

A flexible scalable video coding framework with adaptive spatio-temporal decompositions

Nikola Šprljan

Department of Electronic Engineering

Queen Mary, University of London

Thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

August 2006

Abstract

The work presented in this thesis covers topics that extend the scalability functionalities in video coding and improve the compression performance. Two main novel approaches are presented, each targeting a different part of the scalable video coding (SVC) architecture: motion adaptive wavelet transform based on the wavelet transform in lifting implementation, and a design of a flexible framework for generalised spatio-temporal decomposition.

Motion adaptive wavelet transform is based on the newly introduced concept of connectivity-map. The connectivity-map describes the underlying irregular structure of regularly sampled data. To enable a scalable representation of the connectivity-map, the corresponding analysis and synthesis operations have been derived. These are then employed to define a joint wavelet connectivity-map decomposition that serves as an adaptive alternative to the conventional wavelet decomposition. To demonstrate its applicability, the presented decomposition scheme is used in the proposed SVC framework, as a substitute for the spatial transform of the high-pass temporal frames.

The proposed novel decomposition framework unifies different SVC architectures, thus providing a flexible platform for adaptive selection of the decomposition path according to the requirements. A concept of the spatio-temporal decomposition tree is introduced, which describes flexible decomposition setting. The corresponding SVC bit-stream is organised in such way that, apart from providing an uneven spatio-temporal plane partition, supports an unequal number of quality layers within each spatio-temporal resolution. Optimisation of performance on the target bit-rates is accomplished by developing a

simple statistical model for reconstruction distortion estimation and by applying a post-compression optimisation algorithm on the embedded video bit-stream.

The developed framework for scalable video coding provides targeted highly flexible decompositions and therefore can be use in different demanding scenarios. Newly introduced methods for rate-distortion optimisation and adaptive transformation enhance the compression performance and adaptability of the coding scheme to the underlying content.

Acknowledgements

First of all I would like to express my gratitude to Prof. Ebroul Izquierdo, who has supervised my work for the last three years and made my stay in London during that period feasible. During this time I had a chance to work on different challenging topics and to learn from many experts, both internal to the Multimedia and Vision (MMV) Group as well as from academic visitors and project partners. Specifically, my research work has been supported and conducted within the aceMedia project of the European Commission. I am particularly grateful for the constructive comments and ideas from the following people with whom I worked on the aceMedia project: Dr. Charith Abhayaratne, Toni Žgaljić, Luis Herranz, Tony May, Dr. Jonathan Teh, Naeem Ramzan and Shuai Wan.

I would like to thank all my colleagues from MMV group with whom I spent many hours in discussion and solving problems. I would not have been able to start my PhD research without the knowledge I gained during my engagement with the University of Zagreb. Therefore I would like to thank all the members of Video Communication Laboratory there for all their support that led to my fascination with multimedia technology.

A special thanks goes to my partner in all senses of the word - my wife Marta, for her encouragement, inspiration and all the sacrifices she has made for me. Finally, I would like to thank my friends and family for constant support and love despite my often absence from their lives caused by my obsession with work.

Contents

Glossary	6
1 Introduction	10
1.1 Basic Concepts in Scalable Video Coding	15
1.2 Thesis Contributions	19
1.3 Thesis Structure	20
2 Wavelet Transform	22
2.1 Multiresolution Formulation	24
2.2 Relation to FIR Filterbanks	27
2.3 Wavelet Filterbank Existence Conditions and Properties	34
2.4 The Lifting Scheme	39
3 SVC Architectures	44
3.1 Motion Compensated Temporal Filtering	47
3.2 Spatial Domain MCTF	53
3.3 Multi-scale Pyramid Architectures	55
3.4 In-band MCTF	57
3.5 Generalised Spatio-Temporal Scalability	60
4 SVC framework for adaptive spatio-temporal decomposition	62
4.1 Full Scalability with Constrained Number of Layers	62
4.2 Spatio-Temporal Decomposition Tree	67
4.3 Comparison of Architectures Using Decomposition Trees	79
4.4 Post-Compression Rate-Distortion Optimisation	83

4.5 Comparison with Existing SVC Codecs	93
5 Adaptive Spatial Transform for Scalable Coding	95
5.1 Properties of the Temporal High-pass frames	97
5.2 A Joint Wavelet Connectivity-Map Decomposition	103
5.3 Motion-Driven Adaptive Transform (MDAT)	113
5.3.1 MDAT for General Motion Vector Fields	113
5.3.2 MDAT for Intra-Inter Boundaries	116
5.3.3 Postprocessing	119
5.3.4 Experiments in SVC Framework	121
5.4 Object-based Coding Using Connectivity-Map Decomposition . .	130
6 Conclusions	135
Appendix A - Wavelets	139
A.1 Wavelet and Scaling Functions and Filters	139
A.2 Frequency and Phase Characteristics	144
A.3 Properties and Lifting Coefficients	146
Appendix B - Subband Weighting Tests	148
B.1 Distortion Estimation	148
B.2 Reconstructed Noise Energy Distribution	151
References	166

Glossary

Roman letters

- G_0 z-transform representation of the wavelet low-pass synthesis filter
- g_0 Wavelet low-pass synthesis filter
- G_1 z-transform representation of the wavelet high-pass synthesis filter
- g_1 Wavelet high-pass synthesis filter
- H_0 z-transform representation of the wavelet low-pass analysis filter
- h_0 Wavelet low-pass analysis filter
- H_1 z-transform representation of the wavelet high-pass analysis filter
- h_1 Wavelet high-pass analysis filter
- $[u, v]$ Motion vector composed of vertical u , and horizontal v displacement
- \mathbb{Z} Set of integer numbers
- \mathbb{Z}^* Set of nonnegative integers
- \mathbb{R} Set of real numbers
- L^2 Space of square integrable functions

Greek letters

- $\delta(x)$ Dirac delta function, with the fundamental property that $\int f(x)\delta(x - a)dx = f(a)$; the discrete version is denoted with δ_k

φ Scaling function

ψ Wavelet function

Operators

◦ Functional composition

* Convolution, as in $(f * g)_m = \sum_n f_n g_{m-n}$

↓ Downsampling

\mathcal{A} Analysis

\mathcal{A}_S Spatial analysis

\mathcal{A}_T Temporal analysis

\mathcal{S} Synthesis

\mathcal{S}_S Spatial synthesis

\mathcal{S}_T Temporal synthesis

$\langle \cdot, \cdot \rangle$ Inner product, for functions $f(t), g(t) \in L^2$ defined as $\int f(t)g(t)dt$

\mathcal{I} Interpolation

\mathcal{P} Prediction

\mathcal{U} Update

Superscripts

$x^{(j)}$ j represents the scale of x

Acronyms

4CIF $4 \times$ CIF; frame of dimensions 704×576 pixels

ACE Autonomous Content Entity

- aceMedia “Integrating knowledge, semantics and content for user-centred intelligent media services” - project supported by European Commission under contract FP6-001765 aceMedia
- aceSVC aceMedia Scalable Video Codec
- AVC Advanced Video Coding, MPEG-4 Part 10 (ISO/IEC 14496-10) standard for coding of video; see H.264
- BIBO Bounded Input Bounded Output
- CIF Common Interchange Format; frame of dimensions 352×288 pixels
- CSRE Cyclostationary Reconstruction Error
- DCT Discrete Cosine Transform
- DIA Digital Item Adaptation
- DWT Discrete Wavelet Transform
- EBCOT Embedded Block Coding with Optimized Truncation of the embedded bit-streams
- EEP Equal Error Protection
- EZBC Embedded ZeroBlock Coder
- FIR Finite Impulse Response
- GOP Group of Pictures
- GSTS Generalised Spatio-Temporal Scalability
- H.264 ITU-T Video Coding Experts Group standard for coding of video; see AVC
- ITU International Telecommunication Union
- JPEG Joint Photographic Experts Group, as in standard on compression of still images

- JSVM Joint Scalable Video Model; see H.264/AVC
- JVT Joint Video Team, a joint project between MPEG and ITU-T for the development of new video coding standard, AVC/H.264
- MC Motion Compensation
- MCTF Motion Compensated Temporal Filtering
- MDAT Motion-Driven Adaptive Transform
- ME Motion Estimation
- MPEG Moving Picture Experts Group, as in standards on compression of video - MPEG-1, MPEG-2, MPEG-4
- MSB Most Significant Bit
- MV Motion Vectors
- OBMC Overlapped Block Motion Compensation
- PCR Peak CSRE Ratio
- PCRD Post-Compression Rate-Distortion (optimisation)
- PR Perfect Reconstruction
- PSNR Peak Signal-to-Noise Ratio
- QCIF quarter-CIF; frame of dimensions 176×144 pixels
- SPIHT Set Partitioning in Hierarchical Trees
- SSM Structured Scalable Metaformats
- SVC Scalable Video Coding
- $t+2D$ SVC architecture employing motion compensation in the original frame resolution
- UEP Unequal Error Protection

Chapter 1

Introduction

Scalable coding is expected to become more important in the coming decade, as diverse video distribution networks are becoming ever more important parts of larger heterogeneous networks. At the same time, the steady growth of available bandwidth has put multimedia content at the forefront of consumer interest, but to deliver this content “anywhere and anytime” a high degree of adaptability is sought. Two main options are available for achieving these goals. The first is transcoding, which is a mature technology utilising conventional encoding/decoding tools for adaptation of compressed content [1]. The second is scalable coding, based on embedded and hierarchical coding that enables seamless scaling of the content. Since it offers a solution for already available coding formats, the main advantage of transcoding is the ease of employment. Scalable coding, a relatively new technology, represents a low-cost alternative to transcoding, in terms of computational complexity of adaptation. The alternative approach to adaptation has so far been to store and transmit multiple bit-streams, each for a particular set of receiver conditions. However, this scheme is wasteful of both storage space and bandwidth, and is becoming seriously inefficient with the ongoing broadening of the range of possible transmission conditions.

In the past few years research on Scalable Video Coding (SVC) based on wavelets has intensified and has seen many efficient techniques, each improving a certain aspect of scalability. However, the most difficult question in scalable coding has remained open: how close the compression efficiency of a scalable codec has to get close to a non-scalable one, in order that scalability function-

ality outweighs the loss in compression performance. The existence of this performance gap is well known from the Information Theory [2], as by enabling multiple decoding points the optimal codeword is effectively broken down into smaller ones, leading to suboptimal compression. However, compression performance is an important factor only up to a certain level of improvement. Since the current state-of-the-art video compression systems have reached their mature phase, *e.g.*, with the H.264/AVC video coding standard [3], further enhancements have become painstakingly difficult to achieve without a significant increase in complexity. Also, the importance of error protection in multi-user environments has shifted attention from mere compression to robust coding for lossy environments. Methods known under common terms *joint source-channel coding* and *multiple description coding* are at the forefront of current research activities [4]. SVC fits nicely into this context, as it already provides an inherent prioritisation among the compressed data.

Looking from this perspective, other functional properties of scalable coding are becoming more interesting, as they can enable utilisation in yet unexplored application niches. One example is the possibility of recombining the higher quality bit-stream, using the already downloaded lower quality bit-stream and subsequently downloaded *refinement* quality layers. This could pave the way to new services using innovative pricing policies [5]. The ease with which the scalable coded content can be manipulated and shared on one side, and the emergence of various types of communication and distribution networks and the numerous ways in which these networks interact on the other side, will certainly open a path to creative new ways of utilisation of the content, many of which are difficult to foresee at the present. Possibilities include remote browsing [6; 7], fast catalogue or summary creation [8], fast video editing and production, on-the-fly switching between different devices and networks, *etc.*

One emerging application is a video distribution system with *multistage* adaptation, whose hypothetical application scenario is depicted in Fig. 1.1. Providers of video content typically require content of different fidelities - high quality material for storage and future editing as well as lower bit-rate content for distribution. If encoded in a scalable way, content can be efficiently scaled for different distribution requirements. In the example from Fig. 1.1, the provider supports various

distribution routes, during which a further adaptation of the content to the specific requirements can take place. In the scenario depicted in Fig. 1.1 the content provider directly supplies several customers, for instance a digital cinema enabled customer with a content of cinema-projection quality, an Internet provider with the standard definition quality, and a company that owns both mobile and terrestrial TV with high definition quality. For some applications, such as live event streaming [9], it is essential that content can be adapted on route to delivery, to accommodate end-users with different display capabilities. To achieve this, the service providers dynamically adapt the content at adaptation-enabled routers. This form of adaptation is called *multistage* adaptation, as it involves the adaptation of an already adapted content. Although this example represents an overly-optimistic case scenario, it gives a good sense of the wide fidelity range which SVC is striving for.

Another possible application targets transmission of digital video in the lossy environments [10; 11]. The multi-layered organisation of scalable coded video offers a possibility to employ the Unequal Error Protection (UEP) scheme, where the channel bits designated to protect the source bits are allocated in such way that layers that convey more information are protected more than other layers. In the contrast to Equal Error Protection (EEP) schemes, UEP offers “graceful degradation” of quality of reception when the transmission conditions on the channel vary. This is depicted in Fig. 1.2 where three methods of protecting the video bit-stream are presented. As the name suggests, the EEP schemes protect the bit-stream equally, so in the case that the channel state is unknown, the strength of protection is adjusted for the most likely state. For a case when protection is adjusted for transmission over a channel in a bad state, the available channel capacity is presumed to be low, which results in over-protection for time intervals when the channel is in a good state. On the other hand, when the transmitted bit-stream is adjusted for reception when the channel is in a good state, when the channel enters a bad state, e.g., during a fading interval, the video bit-stream cannot be decoded correctly. This results in a drastic drop in the visual quality if decoding of the transmitted video bit-stream is attempted - the so-called “cliff effect” occurs. On the other hand, UEP approach can achieve robustness of signal reception that is found in analog systems, where the perceptual quality

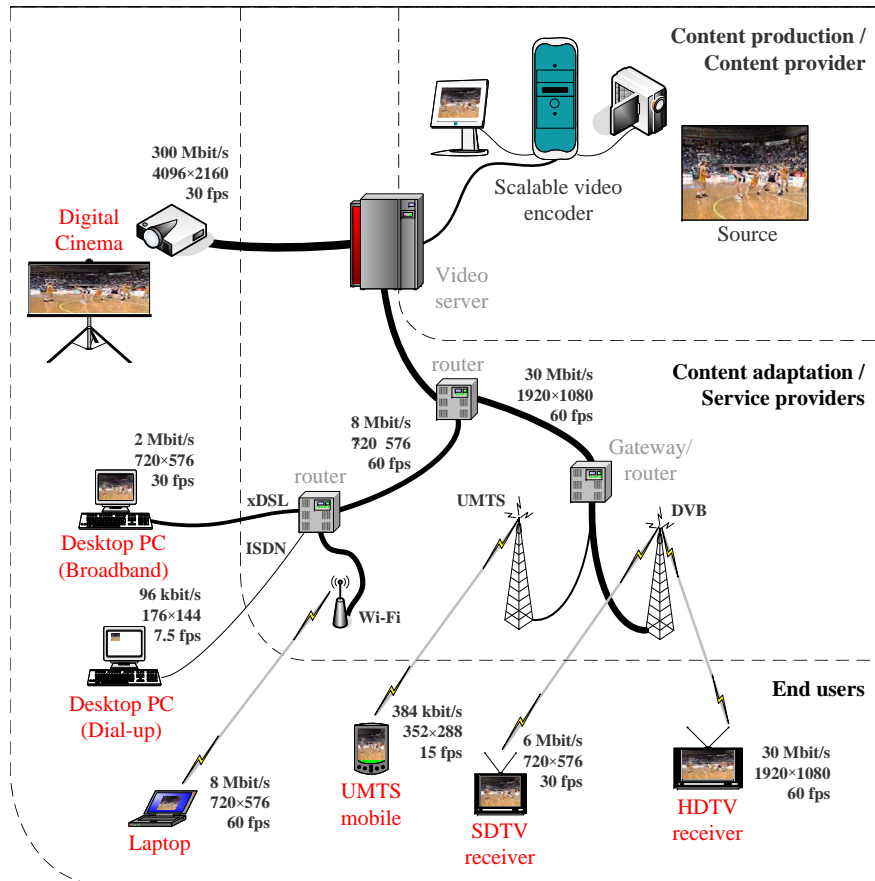
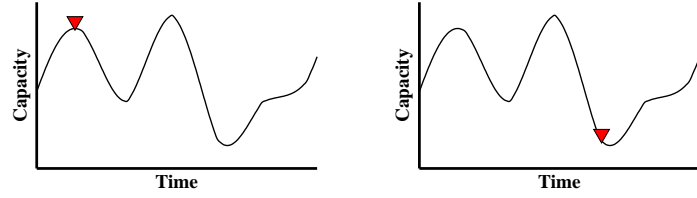


Figure 1.1: SVC in a hypothetical application scenario – multistage adaptation of video for distribution over heterogeneous network.

is almost in a direct correspondence with the channel capacity. This property of transmission is called “graceful degradation” of quality. There are already many UEP methods devised for scalable coding of still images and video [12; 13; 14; 15; 16; 17].

The potentials of scalable coding have been recognised by the aceMedia project consortium [18]. The main target of the aceMedia project is to create a framework in which content will be self-adaptive and that provides knowledge discovery. Central to aceMedia is the concept of Autonomous Content Entity (ACE) that can be created by content provider or can be a personal content. In both cases adaptability plays an important role because of the high demands for content scaling and sharing. Since video is the most demanding type of content the fo-



(a) Channel in a good state (b) Channel in a bad state



(c) Good state, EEP, low source bit-rate (d) Good state, EEP, high source bit-rate (e) Good state, UEP



(f) Bad state, EEP, low source bit-rate (g) Bad state, EEP, high source bit-rate (h) Bad state, UEP

Figure 1.2: Transmission of video in lossy environment.

cus on research within the project has been on the development of the aceMedia Scalable Video Codec (aceSVC). The main part of that work is presented in this thesis. Specifically, besides the background overview of the techniques behind scalable video coding, this thesis presents the architecture of aceSVC and new schemes that enhance the compression performance in scalable coding.

