q(\cdot)$ given by

$$q(\gamma) \triangleq \arg \min_q \text{KL}(p||q) \quad (87a)$$

$$= \arg \max_q \int p(\gamma) \log q(\gamma) d\gamma \quad (87b)$$

$$= \arg \max_q \int p(\gamma) [\alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log(\gamma) - \beta \gamma] d\gamma \quad (87c)$$

$$= \arg \max_q (\alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) E_p[\log(\gamma)] - \beta E_p[\gamma]). \quad (87d)$$

Differentiating the objective function with respect to β , setting the result equal to zero, and solving for β , gives

$$\beta = \frac{\alpha}{E_p[\gamma]}. \quad (88)$$

Differentiating the objective function with respect to α , setting the result equal to zero, and inserting β given in (88), gives

$$\log(\alpha) - \psi_0(\alpha) + E_p[\log(\gamma)] - \log(E_p[\gamma]) = 0. \quad (89)$$

□

A.6 Proof of Theorem 2

Proof: We have $q(\cdot)$ given by

$$q(\gamma) \triangleq \arg \min_q \text{KL}(p||q) \quad (90a)$$

$$= \arg \max_q \int p(\bar{\gamma}_1) \log q(\bar{\gamma}_1) d\bar{\gamma}_1 \quad (90b)$$

$$= \arg \max_q \int p(\bar{\gamma}_1) \left[\log \Gamma(a+b) - \log \Gamma(a) - \log \Gamma(b) \right. \\ \left. + (a-1) \log(\bar{\gamma}_1) + (b-1) \log(1-\bar{\gamma}_1) \right] d\bar{\gamma}_1 \quad (90c)$$

$$= \arg \max_q \left(\log \Gamma(a+b) - \log \Gamma(a) - \log \Gamma(b) \right. \\ \left. + (a-1) E[\log(\bar{\gamma}_1)] + (b-1) E[\log(1-\bar{\gamma}_1)] \right). \quad (90d)$$

Differentiating the objective function with respect to a , and setting the result equal to zero gives

$$\psi_0(a+b) - \psi_0(a) + E[\log(\bar{\gamma}_1)] = 0. \quad (91)$$

Differentiating the objective function with respect to b , and setting the result equal to zero gives

$$\psi_0(a+b) - \psi_0(b) + E[\log(\bar{\gamma}_2)] = 0, \quad (92)$$

where $\bar{\gamma}_2 = \frac{\gamma_2}{\gamma_1 + \gamma_2} = 1 - \bar{\gamma}_1$. \square

A.7 Proof of Theorem 3

Proof: We have $q(\cdot)$ given as

$$q(X) \triangleq \arg \min_q \text{KL}(p||q) \quad (93a)$$

$$= \arg \max_q \int p(X) \log(q(X)) dX \quad (93b)$$

$$= \arg \max_q \left[\frac{1}{2}(v-d-1) E_p[\log|X|] - \frac{1}{2} \text{Tr}(V^{-1} E_p[X]) \right. \\ \left. - \frac{1}{2} v d \log(2) - \Gamma_d(v/2) - \frac{1}{2} v \log|V| \right] \quad (93c)$$

Taking the derivative of the objective function with respect to V , equating the result to zero, and solving for V , we get

$$V = \frac{1}{v} E_p[X] \quad (94)$$

Now, we take the derivative of the objective function with respect to v , equate the result to zero, and insert the V in (94), to obtain

$$\sum_{i=1}^d \psi_0((v-i+1)/2) + d \log(v/2) - E_p[\log|X|] + \log|E_p[X]| = 0. \quad (95)$$

□

A.8 Proof of Theorem 4

Proof: We have $q(\cdot)$ given as

$$q(X) \triangleq \arg \min_q \text{KL}(p||q) \quad (96a)$$

$$= \arg \max_q \int p(X) \log(q(X)) dX \quad (96b)$$

$$= \arg \max_q \left[\frac{1}{2}(v-d-1) \log|V| - \frac{1}{2} \text{Tr}(V E_p(X^{-1})) - \frac{1}{2}(v-d-1)d \log(2) \right. \\ \left. - \log \Gamma_d((v-d-1)/2) - \frac{1}{2}v E_p(\log|X|) \right] \quad (96c)$$