1.1 Basic Concepts in Scalable Video Coding

Scalable coding is sometimes referred to as *progressive* coding. However, progressive coding refers only to an encoded content that is organised in layers and can be rendered progressively, while scalable coding is a more general term as it encompasses progressive coding, but also adopts some other concepts, such as *scalable complexity* coding [19]. Although many efficient solutions for scalable coding of still images have been devised in the past, scalable coding of video is still a topic of intensive research. The main reason is that when *combined scalability* is sought, the problem of efficient coding becomes more difficult. In contrast to *limited scalability*, when only one dimension of scalability is sufficient, for instance either spatial or quality scalability but not both, combined scalability refers to a case where several modes of scalable content representation “coexist” in the same bit-stream. Limited scalability can also refer to a case when only a small number of layers exist in a bit-stream, and therefore a small range of bit-rates is available, while the opposite is *full scalability* where the range is wide and many decoding points are available. The ultimate case of full scalability is *fine-granular scalability* where the content is encoded in an embedded way and can be truncated at any byte position, while still remaining decodable. Of course, the exact quantification of “many points” and “wide range” can be the subject of debate. However, it must be kept in mind that the crucial determining factor here is the particular application scenario as it dictates the required type of scalability.

As the existing video coding standards provide only limited scalability functionalities [20], new ways of improving adaptability have been one of the main areas of research in video coding during last few years. Early work on fully scalable video coding has been based on employment of the wavelet transform due to its multiresolution signal representation capabilities and high compression performance. Soon after this, the main requirements for scalable video coding were determined - the main features that a scalable video framework has to fulfil are:

- Spatial scalability – related to scaling of frame dimensions, *i.e.*, frame resolution. A common functionality needed in systems that require video adaptation is scaling of resolution, usually by the power of 2 in each dimension.

1.1 Basic Concepts in Scalable Video Coding

For example, if the original frame resolution is 4CIF (704×576 pixels), then two levels of spatial resolution scalability mean that the encoded sequence can be scaled to half resolution: CIF (352×288 pixels) and quarter resolution QCIF (176×144 pixels). Spatial scalability is important for applications that target different devices and also provide an efficient means of bit-rate reduction.

- Temporal scalability – related to the reduction in number of frames per second (fps, Hz), *i.e.*, to the scaling of frame rate, or temporal resolution. If dyadic scaling is assumed, then the embedded representation of a sequence of originally 60 Hz can contain layers of 30 Hz, 15 Hz, 7.5 Hz, *etc.* The applications that require this type of scalability are related to devices that because of limited display capabilities, for instance a limited display refresh rate, processing power and memory requirements, cannot handle high frame-rate sequences.
- Quality scalability – related to reduction of the quality of the reconstructed sequence, also known as SNR (signal-to-noise ratio) scalability. While temporal and spatial scalabilities provide only a finite number of adaptation points, fine tuning of the bit-rate on each of the supported spatio-temporal resolutions can be achieved using fine-granular quality scalability. Although the subjective quality of a decoded video sequence is not enhanced by any amount of additional bits, this functionality facilitates some applications that rely on precise truncation, like Internet streaming and data storage, and provides a base for unequal error protection with graceful degradation of quality.

While the encoder produces compressed content and organises it into a scalable bit-stream, and thus undertakes most of the computationally complex operations, the essential role in scalable video coding is played by a module that performs adaptation - the so called *extractor*. An extractor is intended to be the simplest possible component in the scalable video compression and transmission chain, as it replaces the transcoder that is used in adaptation of non-scalable content. While the transcoder relies on relatively complex operations, including entropy coding and either a full or partial

prediction and transform coding, an extractor only performs parsing of the control data in the input scalable coded bit-stream, and in this way selects and extracts only the relevant bit-stream portions. The output bit-stream is composed only of the extracted parts of the input bit-stream, corresponding to the targeted adaptation parameters, and can be decoded by a scalable video decoder. These three modules and their roles in the transmission chain are depicted Fig. 1.3. The description of the notation used for the depicted scalable bit-stream is provided in Chapter 4. The “bit-stream description” refers to the control data, necessary for the extractor to perform the adaptation.

Basic scalability functionalities in still image coding have already been available within the Discrete Cosine Transform (DCT) based JPEG still image coding standard, but only in a limited way, since combined scalability decreases coding efficiency considerably [21]. Apart from the baseline mode, JPEG in progressive mode supports progression by quality with successive approximation and spectral selection methods, in which the DCT coefficients are encoded in bit-plane by bit-plane scans or are partitioned into successive scans [22]. Resolution scalability (called “hierarchical refinement” in JPEG) can be achieved by compression of multiple images that correspond to different resolution levels. However, this particular mode of JPEG considerably reduces the overall performance. It has been shown that solutions based on the application of wavelet transform, such as Fully Scalable-Set Partitioning in Hierarchical Trees (FS-SPIHT) [23] and Embedded ZeroBlock Coder (EZBC) [24], provide flexible frameworks for achieving both fine granular quality as well as resolution scalability inherited from multiresolution properties of the wavelet transform. These new functionalities are standardised within the JPEG-2000 still image compression standard [22; 25; 26] that employs the wavelet transform coupled with Embedded Block Coding with Optimized Truncation of the embedded bit-streams (EBCOT) entropy coding scheme [27].

In video coding the temporal dimension introduces additional challenges for achieving full scalability. A certain degree of scalability has through layered coding already been supported in the DCT-based methods, for instance in MPEG-2 and MPEG-4 [28] video coding standards. Current JVT (MPEG + ITU) stan-

1.1 Basic Concepts in Scalable Video Coding

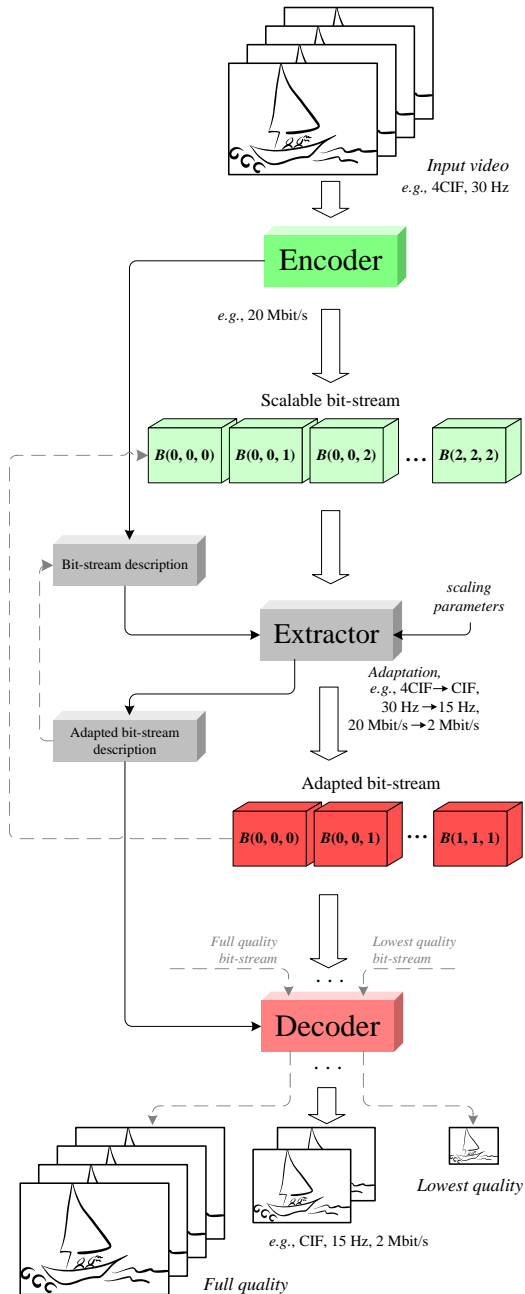


Figure 1.3: Basic modules in SVC – encoder, extractor and decoder.

standardisation activities work toward the scalable extension of H.264 / AVC video coding standard [29], based on the DCT-like block transform. Beside added scalability functionalities, this approach supports backward compatibility with

already commercially used non-scalable H.264 / AVC video compression. On the other hand, scalable coding based on the wavelet transform also provides efficient coding frameworks offering a high degree of adaptability and flexibility. The most recent research activities have resulted in several highly scalable video coding frameworks [30; 31; 32; 33], based on an efficient combination of techniques that provide spatial and temporal scalability functionalities combined with fine-granular quality scalability.

1.2 Thesis Contributions

The primary contribution of this thesis is the novel conceptual framework for scalable video coding embodied in the developed codec - aceSVC. Specific contributions include:

- summary of key properties of wavelet filterbanks influencing their performance when employed in SVC, including the newly introduced measure - peak cyclostationary reconstruction error ratio,
- formalisation of the idea of a spatio-temporal decomposition and design of a decomposition framework supporting adaptive scalability range,
- design of a flexible bit-stream organisation supporting full scalability with constrained number of layers,
- performance analysis for different spatio-temporal decompositions,
- method for post-compression rate-distortion optimisation applied to the EZBC wavelet coder,
- motion adaptive spatial wavelet transform,
- joint wavelet connectivity-map decomposition scheme with application in motion adaptive transform and object based coding.

1.3 Thesis Structure

Chapter 2 introduces the discrete wavelet transform in an accessible way, by building on the intuitive concept of multi-resolution analysis in the $L^2(\mathbb{R})$ space and by establishing the relation to the continuous scaling and wavelet functions. The basic conditions for existence of wavelet and scaling functions are presented, and the properties of discrete wavelet transform that are relevant for application in compression are derived. The summarised properties of a particular wavelet are then used to establish a link to its compression performance behaviour. Lastly, the lifting implementation of the discrete wavelet transform is described, which is a base for the novel adaptive transform presented in Chapter 5.

With wavelet transform, as the basic building block of the wavelet-based scalable video coding systems, introduced, Chapter 3 provides an overview of the architectural aspects of such systems. As a prerequisite, the Motion Compensated Temporal Filtering (MCTF) is explained, as it plays a major role in the compression efficiency. In what follows, the main architectures are described, namely the spatial domain MCTF, in-band MCTF, the redundant multi-scale pyramid and generalised spatio-temporal scalability. By stressing the advantages and drawbacks of each architecture, this chapter brings an in-depth understanding of the technological area.

Chapter 4 presents an efficient decomposition framework that unifies different SVC architectures, thus providing a flexible platform for adaptive selection of the decomposition path according to the requirements. Building on the Structured Scalable Metaformats framework, the concept of constrained full scalability mode is introduced. To support this mode of scalability for video, a spatio-temporal decomposition tree is defined, and its employment to achieve different SVC architectures is examined. The corresponding SVC bit-stream is organised in such a way that, apart from providing an uneven spatio-temporal plane partition, it also supports an unequal number of quality layers within each spatio-temporal resolution. Optimisation of performance for the target bit-rates is accomplished by developing a simple statistical model for estimation of the signal-domain distortion and by applying a post-compression optimisation algorithm on the embedded bit-stream. Some compression performance results are presented demonstrating

the influence of different parameters. Also, a comparison with most popular scalable coding architectures is provided.

Chapter 5 addresses efficient compression of the high-pass temporal frames. The proposed solution for processing this type of frames is based on a novel adaptive wavelet transform, and thus the discussion that follows covers a methodological framework necessary for its introduction. It is based on the well-known lifting technique and the newly introduced connectivity-map concept, which enables locally adaptive filtering. In the proposed scheme the connectivity values are used in the lifting by means of weights applied to the neighbouring signal samples. To enable a scalable representation of the connectivity-map, the corresponding analysis and synthesis operations are derived. Then they are used to define a joint wavelet connectivity-map decomposition that serves as an adaptive alternative to the conventional wavelet decomposition. Since motion compensation imposes a spatial structure on temporal frames, especially with employment of the intra blocks, the connectivity-map is used to capture the local discontinuities and to drive the adaptation. The connectivity-map is generated from motion information, which is available at both the encoder and the decoder sides. In addition to better energy compaction, experimental results reported show improved compression performance for spatial, temporal and quality scalability scenarios.

Chapter 6 concludes the thesis and provides some directions for the future research.

Chapter 2

Wavelet Transform

Wavelets have achieved a tremendous success in signal processing in the past decade and a half. Discrete Wavelet Transform (DWT) has been recognised as a versatile tool for signal description and approximation, which has led into its application for data compression. Some of the important features of the wavelets that have made them so attractive for the application in image and video coding are the following:

- Wavelets are localised in both space (time) and frequency, unlike sinusoidal basic functions in Fourier transform, that possess a perfect frequency localisation but on the other hand provides no information on time. This makes wavelets very useful for describing local features and discontinuities in the signal.
- Wavelets are based on multi-resolution analysis and therefore provide an efficient description of signals in multiple scales. This fact naturally leads to their application in scalable coding.
- Wavelets are smooth functions, which is useful for describing certain classes of smooth signals – to which also the natural images belong.
- DWT can be designed to be a non-expansive (non-redundant) transform, meaning that the number of transform coefficients is equal to the number of source signal samples.

In the discrete setting, the wavelet transform can be viewed as application of filtering with Finite Impulse Response (FIR) filterbank. As the conditions of existence for a wavelet filterbank offer many degrees of freedom in calculating the filter coefficients, different wavelets have been devised for application in image and video coding. The design criteria may be based on optimising some of the wavelet filter properties, considered to be important for its compression performance, such as smoothness, orthogonality, or the shape of the frequency response; but the design can be also application-driven. Various application requirements can be satisfied by using different wavelets - shorter wavelet filters for low complexity devices; wavelets with better low-pass frequency characteristic causing less spatial aliasing artefacts for an improved spatial scalability performance; wavelets for temporal filtering with delta low-pass filters for shortening the decoding delay and avoiding the temporal artefacts, *etc.*

This chapter provides a basic theoretical background for the discrete wavelet transform given from the perspective of multi-resolution analysis. It also gives definitions of scaling and wavelet functions and requirements for their existence. Well known results from literature are reviewed and presented in concise manner, while a more rigorous and comprehensive treatment can be found in [22; 34; 35; 36; 37; 38]. The discussion here is limited to signals in $L^2(\mathbb{R})$, *i.e.*, space of real-valued square integrable functions, as this is sufficient for the targeted application of wavelets in SVC. Derivation of the basic wavelet relations presented here is limited also to the two-band non-expansive DWT, as this one is of the most practical significance for the SVC techniques described in the following chapters. Besides the presented derivation, wavelet transform can be generalised in various other ways, for instance continuous wavelet transform (CWT), complex-valued wavelet transform, and also multi-band and expansive (over-complete) DWT.

Spatial, or 2D wavelets are usually constructed as tensor products of 1D wavelets, and therefore their 2D transform can be implemented in a separable fashion. The separability refers to the equivalence of the order of directions in which 1D wavelets are applied since the result is identical, *i.e.*, it can be applied first on rows and then on columns or vice-versa. Consequently, as they can be treated separately, the following discussion is restricted to 1D wavelet transform without losing generality but adding clarity to the analysis.

2.1 Multiresolution Formulation

The notion of scale or resolution is related to the notion of space, such that multiresolution analysis can be viewed through the concept of nested spaces:

$$\dots \subset \mathcal{V}^{(-2)} \subset \mathcal{V}^{(-1)} \subset \mathcal{V}^{(0)} \subset \mathcal{V}^{(1)} \subset \mathcal{V}^{(2)} \subset \dots \subset L^2,$$

where superscript (j) denotes the scale of the space; a lower value of j corresponds to a coarser scale, while a higher value corresponds to a finer scale. In other words, the space that contains high resolution signals also contains those of lower resolution. These nested spaces span the whole range of signal resolutions, so that in the limiting cases:

$$\mathcal{V}^{(-\infty)} = \{0\}, \text{ and } \mathcal{V}^{(\infty)} = L^2.$$

Scaling function $\varphi(t) \in \mathcal{V}^{(0)}$, where $\varphi(t) \equiv \varphi^{(0)}(t)$, is such function whose set of scaled (dilated or contracted) and translated versions represent a basis set for a particular function space. Scaling function can be regarded to be a “wavelet”, under a broad meaning of the term that encompasses waveforms spanning a particular function space, while the actual wavelet function is defined later in the discussion. The product of the scaling function with itself is defined to be $\langle \varphi(t), \varphi(t) \rangle = 1$, *i.e.*, it has unit norm in L^2 , expressed as $\|\varphi(t)\| = 1$. With a set of integer translates $k \in \mathbb{Z}$ of a basic scaling function $\varphi(t)$ at scale j , an expansion set for the function space $\mathcal{V}^{(j)}$ can be defined with:

$$\varphi_k^{(j)}(t) = 2^{j/2} \varphi(2^j t - k). \tag{2.1}$$

Defined in this way, scaling is dyadic and unit norm is preserved across the scales, *i.e.*, $\|\varphi_k^{(j)}(t)\| = 1$. It can be seen that the scaled and translated versions of the basic scaling function form a two-dimensional (j, k) family of functions.

Then any $f(t) \in \mathcal{V}^{(j)}$ can be represented in terms of a weighted sum of the scaling functions belonging to the basis set of $\mathcal{V}^{(j)}$:

$$f(t) = \sum_k x_k^{(j)} \varphi_k^{(j)}(t), \tag{2.2}$$

where $x_k^{(j)}$ are the representation coefficients of signal $f(t)$ at scale j . If we assume that the scaling function is orthogonal to its translations, *i.e.*, $\langle \varphi_k^{(j)}(t), \varphi_i^{(j)}(t) \rangle =$

2.1 Multiresolution Formulation

δ_{k-l} , then the representation coefficients are given by computing the inner product of both sides of (2.2) with $\varphi_l^{(j)}(t)$:

$$x_l^{(j)} = \langle f(t), \varphi_l^{(j)}(t) \rangle. \quad (2.3)$$

As no such assumption regarding orthogonality is made here, the representation coefficients $x_l^{(j)}$ will here be derived in a different way.

Since $\mathcal{V}^{(j)} \subset \mathcal{V}^{(j+1)}$, any $f(t) \in \mathcal{V}^{(j)}$ can also be represented with the basis set of $\mathcal{V}^{(j+1)}$. Since $\varphi_k^{(j)}(t)$ is also a function in space $\mathcal{V}^{(j)}$, it follows that it can be also represented in the same way – in terms of a weighted sum of shifted $\varphi_k^{(j+1)}(t)$. The basic *recursion equation* for the scaling function, similar to (2.2), is then formulated as:

$$\varphi_k^{(j)}(t) = \sum_n g_{0,n} \varphi_{2k+n}^{(j+1)}(t), \quad (2.4)$$

where $g_{0,n}$ are coefficients of the *scaling filter* g_0 , and the shifts are $n \in \mathbb{Z}$. The factor 2 for shift k at the scale $j + 1$ is to compensate for the scale difference. Assuming that $(j, k) = (0, 0)$, and by using (2.1), this can be written in a simpler way as:

$$\varphi(t) = \sum_n g_{0,n} \sqrt{2} \varphi(2t - n). \quad (2.5)$$

This recursive relation points to the self-similarity of the scaling function, thus indicating the fractal nature of wavelets.

The basic wavelet function $\psi(t)$, also known under name “mother” wavelet, is defined similarly as the scaling function. It is defined through its associated space $\mathcal{W}^{(0)}$, such that $\psi(t) \in \mathcal{W}^{(0)}$, where $\mathcal{W}^{(j)}$ is required to satisfy:

$$\mathcal{V}^{(j+1)} = \mathcal{V}^{(j)} \cup \mathcal{W}^{(j)}. \quad (2.6)$$

Note that this requirement is weaker than the requirement that the space $\mathcal{W}^{(j)}$ is a complementary space to $\mathcal{V}^{(j)}$, such that $\mathcal{W}^{(j)} \cap \mathcal{V}^{(j)} = \emptyset$. In other words, in the latter case the spaces are orthogonal, $\mathcal{W}^{(j)} \perp \mathcal{V}^{(j)}$, so that $\mathcal{V}^{(j+1)}$ is given by a direct sum, $\mathcal{V}^{(j+1)} = \mathcal{V}^{(j)} \oplus \mathcal{W}^{(j)}$. On the other hand, the definition as in (2.6) allows for a more general wavelets systems, commonly known under term *biorthogonal*.

2.1 Multiresolution Formulation

Since from (2.6) follows that $\mathcal{W}^{(j)} \subset \mathcal{V}^{(j+1)}$, a recursion equation for the wavelet function, analogous to (2.4) can be written as:

$$\psi_k^{(j)}(t) = \sum_n g_{1,n} \varphi_{2k+n}^{(j+1)}(t), \quad (2.7)$$

where $g_{1,n}$ are the coefficients of the wavelet filter g_1 . For $(j, k) = (0, 0)$, it follows:

$$\psi(t) = \sum_n g_{1,n} \sqrt{2} \varphi(2t - n). \quad (2.8)$$

Because of (2.6), any $f(t) \in \mathcal{V}^{(j+1)}$ can be expressed as a combination of scaled and translated scaling and wavelet functions, $\varphi_k^{(j)}(t)$ and $\psi_k^{(j)}(t)$:

$$f(t) = \sum_k x_k^{(j)} \varphi_k^{(j)}(t) + \sum_k y_k^{(j)} \psi_k^{(j)}(t). \quad (2.9)$$

Defined like this, $x_k^{(j)}$ are called *approximation* coefficients, and $y_k^{(j)}$ *detail* coefficients. Through $\varphi_k^{(j)}(t)$ the function $f(t)$ is approximated in the space $\mathcal{V}^{(j)}$, while the space $\mathcal{W}^{(j)}$ contains the difference, or the details, represented with $\psi_k^{(j)}(t)$.

As it has not been assumed that their corresponding function spaces are orthogonal, $\varphi(t)$ and $\psi(t)$ must be taken to be generally non-orthogonal:

$$\langle \varphi_k^{(j)}(t), \psi_l^{(j)}(t) \rangle \neq 0.$$

Therefore, a dual set of functions, *dual scaling* function $\tilde{\varphi}_l^{(j)}(t)$ and *dual wavelet* function $\tilde{\psi}_l^{(j)}(t)$, is introduced, that is required to satisfy:

$$\begin{aligned} \langle \psi_k^{(j)}(t), \tilde{\varphi}_l^{(j)}(t) \rangle &= 0, & \langle \varphi_k^{(j)}(t), \tilde{\varphi}_l^{(j)}(t) \rangle &= \delta_{k-l}, \\ \langle \varphi_k^{(j)}(t), \tilde{\psi}_l^{(j)}(t) \rangle &= 0, & \langle \psi_k^{(j)}(t), \tilde{\psi}_l^{(j)}(t) \rangle &= \delta_{k-l}, \end{aligned} \quad (2.10)$$

or in other words, the dual set is orthogonal to the *primal* set. Due to this property this wavelet system is called biorthogonal. The corresponding recursion equations, analogous to (2.4) and (2.7) are:

$$\tilde{\varphi}_k^{(j)}(t) = \sum_n h_{0,n} \tilde{\varphi}_{2k+n}^{(j+1)}(t), \quad (2.11)$$

$$\tilde{\psi}_k^{(j)}(t) = \sum_n h_{1,n} \tilde{\varphi}_{2k+n}^{(j+1)}(t), \quad (2.12)$$

where $h_{0,n}$ and $h_{1,n}$ are the coefficients of the dual scaling and wavelet filters, h_0 and h_1 , respectively.

By using (2.10) and by taking an inner product on the both sides of (2.9), it can be readily checked that the approximation and details coefficients are given by:

$$x_k^{(j)} = \langle f(t), \tilde{\varphi}_k^{(j)}(t) \rangle \quad (2.13)$$

$$y_k^{(j)} = \langle f(t), \tilde{\psi}_k^{(j)}(t) \rangle. \quad (2.14)$$

As this process can be viewed as analysis of $f(t)$, $\tilde{\varphi}(t)$ and $\tilde{\psi}(t)$ are referred to as analysis scaling and wavelet functions. Similarly, due to (2.9), $\varphi(t)$ and $\psi(t)$ are referred to as synthesis scaling and wavelet functions. These four functions constitute a complete biorthogonal set, achieving a perfect reconstruction of the processed function $f(t)$. Figures showing scaling and wavelet functions and the corresponding filters of the wavelets used in this thesis can be found in [Appendix A.1](#).

2.2 Relation to FIR Filterbanks

One direct consequence of the recursion equations (2.4) and (2.7) is that a knowledge of scaling and wavelet functions is not needed for performing the wavelet analysis and synthesis of a discrete signal. This will be shown for the example of approximation coefficients, while a similar relation will analogously follow for detail coefficients. If (2.11) is substituted into (2.13), it follows:

$$x_k^{(j)} = \int f(t) \sum_n h_{0,n} \tilde{\varphi}_{2k+n}^{(j+1)}(t) dt,$$

which, after changing variables $m = 2k + n$ and interchanging the sum and the integral, can be written as:

$$x_k^{(j)} = \sum_n h_{0,m-2k} \int f(t) \tilde{\varphi}_m^{(j+1)}(t) dt.$$

The term under integration is actually the inner product $\langle f(t), \tilde{\varphi}_m^{(j+1)}(t) \rangle$, that according to (2.13) gives the approximation coefficients $x_m^{(j+1)}$. Thus, the analysis

relation that relates approximation coefficients at two consecutive scales, j and $j + 1$, is:

$$x_k^{(j)} = \sum_m h_{0,m-2k} x_m^{(j+1)}. \quad (2.15)$$

Similarly, for detail coefficients, by using (2.12) and (2.14) it follows:

$$y_k^{(j)} = \sum_m h_{1,m-2k} x_m^{(j+1)}. \quad (2.16)$$

Relations (2.15) and (2.16) completely describe the analysis stage of the DWT in a discrete setting. This can be also seen as filtering, or convolution, with FIR filters where h_0 and h_1 are the analysis low-pass and high-pass filters, respectively; performed along with downsampling by factor 2 since the filters are moved by two samples for each new output sample k . The consequence of this is that the total number of samples in the detail and the approximation signal, $y^{(j)}$ and $x^{(j)}$, is equal to the number of samples in the original signal $x^{(j+1)}$. The actual downsampling lattice can be defined by the samples that coincide with the central sample of the applied filter, with central sample of the filter being assigned with zero index. For odd symmetric filters a common practice is to index the point of symmetry with zero, while for the other filters the choice is less straightforward. Thus, in total there can be four combinations of the sampling lattices, namely: even-even (e, e), even-odd (e, o), odd-even (o, e), odd-odd (o, o); where first letter specifies the lattice for the low-pass samples, and the second for the high-pass samples. According to this, the downsampling lattice applied in (2.15) and (2.16) is (e, e). The alternative lattices can be obtained simply by re-indexing one or both of the applied filters, shifting the indices by an odd number of places.

The wavelet analysis can be performed on approximation signal of each consecutive coarser scale, which results in the dyadic wavelet decomposition, where each time the approximation signal is halved, both in the number of samples, and in its frequency bandwidth W . This is depicted in Fig. 2.1, where ($\downarrow 2$) represents downsampling by factor 2, and symbol $*$ represents the operation of convolution. The low-pass subbands are denoted with L_j , while the high-pass subbands are denoted with H_j , index j being the scale. In the following, the subbands are denoted with ς where $\varsigma \in \{L, H\}$, or equivalently, when used as an index, $\varsigma \in \{0, 1\}$.

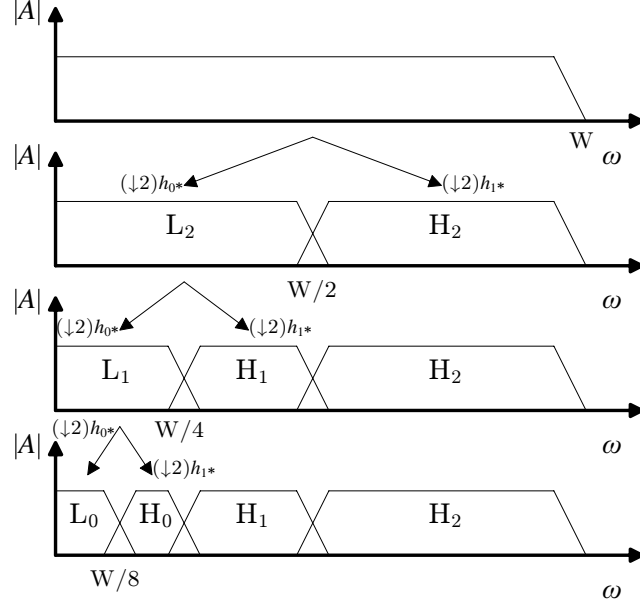


Figure 2.1: Dyadic wavelet decomposition.

By cascading the relations (2.15) and (2.16) for obtaining the relations for the multi-level decomposition it can be seen that the consecutive convolution and downsampling of several levels can be replaced by one filter that performs analysis directly to the required subband [39]. Specifically, to obtain samples of a certain subband ς at scale j , denoted with ς_j , the required filter is:

$$h_{\varsigma}^{(j)} = h_0 * (\uparrow 2)h_0 * \dots * (\uparrow 2)h_0 * (\uparrow 2)h_0 * (\uparrow 2)h_{\varsigma}, \quad (2.17)$$

where $(j_0 - j)$ is the number of filters present in the relation, and $(\uparrow 2)$ represents upsampling by factor 2. Here j_0 denotes the original signal scale, for instance $j_0 = 3$ for example in Fig. 2.1. When the filter given with (2.17) is directly applied via convolution to the original signal samples, the result has to be downsampled by factor $2^{j_0 - j}$ in order to get exactly the same approximation coefficients that would be obtained by decomposition performed level by level, where at each level downsampling by factor 2 is applied. Also, for one-level decomposition case it can be seen that $h_0^{(j_0 - 1)} \equiv h_0$ and $h_1^{(j_0 - 1)} \equiv h_1$.

The synthesis relation is obtained similarly, by substituting $\varphi_k^{(j)}(t)$ and $\psi_k^{(j)}(t)$ in (2.9) with relations (2.4) and (2.7), changing the variable k into m , and finding

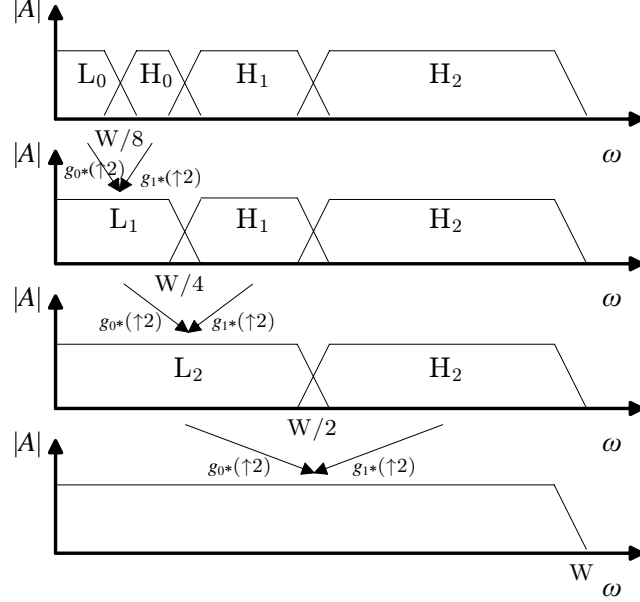


Figure 2.2: Dyadic wavelet reconstruction.

an inner product with $\tilde{\varphi}_k^{(j+1)}(t)$ at both sides, resulting in the following relation:

$$x_k^{(j+1)} = \sum_m x_m^{(j)} g_{0,k-2m} + \sum_m y_m^{(j)} g_{1,k-2m}. \quad (2.18)$$

This can be seen as upsampling of signals $x^{(j)}$ and $y^{(j)}$, followed by the filtering with FIR filters where coefficients of g_0 and g_1 as the synthesis low-pass and high-pass filters, respectively. The reconstruction procedure, that reverses the decomposition shown in Fig. 2.1 is depicted in Fig. 2.2.

Similarly to (2.17), to directly reconstruct a particular subband into the original signal scale the required filter is obtained with:

$$g_\zeta^{(j)} = g_0 * (\uparrow 2)g_0 * \dots * (\uparrow 2)g_0 * (\uparrow 2)g_0 * (\uparrow 2)g_\zeta, \quad (2.19)$$

where the notation is the same as for (2.17). The interpretation of this relation is that each sample from the subband ζ_j represents a waveform given by $g_\zeta^{(j)}$ in the reconstructed signal. This has an important relation with the distortion estimation, as will be shown in Section 4.4.

As this wavelet system constitutes a two-band filterbank, the filters it consists

of can be represented with their corresponding z -transforms, or *transfer functions*:

$$\begin{aligned}
 H_0(z) &= \sum_n h_{0,n} z^{-n}, \\
 H_1(z) &= \sum_n h_{1,n} z^{-n}, \\
 G_0(z) &= \sum_n g_{0,n} z^{-n}, \\
 G_1(z) &= \sum_n g_{1,n} z^{-n},
 \end{aligned} \tag{2.20}$$

where $z = e^{j\omega}$. Transfer functions of some of the used wavelets are shown in [Appendix A.2](#). The variant of frequency response representation using the angular frequency ω , if not obvious from a given context, is specified by the superscript ω , for example H_0^ω for the analysis low-pass filter. As in [22], here the “delay-normalised” representation of the filterbanks is adopted, meaning that the filters are centred about $n = 0$. Also, one point of clarification is needed regarding the relation (2.15) and (2.16), which, according to the definition of convolution, employs the time-reversed h_0 and h_1 . In a case that a complete notational consistency is pursued, the transfer functions $H_0(z)$ and $H_1(z)$ have to be replaced with their time-reversed counterparts $H_0(z^{-1})$ and $H_1(z^{-1})$. In this thesis the convention is adopted where this requirement is transferred to the wavelet coefficients themselves, where both analysis filters are replaced by their time-reversed counterparts. By observing (2.20), it can be seen now that alternative downsampling lattice is equivalent to introducing a delay z^{-d} into the corresponding transfer function, where d is the number of delayed samples. Generally, for changing the lattice from e to o it is sufficient to set $d = -1$ for the corresponding filter.

Now the wavelet analysis and synthesis operations can be represented with this two-band filterbank, as in graphical depiction in Fig. 2.3, where for the system to achieve perfect reconstruction (PR), $x_n^{(j)} = \hat{x}_{n-l}^{(j)}$ must be satisfied. Therefore, the output signal is identical to the input one, with the time shift of l samples. To design a PR wavelet filterbank system, the following relation has to be satisfied,

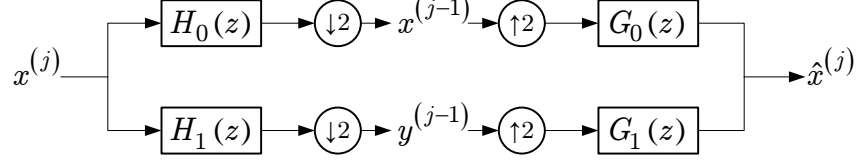


Figure 2.3: Discrete wavelet transform represented as a two-band filterbank.

which is a direct interpretation of the scheme from Fig. 2.3:

$$\begin{aligned}\hat{X}(z) &= \frac{1}{2}(H_0(z)G_0(z) + H_1(z)G_1(z))X(z) \\ &\quad + \frac{1}{2}(H_0(-z)G_0(z) + H_1(-z)G_1(z))X(-z).\end{aligned}$$

The first term is equivalent to the transfer function of the system without up-sampling and downsampling, while the second is due to aliasing created by down-sampling. In order that PR is achieved, $\hat{X}(z) = z^{-l}X(z)$, *i.e.*, the reconstructed signal is the time-delayed version of the original, the following has to be satisfied:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2z^{-l} \quad (2.21)$$

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0. \quad (2.22)$$

The condition (2.21) is known as *PR condition*, while the condition (2.22) is known as aliasing cancellation condition, or *AC condition*. The following solution satisfies the AC condition and defines the synthesis filters via the analysis ones:

$$\begin{aligned}G_0(z) &= K \cdot H_1(-z), \\ G_1(z) &= -K \cdot H_0(-z),\end{aligned} \quad (2.23)$$

where K is some constant term, that at the moment is assumed to be 1. Therefore, for $K = 1$, this relation can be equivalently written as $G_0(\omega) = H_1(\omega + (2k+1)\pi)$ and $G_1(\omega) = -H_0(\omega + (2k+1)\pi)$, for $k \in \mathbb{Z}$.