Taking the derivative of the objective function with respect to V , equating the result to zero, and solving for V , we get

$$V = (v-d-1) [E_p(X^{-1})]^{-1} \quad (97)$$

Now, we take the derivative of the objective function with respect to v , equate the result to zero, and insert the V in (97), to obtain

$$- \sum_{i=1}^d \psi_0((v-d-i)/2) + d \log((v-d-1)/2) - E_p(\log|X|) - \log|E_p(X^{-1})| = 0. \quad (98)$$

□

Bibliography

- Y. Bar-Shalom. *Multitarget-multisensor tracking: applications and advances*, volume II of *Multitarget-multisensor Tracking: Applications and Advances*. Artech House, 1992.
- Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Ajman, United Arab Emirates, December 2009.
- M. Baum and U. D. Hanebeck. Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 338–345, Chicago, IL, USA, July 2011.
- M. Baum, B. Noack, and U. D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Edinburgh, UK, July 2010.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- D.E. Clark and S. Godsill. Group target tracking with the gaussian mixture probability hypothesis density filter. In *International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 149–154, Melbourne, Australia, December 2007.
- M. Feldmann and W. Koch. Comments on “Bayesian Approach to Extended Object and Cluster Tracking using Random Matrices”. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1687–1693, April 2012.
- M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.
- A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2004.
- K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.
- K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 2012a. doi: 10.1109/TSP.2012.2212888.

- K. Granström and U. Orguner. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 2170–2176, Singapore, July 2012b.
- K. Granström and U. Orguner. On Spawning and Combination of Extended/Group Targets Modeled with Random Matrices. *IEEE Transactions on Signal Processing*, 2012c.
- K. Granström, C. Lundquist, and U. Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 592–599, Chicago, IL, USA, July 2011.
- K. Granström and U. Orguner. Implementation of the GIW-PHD filter. Technical Report LiTH-ISY-R-3046, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, March 2012. URL <http://www.control.isy.liu.se/publications/doc?id=2508>.
- M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.
- M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `.../.../cvx`, April 2011.
- A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman & Hall/CRC monographs and surveys in pure and applied mathematics. Chapman & Hall, 2000. ISBN 9781584880462.
- M. V. Jambunathan. On information and sufficiency. *The Annals of Mathematical Statistics*, 25(2):401–405, June 1954.
- J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.
- S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- B. Lau, K. O. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, 2(1):19–30, March 2010.
- F. Lian, C.-Z. Han, W.-F. Liu, X.-X. Yan, and H.-Y. Zhou. Sequential Monte Carlo implementation and state extraction of the group probability hypothesis density filter for partly unresolvable group targets-tracking problem. *IET Radar, Sonar and Navigation*, 4(5):685–702, October 2010.

- J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL <http://users.isy.liu.se/johanl/yalmip>.
- J. Löfberg. Automatic robust convex programming. *Optimization methods and software*, 27(1):115–129, 2012.
- C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 49–56, Chicago, IL, USA, July 2011.
- R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, USA, 2007.
- R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 915–921, Seattle, WA, USA, July 2009.
- A. R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, July 2007.
- D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):389–395, December 2003.
- D. Schieferdecker and M. F. Huber. Gaussian Mixture Reduction via Clustering. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Seattle, WA, USA, July 2009.
- D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, August 2008.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, second edition, 1993.
- B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.
- J. L. Williams and P. S. Maybeck. Cost-Function-Based Gaussian Mixture Reduction for Target Tracking. In *Proceedings of the International Conference on Information Fusion (FUSION)*, Cairns, Queensland, Australia, July 2003.
- H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 73–78, Chicago, IL, USA, July 2011.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.

O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.

R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.

D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.

R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.

J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.

E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.

M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.

T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

J. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.

C. Lundquist: Sensor Fusion for Automotive Applications. Thesis No. 1409, 2011. ISBN 978-91-7393-023-9.

P. Skoglar: Tracking and Planning for Surveillance Applications. Thesis No. 1432, 2012. ISBN 978-91-7519-941-2.