To obtain the analysis part, it is useful to simplify the PR condition as in the following. If (2.23) is substituted into (2.21), and with definition $P_0(z) = H_0(z)H_1(-z)$, we have:

$$P_0(z) - P_0(-z) = 2z^{-l}. \quad (2.24)$$

It can be seen that all even powers of z are cancelled in (2.24), and therefore the delay l is odd. With compensation for the delay, by defining $P(z) = z^l P_0(z)$, it follows:

$$P(z) + P(-z) = 2. \quad (2.25)$$

$P(z)$ is a halfband filter, so its even powers, except the constant term 1, are zero; or in other words, for symmetric filters with zero-indexed central samples every even-indexed sample of its impulse response, except the central one, is zero. One way of constructing the wavelet analysis filterbank is finding the coefficients of filter $P(z)$, and then factorising $P_0(z)$ into $H_0(z)$ and $H_1(z)$. The filter $P(z)$ itself can be designed by borrowing some of the well-known techniques for design of digital filters, the choice of which will depend on the requirements that $H_0(z)$ and $H_1(z)$ have to fulfil.

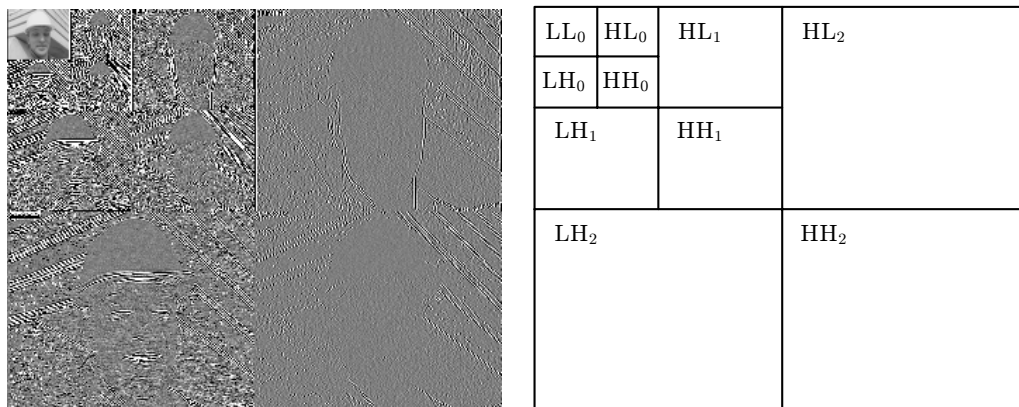
It can be checked that if the constant K has been kept without assuming its value, it would have to satisfy the following condition, which is obtained by substituting (2.23) into (2.21), and setting $z = 1$:

$$K = \frac{2}{H_0^\omega(0)H_1^\omega(\pi) - H_0^\omega(\pi)H_1^\omega(0)}. \quad (2.26)$$

The value of K can therefore be determined by the response of the transfer functions at $\omega = 0$ and $\omega = \pi$. Since for the most practical wavelet designs $H_0^\omega(\pi) = H_1^\omega(0) = 0$, and $H_0^\omega(0) = H_1^\omega(\pi) = \sqrt{2}$, the assumed value of $K = 1$ follows readily.

The notation for subbands used for two-dimensional (spatial) wavelet decomposition is introduced here. The first letter denotes the horizontal direction type of subband, while the second denotes the vertical direction type, as in the example in Fig. 2.4. For example, subband LH is obtained as a result of low-pass filtering in horizontal direction and high-pass filtering in vertical direction. Since in this case the filtering is separable, filters attributed to a particular subband can be obtained by a tensor product of the corresponding filters for horizontal and vertical directions. If ς_H denotes a type of subband in horizontal direction, and ς_V in vertical direction, and if the filters are represented in their vector and

2.3 Wavelet Filterbank Existence Conditions and Properties



(a) DWT coefficients

(b) Subband notation

Figure 2.4: Dyadic 2D-DWT of 3 levels of decomposition for the first frame of “Foreman” sequence, and the corresponding subband notation.

matrix forms, then the corresponding 2D filters can be written as:

$$\begin{aligned} \mathbf{h}_{SHSV}^{(j)} &= \mathbf{h}_{SH}^{(j)} \cdot (\mathbf{h}_{SV}^{(j)})^T, \\ \mathbf{g}_{SHSV}^{(j)} &= \mathbf{g}_{SH}^{(j)} \cdot (\mathbf{g}_{SV}^{(j)})^T. \end{aligned} \tag{2.27}$$

2.3 Wavelet Filterbank Existence Conditions and Properties

In the following, some measurable properties of wavelet filterbanks and their basic existence conditions are summarised. Only the conditions for the existence of biorthogonal filterbanks will be considered, as this type of wavelets can provide a linear phase response, *i.e.*, symmetric impulse response of filters, which is a characteristic mainly required in compression applications, and also in SVC. Note that these conditions are weaker than for the orthogonal filterbanks, and therefore also hold for them. The numerical values corresponding to some of the properties presented in this section, of the wavelets used in this thesis can be found in [Appendix A.3](#).

Condition - DC component of low-pass filters

By taking the integral of both sides in relations (2.5) and (2.11), the sum of the

2.3 Wavelet Filterbank Existence Conditions and Properties

wavelet low-pass filter coefficients is found to be:

$$\sum_n h_{0,n} = \sum_n g_{0,n} = \sqrt{2}. \quad (2.28)$$

This is the most basic necessary condition for the existence of $\varphi(t)$. This ensures that the response for the DC component is $H_0^\omega(0) = G_0^\omega(0) = \sqrt{2}$. Also, from (2.23) it follows that $H_1^\omega(\pi) = -G_1^\omega(\pi) = \sqrt{2}$.

Condition - DC component of high-pass filters

The common requirement for a high-pass wavelet filter is that it completely suppresses the DC component, *i.e.*, $H_1^\omega(0) = G_1^\omega(0) = 0$. This can be equivalently written as:

$$\sum_n h_{1,n} = \sum_n g_{1,n} = 0, \quad (2.29)$$

which is equivalent to $\int \psi(t)dt = 0$, and can be obtained from (2.8). Also, from (2.23) it follows that $H_0^\omega(\pi) = G_0^\omega(\pi) = 0$.

Condition - relation of synthesis and analysis filterbanks

From (2.23), the synthesis filter coefficients are expressed through the analysis filter coefficients:

$$\begin{aligned} g_{0,n} &= (-1)^n h_{1,n}, \\ g_{1,n} &= (-1)^{n+1} h_{0,n}. \end{aligned} \quad (2.30)$$

Since the other pair of filter can be derived from the known one, only the coefficients of analysis or synthesis filters are required in order to have a description of a complete wavelet filterbank.

Condition - sum of the even and odd terms of low-pass filter

If and only if the condition expressed with (2.29) is satisfied, by its substitution into (2.30), it follows that:

$$\sum_n h_{0,2n} = \sum_n h_{0,2n+1} = \sum_n g_{0,2n} = \sum_n g_{0,2n+1} = \frac{1}{\sqrt{2}}. \quad (2.31)$$

Property - Riesz bounds

The Riesz bounds [40] are one of the wavelet properties that measure the orthogonality of the observed filterbank. Orthogonality of filterbank can be defined

2.3 Wavelet Filterbank Existence Conditions and Properties

as how close the observed filterbank is to being orthogonal, for which the exact criteria can depend on the application. In [40] it is argued that the orthogonality measures based on energy preservation properties are the most relevant for applications related to signal approximation and compression using biorthogonal wavelets, specifically for the rate-distortion optimisation techniques. This can be interpreted in a way that the energy of a wavelet coefficient needs to be as close as possible to the energy it represents in the signal domain. Due to the linearity of the transform the same holds for the error signals, thus the orthogonality is decisive for estimation of the signal domain error that is done in the rate-distortion algorithm of the encoder. The variation of the difference between energies in transform and signal domain can be expressed with the Riesz bounds.

The signal energy E_x for some signal at scale $j + 1$ is:

$$E_x = \sum_k (x_k^{(j+1)})^2, \quad (2.32)$$

while the energy of the signal in the wavelet (transform) domain $E_{\tilde{x}}$ is:

$$E_{\tilde{x}} = \sum_k (x_k^{(j)})^2 + \sum_k (y_k^{(j)})^2, \quad (2.33)$$

where $x_k^{(j)}$ are the low-pass subband (approximation) coefficients and $y_k^{(j)}$ are the high-pass subband (detail) coefficients. The Riesz bounds are defined as two values A and B , for which:

$$AE_{\tilde{x}} \leq E_x \leq BE_{\tilde{x}}, \quad (2.34)$$

so that for the orthogonal filterbank $A = B$ ($A = B = 1$ for orthonormal), while the ratio of the bounds for non-orthogonal filterbanks is $B/A > 1$. The closer the ratio B/A is to 1, the more orthogonal the observed wavelet is.

Property - Vanishing moments

The number of vanishing moments for wavelet analysis and synthesis functions, $V_{\tilde{\psi}}$ and V_{ψ} , are defined as:

$$V_{\tilde{\psi}} = \max\left\{n : \int t^n \tilde{\psi}(t) dt = 0, n \in \mathbb{Z}^*\right\} \quad (2.35)$$

$$V_{\psi} = \max\left\{n : \int t^n \psi(t) dt = 0, n \in \mathbb{Z}^*\right\}, \quad (2.36)$$

2.3 Wavelet Filterbank Existence Conditions and Properties

or in other words, it specifies the highest degree polynomial that is cancelled by these functions. It implies that the scaling function $\varphi(t)$ reproduces all polynomials of degree lesser or equal to $n = V_{\tilde{\psi}}$, and that $V_{\tilde{\psi}} \geq V_{\psi}$. This property of polynomial cancellation applies as well to the corresponding wavelet FIR filters, thus the number of vanishing moments is the same for discrete and continuous case. This property is important for the *approximation power* of a particular wavelet [41], *i.e.*, it determines how efficient is the wavelet representation of smooth parts of the signal.

Property - L^2 norm

For orthonormal wavelets, the norms of all filters used in decomposition and reconstruction are the same and are equal to 1, while for other wavelets they generally differ. Using the notation from Section 2.2, L^2 norms of the analysis and synthesis filters are defined with:

$$\begin{aligned} \|h_{\zeta}^{(j)}\| &= \sqrt{\sum_k (h_{\zeta,k}^{(j)})^2}, \\ \|g_{\zeta}^{(j)}\| &= \sqrt{\sum_k (g_{\zeta,k}^{(j)})^2}. \end{aligned} \tag{2.37}$$

Also, due to relation between analysis and synthesis filters given with (2.30), the following L^2 norms are equal:

$$\begin{aligned} \|h_0^{(j)}\| &= \|g_1^{(j)}\| \\ \|h_1^{(j)}\| &= \|g_0^{(j)}\|. \end{aligned}$$

For the 2D DWT, since the 2D waveforms are obtained as tensor products of the equivalent 1D waveforms, from (2.27) it follows that the 2D L^2 norms are given with products of the corresponding 1D filters:

$$\begin{aligned} \|h_{\zeta_H \zeta_V}^{(j)}\| &= \|h_{\zeta_H}^{(j)}\| \cdot \|h_{\zeta_V}^{(j)}\|, \\ \|g_{\zeta_H \zeta_V}^{(j)}\| &= \|g_{\zeta_H}^{(j)}\| \cdot \|g_{\zeta_V}^{(j)}\|. \end{aligned} \tag{2.38}$$

Similarly to the Riesz bounds, the L^2 norm can be used as a measure for orthogonality. However, while the Riesz bounds represent the maximum variation of the ration of energies in signal and wavelet domain, the L^2 can be interpreted

2.3 Wavelet Filterbank Existence Conditions and Properties

as the mean expected ratio of these energies, or a ratio when the observed signal spectrum is flat, *e.g.*, in the case of the uncorrelated quantisation noise.

Property - BIBO gain

Bounded Input Bounded Output (BIBO) gain determines the amplitude range of a DWT, and is defined as a ratio between the maximum absolute value of the output samples and the maximum absolute value of the input samples. The BIBO gain of a particular subband ς_j is denoted with $BIBO(\varsigma_j)$. If the original signal has values in the interval $[-A, A]$, BIBO gain is defined with the following relation:

$$BIBO(\varsigma_j) = \sum_k |h_{\varsigma,k}^{(j)}|. \quad (2.39)$$

An incremental change in the BIBO gain is defined as:

$$\Delta BIBO(\varsigma_j) = \frac{BIBO(\varsigma_j)}{BIBO(L_j)}. \quad (2.40)$$

The BIBO gain is important for determining the precision in which coefficients are to be stored, and also can be used to detect when the values fall outside the allowed range, for instance in the capped interpolation for the purpose of prediction.

Property - Peak cyclostationary reconstruction error ratio

Cyclostationary Reconstruction Error (CSRE) is the consequence of the reconstruction error statistics modulation by the synthesis filterbank, when the quantisation error of the subband samples is reconstructed into the signal domain [42]. If the variance of the quantisation error is assumed to be equal for all subbands, and is denoted by σ^2 , the variances of the even and odd indexed samples errors in the reconstructed signal, σ_0^2 and σ_1^2 , after one level of the inverse DWT are:

$$\begin{aligned} \sigma_0^2 &= \sigma^2 \sum_k g_{0,2k}^2 + \sigma^2 \sum_k g_{1,2k}^2, \\ \sigma_1^2 &= \sigma^2 \sum_k g_{0,2k+1}^2 + \sigma^2 \sum_k g_{1,2k+1}^2. \end{aligned} \quad (2.41)$$

Here the filter coefficient indexing convention is employed in which the filter coefficients $g_{\varsigma,2k}$ coincide with the even indexed signal samples, while the filter coefficients $g_{\varsigma,2k+1}$ coincide with the odd indexed signal samples. The relation (2.41)

hence describes a cyclostationary process with the period of 2 samples, which manifests itself as a fluctuating error that for the even reconstructed samples will be of variance σ_0 while for the the odd reconstructed samples will be of variance σ_1 . The relation (2.41) is given for one-level DWT, while for multiple levels this process is of period 2^l where l is the number of decomposition levels. Then, the error variance can be described by a periodic signal $\sigma_{(l)}^2$, where $\sigma_{(l),k}^2 = \sigma_{(l),k+2^l}^2$, and is given by a recursive relation for $l \geq 1$:

$$\begin{aligned}\sigma_{(l),2m}^2 &= \sum_k \sigma_{(l-1),k+m}^2 g_{0,2k}^2 + \sigma^2 \sum_k g_{1,2k}^2, \\ \sigma_{(l),2m+1}^2 &= \sum_k \sigma_{(l-1),k+m}^2 g_{0,2k+1}^2 + \sigma^2 \sum_k g_{1,2k+1}^2,\end{aligned}\tag{2.42}$$

where $m = 0, \dots, 2^{l-1} - 1$. Since $\sigma_{(0)}^2 = \sigma^2$, this relation for $l = 1$ gives (2.41).

Here a measure of Peak CSRE Ratio (PCR) for inverse DWT of l levels is introduced, denoted with $PCR(l)$ and defined with:

$$PCR(l) = \frac{\max \sigma_{(l)}^2}{\min \sigma_{(l)}^2}.\tag{2.43}$$

It basically measures the maximum variation of the expected error in the reconstructed samples.

The CSRE and PCR measures are important for determining the amount of variation of PSNR between frames in a reconstructed sequence. This variation manifests as a visually annoying temporal artefact, especially when higher number of temporal decompositions is used, causing cyclical “b”lurring and sharpening of the reconstructed frames.

2.4 The Lifting Scheme

A common implementation of the wavelet transform is the so-called *lifting* [43; 44]. The lifting technique laid the groundwork for a second-generation of wavelet transform methods, developed to be used in situations when the conventional filter-bank implementation is not applicable, *e.g.*, when used in non-Euclidean spaces, and where translation and dilation of the wavelet functions are not defined. Using lifting, it becomes particularly easy to design transforms for other special cases

while still preserving the PR property, that would not be possible or at least be very complex in filterbank implementations, *e.g.*, for non-linear transforms, including the integer-to-integer reversible transforms. Contrasting the filterbank implementation which is based on convolution and is commonly designed in the frequency domain, as in Section 2.2, the lifting implementation can be entirely constructed in the spatial domain. As it provides a spatial domain implementation of any wavelet filterbank, lifting offers a flexible framework for exploiting local characteristics of a signal.

The lifting is performed through a ladder structure, divided into *lifting steps*. A general lifting scheme consists of four main steps, which will be described in the following. The discrete signal x of length K can be represented with vector $\mathbf{x} = [x_0, x_1, \dots, x_{K-1}]$. If its scale is specified, this is denoted as $\mathbf{x}^{(j)}$, where j is the scale. With j_0 is denoted the original, or the finest, signal scale. Signal $\mathbf{x}^{(j)}$ is first transformed using the so-called *lazy* wavelet, *i.e.*, it is split into two *polyphase* components – one comprising the even samples $\mathbf{x}_e^{(j)} := \mathbf{x}_{2k}^{(j)}$ and other the odd samples $\mathbf{x}_o^{(j)} := \mathbf{x}_{2k+1}^{(j)}$. The even signal is declared to be an *approximation signal* $\mathbf{a}^{(0)} = \mathbf{x}_e^{(j)}$, and the odd to be a *detail signal* $\mathbf{b}^{(0)} = \mathbf{x}_o^{(j)}$. The first lifting step is the *prediction step*, where the detail signal is predicted using the approximation signal, and is replaced with the prediction error:

$$\mathbf{b}^{(1)} = \mathbf{b}^{(0)} - \mathcal{P}(\mathbf{a}^{(0)}), \quad (2.44)$$

with \mathcal{P} denoting the prediction operator. The approximation signal is left unchanged $\mathbf{a}^{(1)} = \mathbf{a}^{(0)}$. The second lifting step is the *update step*, where the approximation signal is updated with the detail signal:

$$\mathbf{a}^{(2)} = \mathbf{a}^{(1)} + \mathcal{U}(\mathbf{b}^{(1)}), \quad (2.45)$$

with \mathcal{U} denoting the update operator. The detail signal is left unchanged $\mathbf{b}^{(2)} = \mathbf{b}^{(1)}$. The key property of the update step is that the average value of the processed signal is preserved. This ensures that the lowest subband, if a maximum number of decompositions is performed, consists of one coefficient that represents the DC component of the signal, scaled with some constant that depends only on the used wavelet and the number of decomposition levels. It should be noted that the order of prediction and update steps can be reversed, leading to the update-first

schemes, as in [45]. However, since the support size of the commonly used wavelets dictate the prediction-first scheme, the corresponding representation of wavelet lifting is used here. Alternatively, the first prediction step can be skipped, *i.e.*, the lifting coefficient is set to zero. The prediction and update steps can be alternately repeated L times, leading to a series of approximation and detail signals - $\mathbf{a}^{(l)}$ and $\mathbf{b}^{(l)}$ for $l = 1, \dots, L$. Depending on the wavelet filter lengths, the number of prediction steps $L_{\mathcal{P}}$ can be different to the number of update steps $L_{\mathcal{U}}$, where $L_{\mathcal{P}} + L_{\mathcal{U}} = L$. If the DC and Nyquist frequency component energies preservation is required, in a final step $\mathbf{a}^{(L)}$ and $\mathbf{b}^{(L)}$ are normalised by multiplication with the factors K_a and K_b . Consequently, approximation signal $\mathbf{x}^{(j-1)}$ and detail signal $\mathbf{y}^{(j-1)}$ on the next coarser scale are obtained.

As most of the biorthogonal wavelet filterbanks used in image and video compression consist of symmetric filters with odd number of taps, the same lifting coefficient is used for both neighbouring pixels, so the corresponding lifting relations can be summarised with the following algorithmic steps:

- Step 1: “lazy” transform

$$\begin{aligned} a_k^{(0)} &= x_{2k}^{(j)} \\ b_k^{(0)} &= x_{2k+1}^{(j)} \\ l &= 0 \end{aligned}$$

- Step 2: Prediction step (l is even)

$$\begin{aligned} b_k^{(l+1)} &= b_k^{(l)} - \mathcal{P} \left(a_k^{(l)}, a_{k+1}^{(l)} \right) = b_k^{(l)} + \lambda_{l+1} \left(a_k^{(l)} + a_{k+1}^{(l)} \right) \\ a_k^{(l+1)} &= a_k^{(l)} \\ l &= l + 1 \end{aligned}$$

- Step 3: Update step (l is odd)

$$\begin{aligned} a_k^{(l+1)} &= a_k^{(l)} + \mathcal{U} \left(b_{k-1}^{(l)}, b_k^{(l)} \right) = a_k^{(l)} + \lambda_{l+1} \left(b_{k-1}^{(l)} + b_k^{(l)} \right) \\ b_k^{(l+1)} &= b_k^{(l)} \\ l &= l + 1 \end{aligned}$$

Go back to Step 2, if required.

- Step 4: Normalisation

$$\begin{aligned} x_k^{(j-1)} &= K_a a_k^{(L)} \\ y_k^{(j-1)} &= K_b b_k^{(L)}. \end{aligned}$$

Here λ_l denotes the lifting coefficient for step l . The inverse transform is easily derived, and consists of lifting steps performed in reverse order, thus “undoing” the forward transform. A simplified lifting step for symmetric wavelets, with lifting index dropped, can be written as:

$$\tilde{x}_k = x_k + \lambda \cdot (x_{k-1} + x_{k+1}), \quad (2.46)$$

where \tilde{x}_k is a source sample, or a pixel, modified by the lifting step. It can be seen that each lifting step takes only two neighbouring pixels into account. In the case that the “lifted” pixel is at the signal boundary, *i.e.*, $k = 0$ or $k = K - 1$, the neighbouring pixel that falls outside the boundary is replaced with the neighbouring pixel from the other side. It can be shown that this is equivalent to the symmetric signal extension in the convolution implementation of the DWT, in which case the PR property of transform is preserved [22]. There are more advanced methods for signal extension, besides the trivial zero-padding or periodic, that can be adapted for DWT with non-symmetric wavelets [46].

The splitting of the signal to even and odd polyphase components is equivalent to using the (e, o) downsampling lattice. In other words, if represented with convolution, the central coefficients of the symmetric filters coincide with the pixels preserved in the downsampling – the low-pass filters are centered at the even indexed pixel positions, while the high-pass at the odd indexed pixel positions.

The lifting steps of a particular wavelet filterbank can be obtained by factoring its equivalent Laurent polynomial using the Euclidean algorithm. The fact that the lifting can be obtained by factorisation of the given filterbank can easily be recognised if z-transform representation is used:

$$\begin{bmatrix} x^{(j-1)}(z) \\ y^{(j-1)}(z) \end{bmatrix} = \begin{bmatrix} K_a & 0 \\ 0 & K_b \end{bmatrix} \cdot \prod_{l=1}^L \Lambda_l \cdot \begin{bmatrix} x_e^{(j)}(z) \\ x_o^{(j)}(z) \end{bmatrix}, \quad (2.47)$$

where:

$$\Lambda_l = \begin{cases} \begin{bmatrix} 1 & \lambda_l(z) \\ 0 & 1 \end{bmatrix} & \text{if } l \text{ is even,} \\ \begin{bmatrix} 1 & 0 \\ \lambda_l(z) & 1 \end{bmatrix} & \text{if } l \text{ is odd.} \end{cases} \quad (2.48)$$

Here $\lambda_l(z)$ has at most two terms, that correspond to the neighbouring pixels in the approximation or the detail signal.

Chapter 3

SVC Architectures

Research on scalable video coding has resulted in different approaches aiming to provide scalability functionalities, which can be classified based on their architectural aspects. Here, the term “SVC architecture” is used in the sense of the employed spatio-temporal decomposition, as it critically affects the overall implementation and determines the capabilities of an SVC system. Key differences between SVC architectures are in the structure of encoder and decoder modules; the differences in corresponding extractors are mostly irrelevant as they perform only very simple operations. Implementation of the decoder can vary depending on the device it is going to be used on. For instance, a decoder can be specialised for only a limited number of decoding points in cases where the complexity cannot be disregarded, *e.g.*, on the low-power devices. However, it can generally be said that the modules of the decoder perform operations inverse to the ones of the encoder.

Since in the SVC the key role belongs to the encoder, to gain an understanding of the particular SVC architecture it is sufficient to examine the encoder only. Therefore, its possible realisations will be analysed in the following discussion. The choice of architecture mainly depends on given requirements, *e.g.*, targeted spatio-temporal decoding points and required number of quality layers in these points. The design of a scalable video framework is also restricted by the complexity issues as well as with backward compatibility with the existing video coding systems and limitations of the chosen transform coding methods. Initially, solutions from the conventional non-scalable coding were adopted, where tempo-

ral prediction is followed by the spatial transform. Subsequently, schemes have been designed that employed temporal and spatial transforms in different ways and combined them with various prediction techniques. The architectures used in wavelet-based SVC are mainly based on *open-loop* prediction, *i.e.*, the prediction is done using video frames before they are quantised. Contrasting this, the conventional hybrid coding systems, *e.g.*, the MPEG-4 video standard, use *closed-loop* schemes which imply prediction based on previously quantised samples. When fine-granular scalability is sought, the main drawback of conventional closed-loop schemes is the frame-to-frame dependent quantisation structure. This is due to the fact that the optimal bit allocation problem for dependently quantised samples can be excessively complex [11], which inevitably leads to a performance loss.

The purpose of the spatial transform applied to a still image or a video frame is to decorrelate its content and to compact its energy into lower spatial subbands, thus facilitating compression. Besides the popular DCT that has been used in image and video compression for almost three decades, wavelet transform is a relatively recent addition to commonly used tools for video compression. In contrast to the DCT, which is applied to rectangular blocks of sizes ranging from 8×8 to 4×4 pixels, the wavelet transform is applied to the whole image. By extending the wavelet transform to the temporal axis, 3D (spatio-temporal) coding schemes were designed. Early attempts to apply 3D transforms to video sequences, and in that context filtering in the temporal direction, did not use motion information [47]. In the later schemes, a simple frame deformations were used to account for global motion, for instance in [48] a global camera pan compensation is used. Although these 3D wavelet-based methods did not achieve high coding gains, compared with algorithms that use inter-frame motion-compensated prediction techniques, they provided a framework for highly scalable video coding. In the subsequent development Motion Compensation (MC) was incorporated into the temporal wavelet transform, thus improving both coding gain and visual quality [49]. Such schemes are known by the term Motion Compensated Temporal Filtering. While the initial implementation of MCTF was based on convolution implementation of filtering, current solutions are based on the lifting implementation as it separates filtering into steps of which each can utilise compensation

with sub-pixel accuracy. To avoid confusion, it should be noted here that for the same technique some authors use the term Motion Aligned Temporal Filtering (MATF) [32].

The term “decomposition” itself refers to the sequential process of splitting the spatio-temporal subbands into two components, namely the low-pass and high-pass subbands. In the one-level decomposition, the resulting low-pass subband is at one level coarser scale than the signal corresponding to the initial subband. The number n in n -level decomposition therefore refers to the scale difference between the lowest subband produced and the original signal subband. Decomposition of one level is equivalent to the operation of analysis \mathcal{A} . The process reversing the decomposition is reconstruction, where one level reconstruction is equivalent to the operation of synthesis \mathcal{S} .

SVC architectures can be roughly divided into two groups, according to the redundancy of the employed spatio-temporal decomposition. Redundancy is a property of decomposition that is determined by the ratio of the number of samples in output subbands to the number of input signal samples. For a non-redundant decomposition this ratio equals one, while for redundant decomposition the ratio is larger than one, *i.e.*, the output of the decomposition consists of more samples than the input signal. Architectures employing the block-based transform coding generally rely on inter-scale prediction, which is a process that makes the decomposition redundant. Therefore, the architectures from this group, or so called *multi-scale pyramid* architectures, are redundant. The redundant architectures potentially can suffer from decreased quality scalability performance, depending on the efficiency of the employed prediction techniques. The non-redundant architectures generally employ a 3D spatio-temporal decomposition, in which each step is non-redundant.

As the motion-compensated decomposition steps imply non-separability of the filtering process, the order in which the spatial and temporal decompositions are performed is of crucial importance. Regarding the order in which these are applied, there are generally two basic types of non-redundant SVC architectures:

- $t+2D$ – temporal transform followed by the spatial transform, also known as spatial domain MCTF, (SD-MCTF);

3.1 Motion Compensated Temporal Filtering

- $2D+t$ – spatial transform followed by the temporal transform, also known as in-band MCTF (IB-MCTF), as it is performed in the wavelet subband domain.

Schemes that try to overcome limitations of these two architectures combine both approaches by performing several levels of pre-temporal spatial transform [50], and are commonly known as $2D+t+2D$ architectures. On the other hand, the architecture proposed in [30], drops the requirement for the perfect reconstruction property in order to improve the spatial scalability performance. Another relatively recent method is the flexible Generalised Spatio-Temporal Scalability (GSTS) [51], that unifies different non-redundant approaches into a common framework.

Since it is the main building block of the SVC architectures under consideration, MCTF is described in the following section. Then the basic SVC architectures are analysed, where only the main decomposition steps are mentioned, *i.e.*, those that are essential for a particular architecture to provide necessary spatio-temporal decoding points. In real implementations additional spatial and temporal decompositions are usually supported, providing a further compression gain. Also, entropy coding, rate-distortion optimisation and bit-stream allocation are ignored in the following discussion, as these modules usually do not influence the decomposition. All discussed architectures, except for one variant of the multi-scale pyramid architecture, are explained in the context of application of the wavelet transform in both temporal and spatial domains.

3.1 Motion Compensated Temporal Filtering

Motion compensation is an indispensable tool in video coding due to its capability to reduce temporal redundancy. It can be also viewed as a special type of temporal filtering in which compensation represents generation of the high-pass signal, and the low pass sequence is obtained by the delta low-pass filter, *i.e.*, only the temporal downsampling is performed. Since filtering without taking into account the spatial displacement between objects in subsequent video frames cannot achieve high compression, the concept of motion-compensated temporal

3.1 Motion Compensated Temporal Filtering

filtering was introduced into wavelet-based video coding techniques. MCTF in scalable video coding provides two main functions - provides a base for achieving temporal scalability and for reduction of temporal redundancy.

In the dyadic temporal decomposition the original sequence of frames, represented by vector $\mathbf{F} = [F_0, F_1, \dots, F_{N-1}]$, as in Fig. 3.1, is firstly divided into two sets. The first set contains even frames F_{2i} , while the second contains odd frames F_{2i+1} . By convention, the former become the low-pass temporal subband, while the latter become the high-pass temporal subband. The even frames are used as the reference for the motion-compensated prediction of the odd frames. Odd frames are divided into areas of different classes, depending on the prediction method:

- from both temporal directions – bidirectional or bi-inter areas;
- from just one temporal direction – unidirectional or inter areas;
- no temporal prediction at all – intra areas.

Reference frame is divided into areas that differ by the number of references provided, or in other words, by the number of connections per pixel. Non-connected areas contain pixels not connected to any of pixels in the neighbouring frame; uni-connected are those areas with one-to-one correspondence between pixels in the reference and the neighbouring frame; in multi-connected areas the correspondence is one-to-many. Further classification can be made according to the number of prediction $L_{\mathcal{P}}$ and update steps $L_{\mathcal{U}}$ in temporal lifting, relating them to the equivalent filterbanks:

- $L_{\mathcal{P}} \geq 1, L_{\mathcal{U}} \geq 1$ – for instance, Haar orthogonal 2/2 or biorthogonal 5/3 or longer filterbanks. Consecutive pairs of frames are transformed into pairs of high-pass and low-pass frames. Several approaches exist on how to treat pixels in the non-connected and multi-connected areas of the low-pass frame. For instance, a simple heuristic approach is adopted in [52], while in [53] a normalised averaging of the multi-connected pixels is selected as an optimal strategy. In [54] a relation for a globally optimal update operator is established, with the predetermined prediction operator, which is given

3.1 Motion Compensated Temporal Filtering

by the motion information. A higher coding gain can be expected using longer filters in MCTF, however the overall performance is determined by the type of motion in the processed sequence and the efficiency of the motion estimation process.

- $L_{\mathcal{P}} = 1, L_{\mathcal{U}} = 0$ – the classic MC, or MCTF using filterbanks with the delta low-pass filter. Effectively it is a $1/x$ filterbank, where x is the number of neighbouring frames used in prediction, or the length of the high-pass filter in the equivalent filterbank. Here the low-pass filtering is omitted altogether, which proves to be beneficial in the cases where the temporal averaging is likely to introduce artefacts, and also helps in lowering the decoding delay as the number of future referenced frames is reduced [55]. Some authors refer to the blocks to which this filter is applied as P-BLOCKS [56].
- $L_{\mathcal{P}} = 0, L_{\mathcal{U}} = 0$ – equivalent to introduction of intra areas in the high-pass frames - effectively a lazy “wavelet” or $1/1$ “filterbank”. As a consequence of intra-blocks in the high-pass frames, the non-connected areas exist in the low-pass frames as well. Intra areas are utilised because, due to occlusions or fast motion, some areas in the reference frame cannot be matched in the neighbouring frames. These areas are then declared as isolated and are not temporally filtered. A more detailed discussion on intra blocks is provided in Chapter 5.

These three cases are depicted in Fig. 3.1 where frame F_{2i+1} is partitioned into equal sized blocks each of which can be compensated separately. The mentioned temporal filters thus can be used globally for the whole frames, or can be chosen adaptively according to the content, where the optimal one is chosen for each frame area. A highly flexible variant of MCTF has been introduced in [55], called Unconstrained MCTF (UMCTF). In a contrast to MCTF, in UMCTF the selection of the temporal filter is adaptive, and depends on the video contents and transmission delay requirements. For example, for a highly correlated sequences with low motion activity a filtering with longer wavelets can yield increased coding gain. However, the motion compensation process cannot always capture the motion correctly, *e.g.*, in the cases of complex and irregular motion, or in presence of fast scene changes. In those cases a poor prediction is obtained and the

3.1 Motion Compensated Temporal Filtering

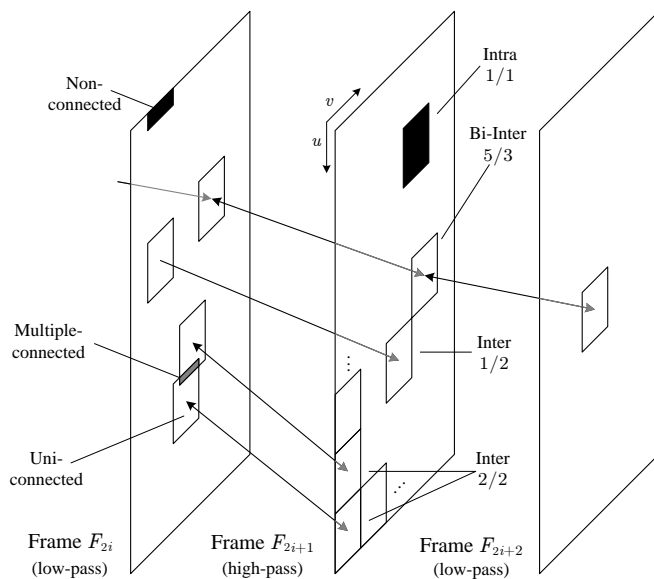


Figure 3.1: Frame areas and corresponding filterbanks.

overall compression is of lower efficiency. This also introduces visually unpleasant artefacts in the low-pass filtered frames, as some areas contain the result of filtering over poorly matched areas in neighbouring frames. These artefacts are relevant for temporal scalability, where only the temporal low-pass frames at a required level of temporal decomposition are decoded. When dyadic temporal decomposition is used, this enables extraction and decoding of a range of frame rates that are in the powers of 2 relations.

MCTF systems with integer-accurate motion compensation, using orthonormal Haar filterbank, where filtering is performed between pairs of consecutive frames F_{2i} and F_{2i+1} , can be represented with the following set of z -transforms, using notation introduced in Section 2.1:

$$\begin{aligned}
 H_0(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (1 + z_r^{-u} z_c^{-v} z_t^{-1}) \\
 H_1(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (1 - z_r^{-u} z_c^{-v} z_t^{-1}) \\
 G_0(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (1 + z_r^{-u} z_c^{-v} z_t^{-1}) z_t \\
 G_1(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (z_t^{-1} - z_r^{-u} z_c^{-v}) z_t,
 \end{aligned} \tag{3.1}$$

3.1 Motion Compensated Temporal Filtering

where z_r and z_c terms correspond to the vertical (rows) and horizontal (columns) directions while z_t corresponds to one-frame delay introduced into the overall system. The motion vector $[u, v]$, is composed of vertical (row-progression direction) u , and horizontal (column-progression direction) v displacement, where for integer-accuracy $u, v \in \mathbb{Z}$. In (3.1) u and v are functions of the sampling position in the reference frame, and one-to-one pixel mapping is assumed to avoid issues arising from the multiple-connected pixels. Here the motion vectors constitute a forward motion vector field, pointing from the current to the previous (reference) frame. While several methods exist for obtaining the motion vector field for the update lifting step [57], here “neighbour-frame-copy” inversion method is used, where $\mathcal{M}(F_{2k,m,n}) = -\mathcal{M}(F_{2k+1,m+u,n+v})$. $\mathcal{M}(\cdot)$ represents the motion vector associated with the sampling position of concern.

For motion compensation with half-pixel and higher precisions, where generally $u, v \in \mathbb{R}$, the analysis relations incorporate the spatial interpolation term $I(z_r, z_c, u, v)$:

$$\begin{aligned} H_0(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (1 + I(z_r, z_c, u, v) z_t^{-1}) \\ H_1(z_r, z_c, z_t) &= \frac{1}{\sqrt{2}} (I(z_r, z_c, \bar{u}, \bar{v}) - z_r^{-[u]} z_c^{-[v]} z_t^{-1}) \\ I(z_r, z_c, u, v) &= \sum_{i=[u]-T}^{\lfloor u \rfloor + T} \sum_{j=[v]-T}^{\lfloor v \rfloor + T} \gamma_{\bar{u}, \bar{v}}(i, j) z_r^{-i} z_c^{-j}, \end{aligned}$$

where $\bar{u} = [u] - u$, $\bar{v} = [v] - v$, $[.]$ denotes rounding to the nearest integer and T determines the number of interpolation coefficients $\gamma_{i,j}$ of the two-dimensional interpolator with support size $2T \times 2T$. Note that when $\bar{u} = \bar{v} = 0$, the interpolation is not used, so in this case $\gamma(i, j) = \delta(u, v)$ and $I(z_r, z_c, u, v) = z_r^{-u} z_c^{-v}$. Clearly, non-invertible interpolation leads to non-invertible temporal filtering, thus preventing the perfect reconstruction of the sequence. This has been a major limitation of the earlier SVC systems as it can be an important factor when good performance at high bit-rates is sought. In this case, even without the quantisation error, the peak signal-to-noise-ratio (PSNR) for two levels of temporal decomposition saturates at about 40 to 45 dB [58]. Also, the exact level of saturation of PSNR will also depend on the used interpolation filter. More recent methods have proposed solutions for this, firstly for half-pixel precision

3.1 Motion Compensated Temporal Filtering

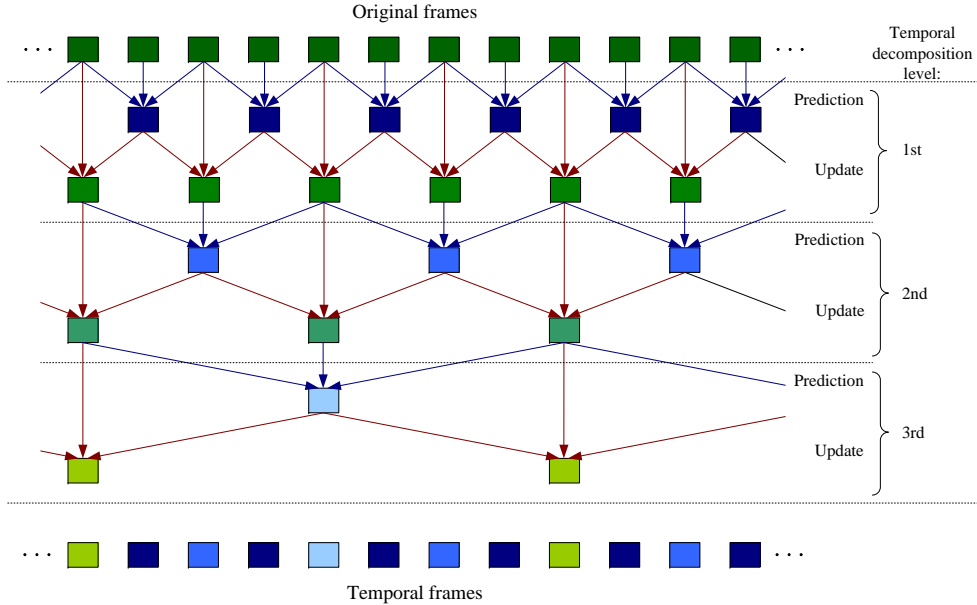


Figure 3.2: Motion compensated temporal filtering of three levels.

by incorporation of spatial interpolation in wavelet filtering [59], and finally for any precision by interleaving the interpolation and wavelet lifting steps in the temporal transform [60]. In this way, it is possible to perform compensation of any precision while maintaining the perfect reconstruction property.

One level of dyadic MCTF, *i.e.*, one level of temporal decomposition, provides temporal scalability of factor 2. The result is the temporally filtered frames, shortly called *temporal frames*. Further temporal decompositions of low-pass temporal frames provide higher levels of temporal scalability and can increase the overall compression. In Fig. 3.2 three levels of temporal transform are performed. In this example 5/3 wavelet transform is used, *i.e.*, high-pass frames are bidirectionally compensated while low-pass frames are updated from the two neighbouring high-pass frames. Each decomposition step consists of a prediction followed by an update step. Two additional decompositions are performed, each time on the set of low-pass frames obtained in the previous level of MCTF, which finally results in one low-pass frame per seven high-pass frames.

In general, original frames as well as low-pass temporal subbands or any spatial subbands can be subjected to the MCTF. Also, conventional approaches that

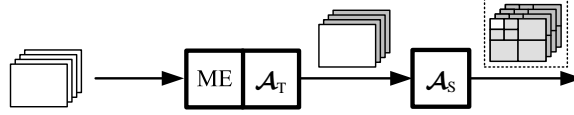


Figure 3.3: $t+2D$ SVC architecture.

use predictive compensation of frames (*e.g.*, unidirectional prediction resulting in P frames and bidirectional prediction resulting in B frames in video coding standards) can be regarded as a special case of non-dyadic MCTF without update. The key condition that enable efficient MCTF is motion estimation (ME) that provides accurate motion trajectories. In the following analysis it is assumed that rate-distortion optimisation that balances the compressed bit-rate of the motion information and the texture data is also part of the ME. And as a final note, the process that inverts MCTF is commonly referred to as IMCTF. In the usual implementation it consists of the lifting steps that invert the steps performed in the forward MCTF.

3.2 Spatial Domain MCTF

Spatial domain MCTF, or $t+2D$ architecture is the initially employed architecture in scalable video coding that uses motion aligned filtering. This architecture is also one of the basic building blocks for several other architectures. The $t+2D$ concept has evolved from the idea of a 3D transform built on top of conventional video coding where temporal prediction is followed by spatial transform.

The basic scheme of $t+2D$ architecture is illustrated in Fig. 3.3. Input video frames are firstly subjected to motion estimation and temporal filtering, represented with blocks labelled with ME and \mathcal{A}_T . In the example from Fig. 3.3, two levels of temporal decomposition are performed, *i.e.*, decomposition of four input frames results in one low-pass frame and three high pass frames, represented by frames in white and grey, respectively.

The subsequent step is the spatial transform whose main purpose in the $t+2D$ scheme is further energy compaction. In Fig. 3.3 the corresponding spatial module is labelled with \mathcal{A}_S and in this example two levels of spatial transform

are performed on all temporal frames. The spatially and temporally decomposed frames, as well as the corresponding motion information produced in ME, have to be encoded in a way that preserves spatial and temporal scalability features and provides quality scalability.

Although spatial transform in $t+2D$ architecture naturally introduces spatial scalability, in this architecture it is not fully supported as it provides only a limited decoding quality at lower spatial resolutions, the reason for which will become clear from the following. By reconstructing the sequence, *i.e.*, by inverting the decomposition path, and applying the inverse transform, a video sequence can be represented with any of the spatio-temporal decomposition subbands that were present in the encoder. Moreover, a different reconstruction path can be chosen, producing the subbands that were not present at the encoder. The downside of this approach is that it can lead to visually unpleasant artefacts in the decoded video, as the spatial and temporal transforms are generally not commutative:

$$(\downarrow 2)\mathcal{A}_S \circ (\downarrow 2)\mathcal{A}_T(\mathbf{F}) \neq (\downarrow 2)\mathcal{A}_T \circ (\downarrow 2)\mathcal{A}_S(\mathbf{F}).$$

In other words, motion vectors produced for one set of subbands in the encoder may not produce the expected results if “reused” on the different set of subbands that are available to the decoder. Commutativity is achieved only if the motion field on which the temporal transform operates is globally purely translational. This can only happen with smooth panning on stationary scenes, and when the motion vectors correspond to single pixel-to-pixel connections between the frames, as happens with integer-accurate ME. This problem is sometimes referred to as drift [61; 62]. However, to make a distinction from the error drift related to scalable predictive schemes where the reference for prediction is quantised, the term *spatio-temporal mismatch* is used in this work. The difference comes from the fact that the mismatch is still present even if the full precision wavelet coefficients of the available subbands are used in decoding. An illustrative example is shown in Fig. 3.4, where the subband enclosed in a red frame will have different values, depending on which order of applying the spatial and temporal decomposition steps is chosen. Note that other subbands are also different, however only the subband enclosed in a red frame is used for representing the sequence at half the temporal and spatial resolution. A detailed treatise on the issues related to the

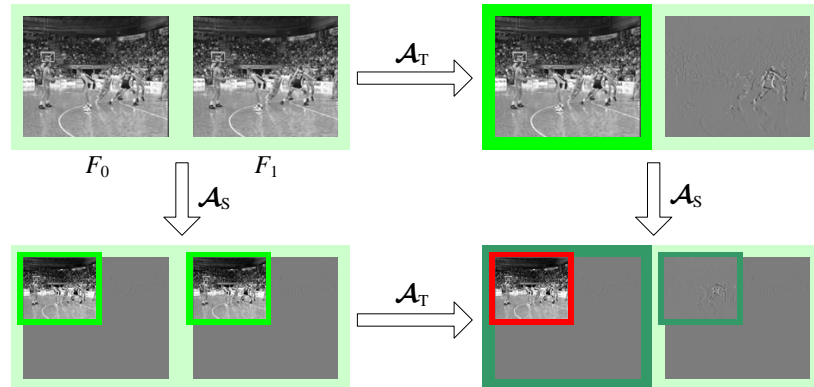


Figure 3.4: Depiction of the spatio-temporal mismatch. The result of decomposition is different depending on the chosen order of spatio-temporal decomposition steps.

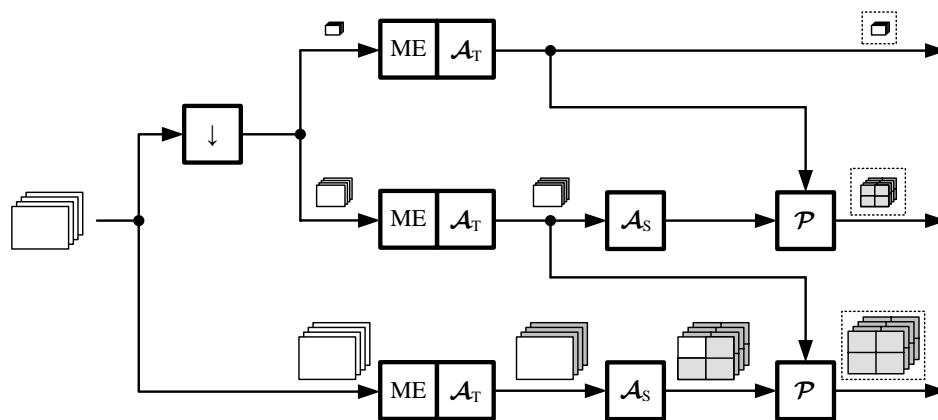
spatio-temporal mismatch can be found in [30].

For applications where only quality and temporal scalabilities are needed the $t+2D$ architecture provides the best compression performance. This is due to the fact that motion compensation provides the highest compression if applied at the original sequence resolution. Although the spatio-temporal mismatch does not allow for high quality decoding at lower spatial resolution, the possibility to decode at these resolutions still has advantages. Although some other SVC architectures, for instance $2D+t$, which will be described later, would certainly be a better choice regarding spatial scalability, the spatial scalability in $t+2D$ is nonetheless useful for applications where spatial scaling is not a functionality required for the final compressed sequence distribution, but rather for intermediate processing steps, like fast browsing or automated scene analysis.

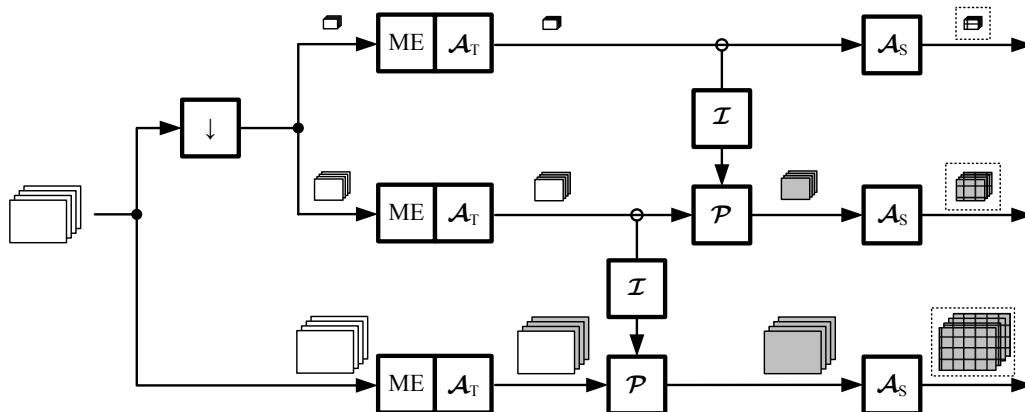
3.3 Multi-scale Pyramid Architectures

Since the main drawback of $t+2D$ architecture is poor spatial scalability performance, new approaches for improved performance at lower resolutions have been developed. Approaches that are based on the application of the $t+2D$ scheme to the video sequence represented in different spatial resolution are called multi-scale pyramid architectures. The main difference between various schemes using

3.3 Multi-scale Pyramid Architectures



(a) Prediction of low-pass spatial subbands of the temporal frames



(b) Prediction of temporal frames using interpolation

Figure 3.5: Multi-scale pyramid architectures.

multi-scale pyramid architecture is in prediction employed between $t+2D$ modules on different scales. Fig. 3.5(a) and Fig. 3.5(b) illustrate two such schemes. Input sequence is firstly spatially downsampled as many times as necessary for targeted application. In the examples in Fig. 3.5 downsampling (\downarrow) is performed twice, which corresponds to the scenarios where three different spatial resolutions are targeted.

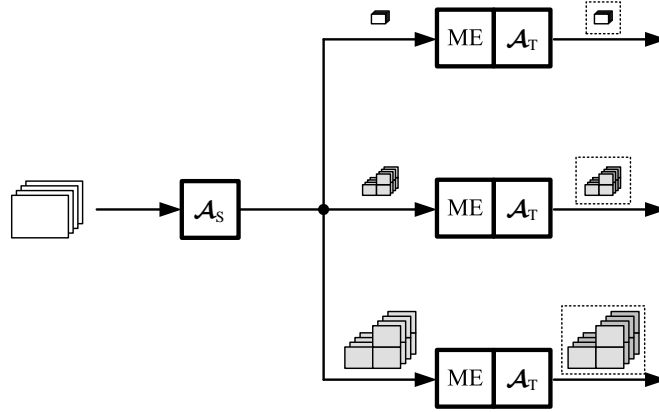
The scheme presented in Fig. 3.5(a) is known as *STool* [63] which has been designed for application in wavelet based SVC. In the *STool* scheme the temporal decomposition is applied at all targeted spatial resolutions. Since a certain degree of correlation exists between temporal frames at different resolutions, *STool*

exploits that similarity after the first level of spatial decomposition. For that reason STool uses prediction at all spatial resolution levels s such that $s < S - 1$, where S is a number of targeted spatial resolutions and $s = 0$ stands for the original sequence resolution. Specifically, low pass spatial subbands produced by one level of spatial transform of the temporal frames are predicted from the temporal frames at one level lower spatial resolution level. In Fig. 3.5(a) the prediction is represented by module labelled with the operator \mathcal{P} . Although in this illustration the STool scheme is represented as an open-loop architecture, actual prediction between spatial layers can be done from quantised, *i.e.*, decoded temporal frames at lower resolution. Since the STool applies MCTF at different resolutions of input sequence independently, it provides good spatial scalability performance.

Similarly to the architecture of STool, a multi-scale pyramid architecture has been developed for application in systems that use a block-based spatial transform. More specifically, multi-scale pyramid architecture with interpolation has been developed for scalable extension of H.264 / AVC. The scheme is the integrated part of Joint Scalable Video Model (JSVM) [29]. Fig. 3.5(b) shows the scheme used in JSVM for achieving spatial scalability. Since H.264 / AVC uses a block-based transform [64], of which the coefficients are not encoded in progression that enable multiresolution decoding, a different prediction than the one used in STool is applied. Specifically, the selected areas of lower resolution temporal frames are interpolated; in Fig. 3.5(b) this process is labelled with the operator \mathcal{I} . The actual prediction \mathcal{P} is then performed at the same spatial resolution. In this scheme only intra coded areas of temporal frames are used for interpolation and prediction. Spatial transform \mathcal{S} in this case is the block-based transform. Although in Fig. 3.5(b) the sequence at all spatial resolutions is of the same frame-rate, the scheme supports different frame rates on different spatial resolutions. Also, the prediction is usually performed from quantised values.

3.4 In-band MCTF

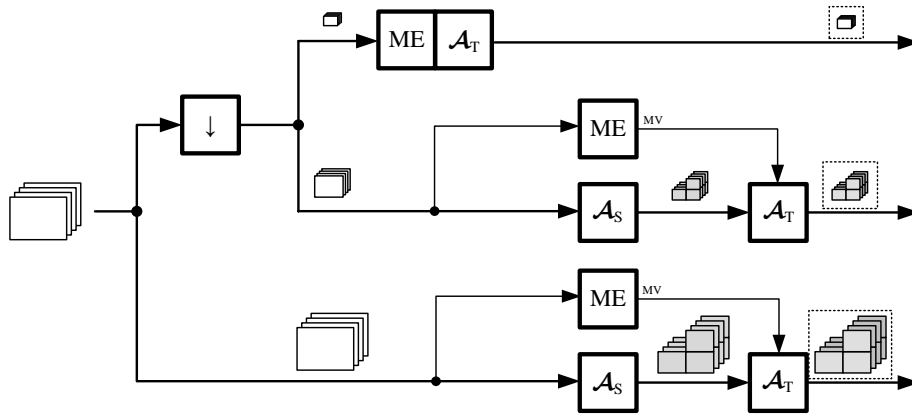
Early scalable video coding systems that addressed spatial scalability have been based on in-band MCTF, *i.e.*, $2D+t$ schemes in which the temporal transform follows the spatial transform [65]. In Fig. 3.6 a general scheme is presented.

Figure 3.6: $2D+t$ SVC architecture.

Here, spatial decomposition is performed as many times as is necessary to reach the smallest targeted spatial resolution. The low-pass spatial subband, *i.e.*, lower spatial resolution sequence, is then treated as in the $t+2D$ scenario - temporal decomposition provides temporal scalability and an arbitrary number of additional spatial transforms can be performed to enhance compression. In this scheme the MCTF is performed on the high-pass spatial subbands as well. Since the reference sequence regarding temporal scalability are obtained by temporal downsampling, the spatio-temporal mismatch can be avoided in the $2D+t$ schemes, if temporal filter that does not employ an update step is used, $L_{\mathcal{U}} = 0$.

The common characteristic of all approaches that use in-band MCTF, or in other words follow the architecture from Fig. 3.6, is that the prediction signal for MCTF on high-pass spatial subbands is obtained from the high-pass subbands of reference frames. Also the common drawback of such schemes is that ME cannot be efficiently performed on those high-pass subbands, either in a direct way or in its overcomplete representation [66], due to shift-variant property of DWT.

Instead of modelling the prediction signal entirely in the high-pass spatial subbands, the in-scale approach performs ME in the frame domain [67], whose scheme is shown in Fig. 3.7. Prior to the ME and the creation of the prediction signal, the input sequence is downsampled. Before the actual compensation, both the current frame that has to be compensated as well as the prediction signal are spatially transformed. The compensation is performed only on the high-pass

Figure 3.7: In-scale $2D+t$ SVC architecture.

temporal subbands while the low-pass subbands are discarded. However, the reconstruction is still possible if the downsampling uses the same filter banks as the spatial decomposition \mathcal{A}_S which is performed before actual compensation. The advantage of $2D+t$ schemes over multi-scale pyramid approaches is that the whole decomposition is non-redundant, while still offering good temporal decorrelation.

The problem with spatial scalability in this scheme, as well in the multi-scale pyramid scheme based on wavelets, is that the wavelet filter is used as a downsampling filter. The wavelets, in order to achieve a good compression performance, have to be near-orthogonal, which implies that their frequency characteristic is close to a halfband filter. Because of this, the attenuation at $\omega = \pi/2$ for a wavelet low-pass filter cannot be sufficient to produce a smooth aliasing-free downsampled image. On the other hand, the filters used in redundant schemes, as in the multi-scale pyramid scheme based on block transform, do not have such a constraint in the filter selection. The widely popular MPEG-B downsampling filter [28], that consists of 13 samples, has a much more favourable frequency characteristic for downsampling. The comparison of its frequency characteristics with the low-pass analysis filter of the 9/7 wavelet is shown in Fig. 3.8(a), from which it can be expected that the frames resulting from downsampling with the 9/7 wavelet will show artefacts caused by aliasing. This is confirmed with the comparison of the low-pass filtered images in Fig. 3.8(b) and Fig. 3.8(c), where

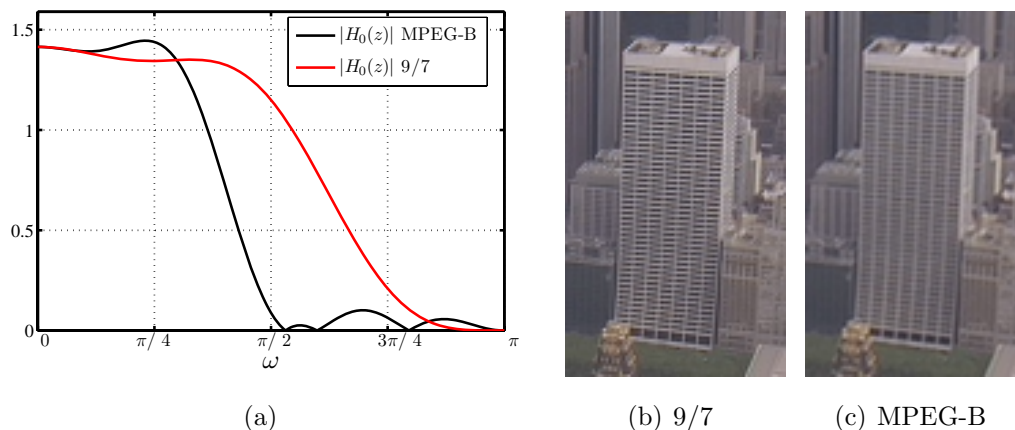


Figure 3.8: (a) Comparison of frequency characteristics of MPEG-B and analysis low-pass filter of the 9/7 wavelet filterbank. The details of the downsampled first frame of the sequence “City” are shown in (b) and (c).

the frame obtained by 9/7 wavelet contains the diagonal stripes that represent the “aliases” of high frequency contents that is present in the image of the original resolution. Not only the visual appearance of the lower resolution sequences is of importance here, but also the suitability of these for the ME process. Since the sequences suffering from aliasing artefacts will probably be harder to temporally estimate and compensate, this will also affect the compression performance.

3.5 Generalised Spatio-Temporal Scalability

Generally, there is more than just one decomposition path that can be followed to reach the required spatio-temporal resolution points. GSTS uses that fact to optimise the decomposition path with respect to the application scenario. This can help in avoiding, or at least minimising, the spatio-temporal mismatch artefacts and in saving bits that otherwise would be spent in organising the bit-stream for more spatio-temporal decoding points. As it provides the generalisation of $t+2D$ and $2D+t$ architectures, GSTS can use those two as the building blocks for obtaining more complicated decomposition paths. Due to its flexibility, GSTS is the architecture of choice in the aceSVC, and its formalisation will be presented

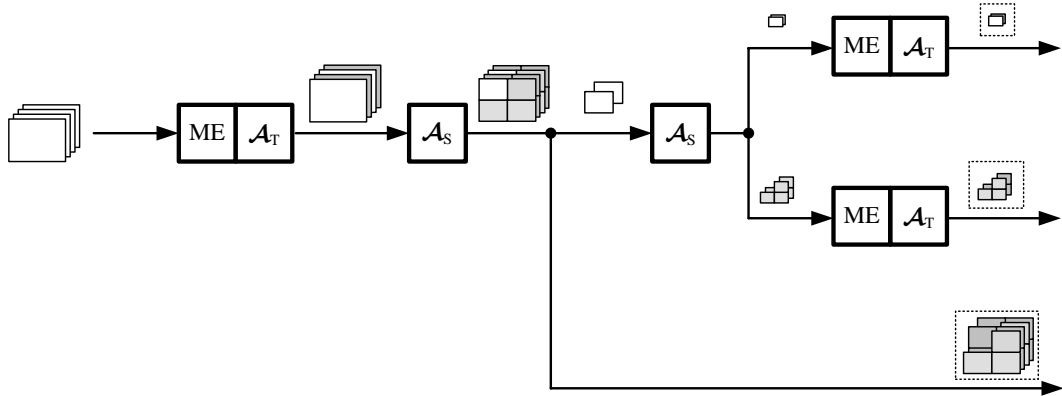


Figure 3.9: One possible GSTS decomposition path.

in the following chapter. It should be noted that, to the best of our knowledge, at this moment no other codec employs GSTS as the architecture. The conceptual description of GSTS was firstly presented in [51]; however, due to the limitations of the used SVC software platform [32], the implementation was restricted to only a few decomposition paths. The reported results were promising and indicated that by carefully choosing the decomposition path a gain in compression performance can be expected. One reported decomposition path from [51] is shown in Fig. 3.9.

Chapter 4

SVC framework for adaptive spatio-temporal decomposition

4.1 Full Scalability with Constrained Number of Layers

The scalable coding methodology and concepts on which the developed codec is based is the Structured Scalable Metaformats (SSM), which establishes a universal model for scalable bit-streams [68]. Many of these concepts have been adopted into the MPEG-21 Part 7 standard entitled Digital Item Adaptation (DIA) [69; 70]. In the SSM framework, a scalable bit-stream generally contains M nested tiers of scalability, where each tier $m = 0, 1, \dots, M - 1$ can consist of at most L_m layers. In this case the bit-stream can be logically represented in an M -dimensional space spanned by vectors representing the scalability tiers, as an M -dimensional *hypercube* of size $L_0 \times L_1 \times \dots \times L_{M-1}$, so that the maximum possible number of bit-stream segments that are determined by these layers is $\prod_{m=0}^{M-1} L_m$. Each bit-stream segment is associated with its coordinates in this space, and is denoted with $B(l_0, l_1, \dots, l_{M-1})$. These bit-stream segments are termed *atoms*, since they represent the smallest logical entity that can be added or removed from the bit-stream, using the *content-agnostic* or *format-agnostic* extractor [68; 71]. Each particular bit-stream can be regarded as a combination of atoms. Since not all combinations of B can be assumed to be decodable, this

4.1 Full Scalability with Constrained Number of Layers

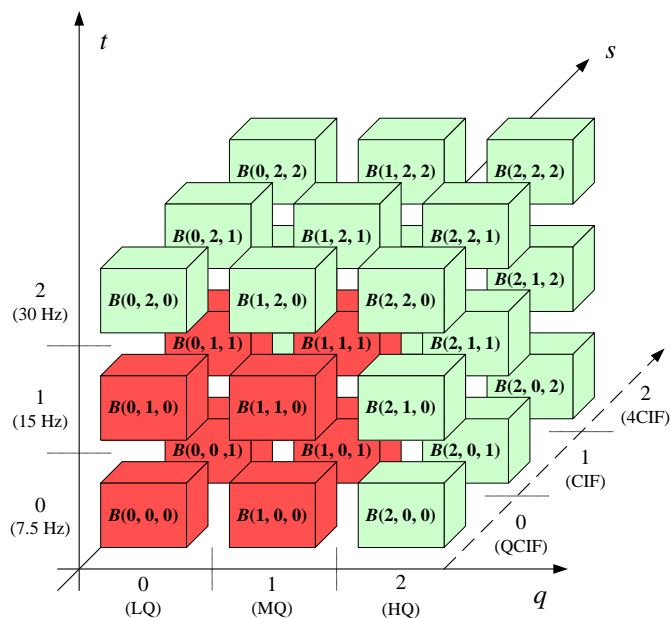
restricts the number of ways in which the bit-stream can be adapted, as well as the possible paths for multistage adaptation. Between different modes of scalability, determined by the valid combinations of atoms, the most relevant here is the *full scalability* mode. It is characterised by the *incremental tiers*, where only atoms with highest indices can be removed from the bit-stream, and where the layers are embedded, so that it is not possible to decode higher layers without having decoded the lower ones. In that case, if an M -tuple $(L_0, L_1, \dots, L_{M-1})$ represents the number of layers contained in a bit-stream, an adaptation of the bit-stream to $(L'_0, L'_1, \dots, L'_{M-1})$ can be described with:

$$\bigcup_{l_0=0, l_1=0, \dots, l_{M-1}=0}^{L'_0-1, L'_1-1, \dots, L'_{M-1}-1} B(l_0, l_1, \dots, l_{M-1}), \quad (4.1)$$

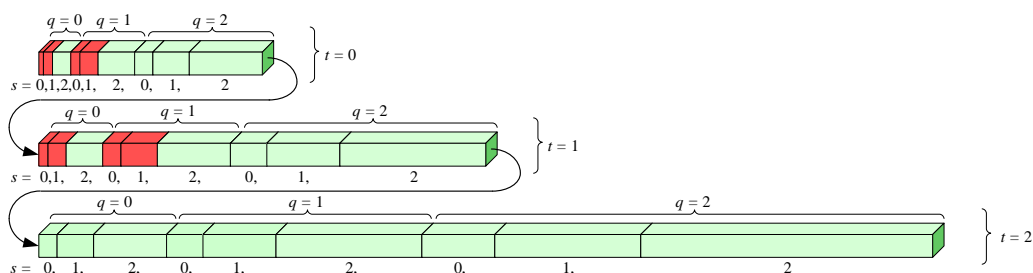
where L'_m is the number of preserved layers in the m -th tier, such that $L'_m \leq L_m$. The full bit-stream is a concatenation of these atoms in any possible progression order, along with the control data that describes content of the bit-stream and the location of each atom, either implicitly or explicitly.

The bit-stream of the scalable video coding framework presented in this thesis is organised according to a three-tier ($M = 3$) quality-spatio-temporal scalability structure, supporting these three scalability dimensions. The tiers can be represented symbolically as $m \in \{Q, S, T\}$, where Q, S and T stand for quality, spatial and temporal tier, respectively. The bit-stream is represented by the QST cube of size $L_Q \times L_S \times L_T$, as in example in Fig. 4.1(a). Each atom is conveniently denoted with $B(q, s, t)$, where $q = 0, 1, \dots, L_Q - 1$, $s = 0, 1, \dots, L_S - 1$ and $t = 0, 1, \dots, L_T - 1$. In the following, the numerical indexing will be used interchangeably with the symbolical one, in which the actual frame rates and resolution sizes are used, together with the following acronyms for the quality layers: LQ (low quality), MQ (medium quality) and HQ (high quality). These quality layers are just an exemplary case here, while in practice there can be many more layers. The bit-stream corresponding to Fig. 4.1(a), ordered by $T \rightarrow Q \rightarrow S$ progression and where realistic atom bit-lengths are used, is shown in Fig. 4.1(b). The progression of tiers is denoted with the symbol \rightarrow , such that the tier that progresses faster is on its right side. According to (4.1) the number of layers in a scalable video bit-stream is represented with triple (L_Q, L_S, L_T) , and the number

4.1 Full Scalability with Constrained Number of Layers



(a)



(b)

Figure 4.1: (a) QST cube representation of a scalable video bit-stream, $(L_Q, L_S, L_T) = (3, 3, 3)$. Red coloured atoms represent one possibility of adaptation to $(L'_Q, L'_S, L'_T) = (2, 2, 2)$. (b) The bit-stream ordered in $T \rightarrow Q \rightarrow S$ progression.

of layers in an adapted bit-stream is denoted with (L'_Q, L'_S, L'_T) .

It should be noted here that the atoms are the elementary bit-stream units only for some content-agnostic extractor, while a specialised extractor should be able to fully exploit the “embeddednes” of the texture coder and to achieve a fine-granular quality scalability with optimal performance. Such extractor can

4.1 Full Scalability with Constrained Number of Layers

truncate any atom at any desired bit location in a way that the bit-stream is still decodable, with the assumption that all the dependent atoms are available. Organising the bit-stream into layers can be advantageous when applying a rate-distortion optimisation method that targets a specific bit-rate, as it will be shown in Section 4.4. The developed extractor can work in both modes, *i.e.*, content-agnostic [71] and fine-granular.

Here a departure from the standard SSM model is made, and the full scalability mode is generalised with *constrained full scalability* mode. In this mode, the number of layers in one tier depends on the current layer index of the other tiers, *i.e.*, $L_m = f(l_0, \dots, l_{m-1}, l_{m+1}, \dots, l_{M-1})$. The rationale behind it is that in SVC not all possible combinations of atoms are of practical meaning. For instance, in the case of wide range of scalability the combination of atoms representing the video on its lowest resolution and in its highest frame rate would very probably never get utilised. The benefit of constrained full scalability mode is in savings in the control part of the bit-stream, which for large range of supported spatio-temporal resolutions and a large number of quality layers tend to cease being insignificant. To this end a set of possible spatio-temporal decoding points - *ST-points*, is defined:

$$\Omega = \bigcup_{\kappa=0}^{K-1} \{\Omega_\kappa\} = \bigcup_{\kappa=0}^{K-1} \{(s_\kappa, t_\kappa)\}, \quad (4.2)$$

where K is the total number of ST-points, and $\Omega_\kappa \equiv (s_\kappa, t_\kappa)$ represents one ST-point. On each of these points a different number of quality layers is specified, *i.e.*, $L_Q(\Omega_\kappa) \equiv L_Q(s_\kappa, t_\kappa)$, where the individual layers are denoted with $q_\kappa = 0, 1, \dots, L_Q(\Omega_\kappa) - 1$. An exemplary QST cube representation for such bit-stream is shown in Fig. 4.2. It can be seen that each atom comprises several (q, s, t) indices, and is associated with the highest indices it encompasses.

The bit-stream from Fig. 4.1 can be represented at any of $K = 9$ ST-points with 3 quality layers in each point, making for a total of 27 decoding points. On the other hand, the bitstream from Fig. 4.2 is constrained to $K = 5$ ST-points, namely: $\Omega_0 = (\text{QCIF}, 7.5\text{Hz})$, $\Omega_1 = (\text{QCIF}, 15\text{Hz})$, $\Omega_2 = (\text{CIF}, 15\text{Hz})$, $\Omega_3 = (\text{CIF}, 30\text{Hz})$ and $\Omega_4 = (4\text{CIF}, 30\text{Hz})$. The indexing of the ST-points is arbitrary, and here a $T \rightarrow S$ progression is used. At each of these ST-points

4.1 Full Scalability with Constrained Number of Layers

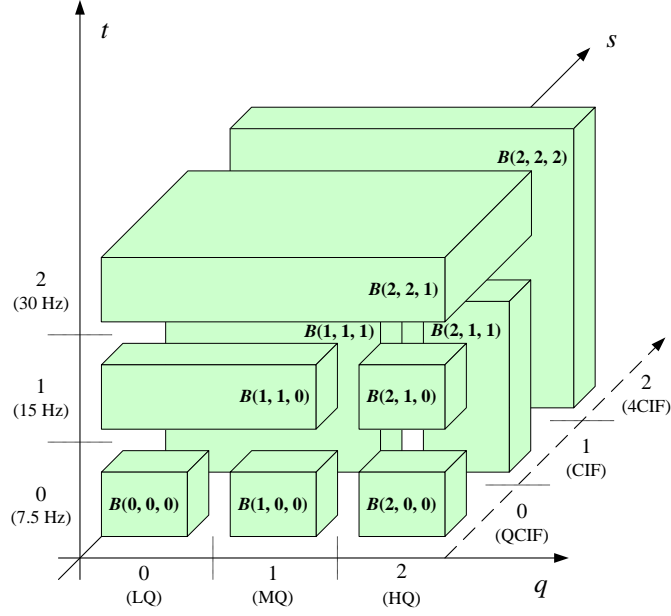


Figure 4.2: QST cube representation of a scalable video bit-stream in constrained full scalability mode.

the following quality layers are available: $q_0 \in \{\text{LQ}, \text{MQ}, \text{HQ}\}$, $q_1 \in \{\text{MQ}, \text{HQ}\}$, $q_2 \in \{\text{MQ}, \text{HQ}\}$, $q_3 \in \{\text{HQ}\}$ and $q_4 \in \{\text{HQ}\}$; and thus this makes for a total of only 9 decoding points. It can be seen that the ST-points with lower temporal and spatial indices will have to contain all the layers that appear on the ST-points with higher indices. The reason is that if a certain lower spatio-temporal resolution is contained within a higher one, then the data on that lower resolution is used in decoding of the higher one, and therefore has to contain the same quality layers, in addition to the ones defined only for that lower spatio-temporal resolution.

To visualise the set of the attainable ST-points, a projection of the QST space on the *ST-plane* can be used. The example shown in Fig. 4.3 corresponds to the QST cube in Fig. 4.2. The plane is partitioned into areas, each of which corresponds to a particular subband, or a set of subbands. A rectangular area with an ST-point in the upper-right corner and the origin of the ST-plane in the lower-left corner represents the spatio-temporal subbands encompassed within that spatio-temporal resolution. Thus, one decomposition step performed on a ST-point divides the corresponding area into two parts, with one part representing

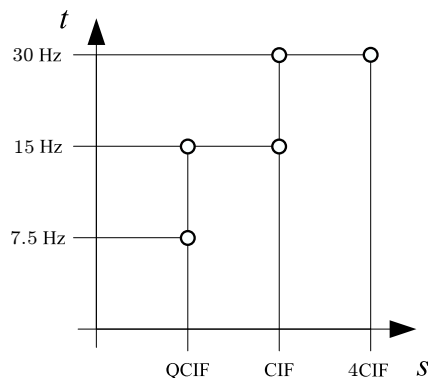


Figure 4.3: ST-plane for the QST cube in Fig. 4.2.

the low-pass and the other the high-pass subbands.

4.2 Spatio-Temporal Decomposition Tree

The main concepts introduced in the developed codec that serve to support the GSTS, introduced in Section 3.5, in constrained full scalability mode are the *spatio-temporal node* and the *spatio-temporal decomposition tree*. The spatio-temporal decomposition tree, in the following referred to as *decomposition tree*, is composed of spatio-temporal nodes, in the following referred to as *ST-node* or only *node*, each one representing a collection of subbands corresponding to the particular spatio-temporal resolution. To provide a detailed description and some constructive examples, firstly the decomposition process will be formalised and the notation introduced. An ST-node is denoted with $\mathcal{N}_{t,\varsigma T}^{s,\varsigma S}$, where s stands for spatial resolution $s = 0, 1, \dots, L_S - 1$, and t for temporal resolution $t = 0, 1, \dots, L_T - 1$; ς denotes a subband, index T indicating the temporal subband, and S the spatial subband. The subband types are defined as $\varsigma_S, \varsigma_T \in \{L, H\}$, where L stands for low-pass and H for high-pass. In some cases the three high-pass spatial subbands, HL, LH and HH, are represented with the same subband type H, as all three are processed jointly. If the set of spatio-temporal subbands contained within the node $\mathcal{N}_{t,\varsigma T}^{s,\varsigma S}$ is denoted with $\Psi_{t,\varsigma T}^{s,\varsigma S}$, and each subband is specified with

4.2 Spatio-Temporal Decomposition Tree

a pair $(\varsigma_S, \varsigma_T)$, the following ST-node subband sets are possible:

$$\begin{aligned}\Psi_{t,L}^{s,L} &= \{(LL_s, L_t)\}, \\ \Psi_{t,H}^{s,L} &= \{(LL_s, H_{t-1})\}, \\ \Psi_{t,L}^{s,H} &= \{(LH_{s-1}, L_t), (HL_{s-1}, L_t), (HH_{s-1}, L_t)\}, \\ \Psi_{t,H}^{s,H} &= \{(LH_{s-1}, H_{t-1}), (HL_{s-1}, H_{t-1}), (HH_{s-1}, H_{t-1})\},\end{aligned}$$

where the subband notation introduced in Section 2.2 is used, *i.e.*, the subscript denotes the scale. In the following, if scale is omitted, a particular subbands refers to any scale.

An ST-node can be either low-pass or high-pass. A node is a low-pass node if and only if $\varsigma_S = L$ and $\varsigma_T = L$, while all other nodes are high-pass nodes, *i.e.*, nodes containing subbands that are either temporal or spatial high-pass. Therefore, the low-pass nodes are the ones that can be used to directly display its contents, *i.e.*, its associated subband, as a reconstructed sequence. High-pass nodes can be regarded to be the detail nodes and the low-pass to be the approximation nodes.

The decomposition tree is created in the process of spatio-temporal decomposition, where each ST-node is analysed in either spatial, denoted with \mathcal{A}_S , or temporal direction, denoted with \mathcal{A}_T . The decomposition starts with the root ST-node, $\mathcal{N}_{L_{t-1},L}^{L_{s-1},L}$, which represents the sequence in its original spatio-temporal resolution. Each analysis step produces two *child* ST-nodes, one corresponding to the low-pass subband and other to the high-pass subband of the specific analysis direction. In the analysis all information represented by the current ST-node can be used, as well as information from the other ST-nodes that are available at that time instance. Using the convenient shortened notation where a subband type is dropped, and can be either of the possible two, the algorithm for decomposition can be summarised with the following set of rules:

1. $\mathcal{A}_S(\mathcal{N}_t^{s,L}) \rightarrow \{\mathcal{N}_t^{s-1,L}, \mathcal{N}_t^{s,H}\}$,
2. $\mathcal{A}_S(\mathcal{N}_t^{s,H})$ not defined,
3. $\mathcal{A}_T(\mathcal{N}_{t,L}^s) \rightarrow \{\mathcal{N}_{t-1,L}^s, \mathcal{N}_{t,H}^s\}$,

4.2 Spatio-Temporal Decomposition Tree

4. $\mathcal{A}_T(\mathcal{N}_{t,H}^s)$ not defined,
5. $\forall \mathcal{N}_t^s \mid_{s>0}, \mathcal{N}_t^{s-1} \neq \emptyset$ and $\forall \mathcal{N}_t^s \mid_{t>0}, \mathcal{N}_{t-1}^s \neq \emptyset$.

Rules 2. and 4. effectively prevent spatial decomposition of spatial high-pass ST-nodes and temporal decomposition of temporal high-pass ST-nodes. The rationale is that such cases would not amount to the new decoding points. However, if such decomposition would be beneficial for compression efficiency, a special type of *node-internal* decomposition can be defined. In that case, the subbands would be decomposed but the resulting subbands would remain in the same ST-node. Thus this process can be regarded as an internal to the node it is applied to. Rule 5. prevents creation of ST-nodes that cannot be reached on any of the reconstruction paths and thus cannot represent a decoding point. However, if further analysis of an ST-node that contradicts this rule is beneficial for compression efficiency, it can also be performed by a node-internal type of decomposition. The case that contradicts the rule 5., *i.e.*, representing an illegal decomposition path, is shown in Fig. 4.4, where in this representation of the ST-plane the arrows point to the order at which the ST-nodes are created, *i.e.*, it shows the decomposition path. The following convention is used in marking the nodes:

- white circle – low-pass ST-node, *i.e.*, an ST-node containing the low-pass spatio-temporal subband on the particular decomposition level;
- black circle – high-pass ST-node, *i.e.*, an ST-node containing high-pass spatio-temporal subbands;
- dashed circles – ST-node not present at the encoder.

In a case that further decomposition steps are performed on the high-pass ST-nodes, as in Fig. 4.4, the representation with ST-plane is difficult to interpret. To facilitate easier visualisation of the decomposition process, here the 3D representation of a decomposition tree is introduced. Two examples of such representation are shown in Fig. 4.5. The order of performing the decomposition is from top to the bottom of the tree, and for reconstruction is in the opposite direction. Each step of decomposition involves critical downsampling, so that the number of coefficients in the children ST-nodes is equal to the number of

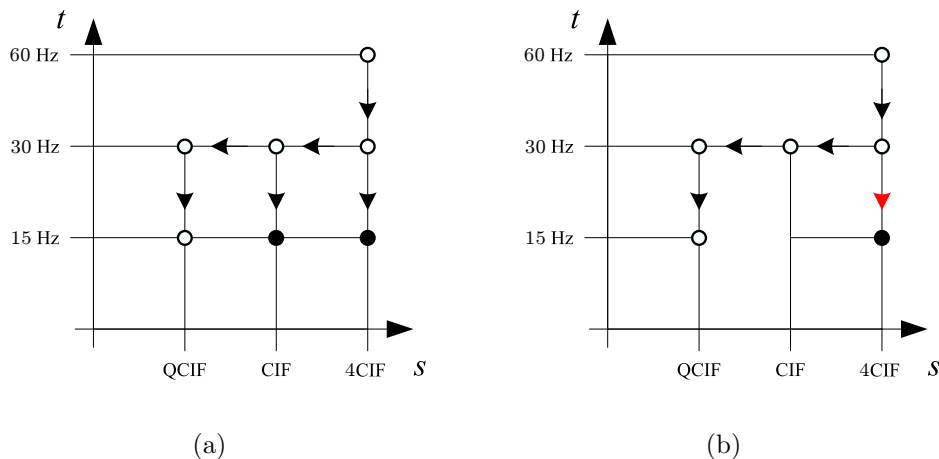


Figure 4.4: Examples of decomposition represented in an ST-plane, where $(L_S, L_T) = (3, 3)$. (a) Correct decomposition path. (b) Illegal decomposition path, where the offending decomposition step is displayed in red.

coefficients in the parent ST-node. As already defined, the dashed circles represent the nodes that can be synthesised using the available subbands, but which were not created during the analysis steps. This is depicted in Fig. 4.5(d), where the lowest temporal subbands that were created by the temporal analysis steps, denoted with \mathcal{A}_T , are spatially synthesised, denoted with \mathcal{S}_S , creating the nodes that are not present in the decomposition tree.

Leaf ST-nodes of a decomposition tree are those that do not have any child ST-nodes, or in other words, the ST-nodes that represent the final subbands given by the decomposition. The set of leaf ST-nodes is denoted with Φ , so for the example from Fig. 4.5(c) this set is $\Phi = \{\mathcal{N}_{0,L}^{0,L}, \mathcal{N}_{1,H}^{0,L}, \mathcal{N}_{1,L}^{1,H}, \mathcal{N}_{2,H}^{1,L}, \mathcal{N}_{2,L}^{2,H}\}$. Additionally, if a set of all spatio-temporal subbands associated with a particular set of ST-nodes, thus representing the subbands that exist in the corresponding bit-stream, is denoted with Θ , for the same example from Fig. 4.5(c), it follows that $\Theta = \{(LL_0, L_0), (LL_0, H_0), (LH_0, L_1), (HL_0, L_1), (HH_0, L_1), (LL_1, H_1), (HL_1, L_2), (LH_1, L_2), (HH_1, L_2)\}$.

Only one of the leaf ST-nodes is an approximation node, *i.e.*, at the same time it represents temporal and spatial low-pass, or $\varsigma_T = \varsigma_S = L$. There is a direct correspondence between leaf ST-nodes and bit-stream atoms, and is such

4.2 Spatio-Temporal Decomposition Tree

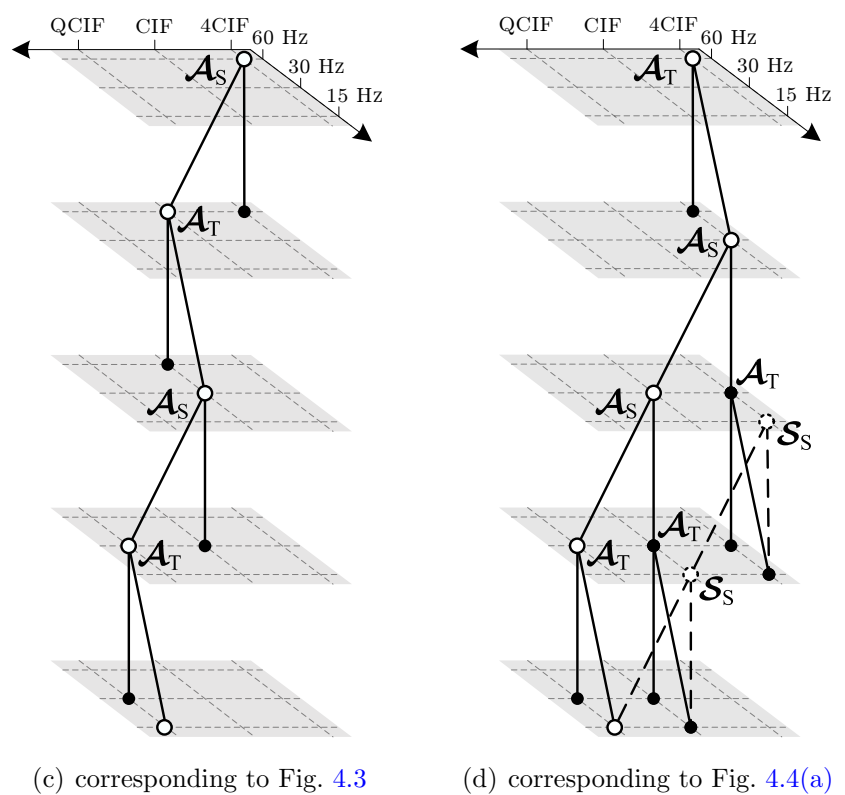
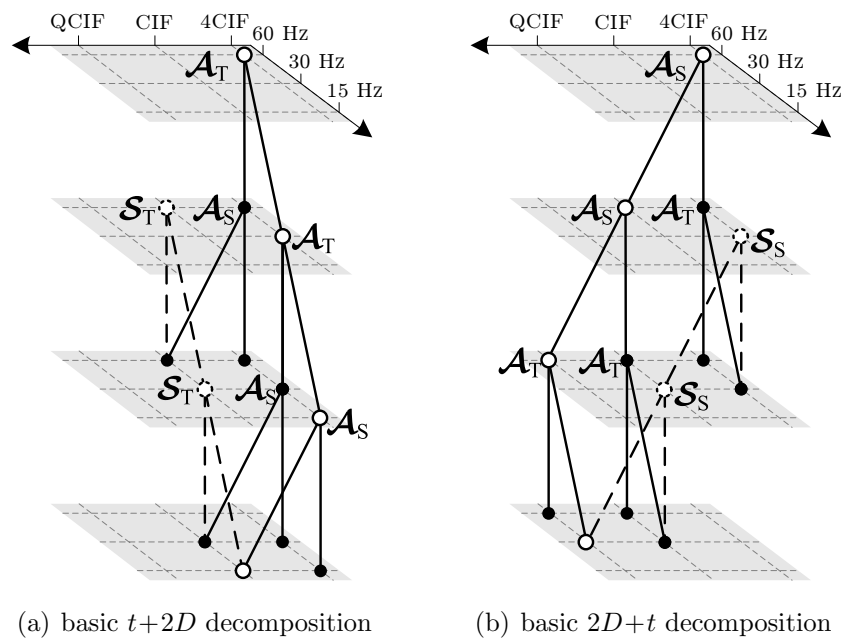


Figure 4.5: Spatio-temporal decomposition trees.

4.2 Spatio-Temporal Decomposition Tree

that the leaf ST-node $\mathcal{N}_{t,ST}^{s,SS}$ is represented in the bit-stream with atoms:

$$\bigcup_{l_q=0}^{L_Q-1} B(l_q, s, t),$$

or in other words, with all the quality layers defined for the corresponding ST-point. Hence, a direct one-to-one correspondence between ST-points and leaf ST-nodes exists, namely to decode a sequence at a particular ST-point (s_κ, t_κ) , the bit-stream atoms of the corresponding leaf ST-node $\Phi_\kappa \equiv \mathcal{N}_{t_\kappa,ST}^{s_\kappa,SS}$ are required, along with atoms from all leaf nodes for which $s \leq s_\kappa$ and $t \leq t_\kappa$.

To provide a more concise representation of a decomposition tree, the following syntax is used to denote one decomposition step:

`ς` : Decomposition step type, number of decomposition levels;

Each command specifies which subband it refers to, the type of the decomposition and the number of decomposition levels. A complete decomposition can be specified as a sequence of recursively applied decomposition commands. For example, the following specifies the decomposition needed to obtain the tree displayed in Fig. 4.5(d):

```
L : T, 1;{
  L : S, 1;{
    L : S, 1;{
      L : T, 1;
      H : T, 1;}
    H : T, 1;}}
```

The following provides an illustrative step-by-step example of decomposition represented with the spatio-temporal tree. The decomposition process can be represented by operations performed on a decomposition stack, or more precisely a Last In First Out (LIFO) queue structure. The nodes resulting from each command are placed on this stack, first the high-pass node and then the low-pass. In the following, the stack is denoted with Λ . Each subsequent command applies to the node that on the top of Λ . To demonstrate this, consider the following decomposition command, specifying the basic $t+2D$ architecture:

```
L : T, 1;{
  L : T, 1;{
    L : S, 2;
    H : S, 2;}
  H : S, 2;}
```

4.2 Spatio-Temporal Decomposition Tree

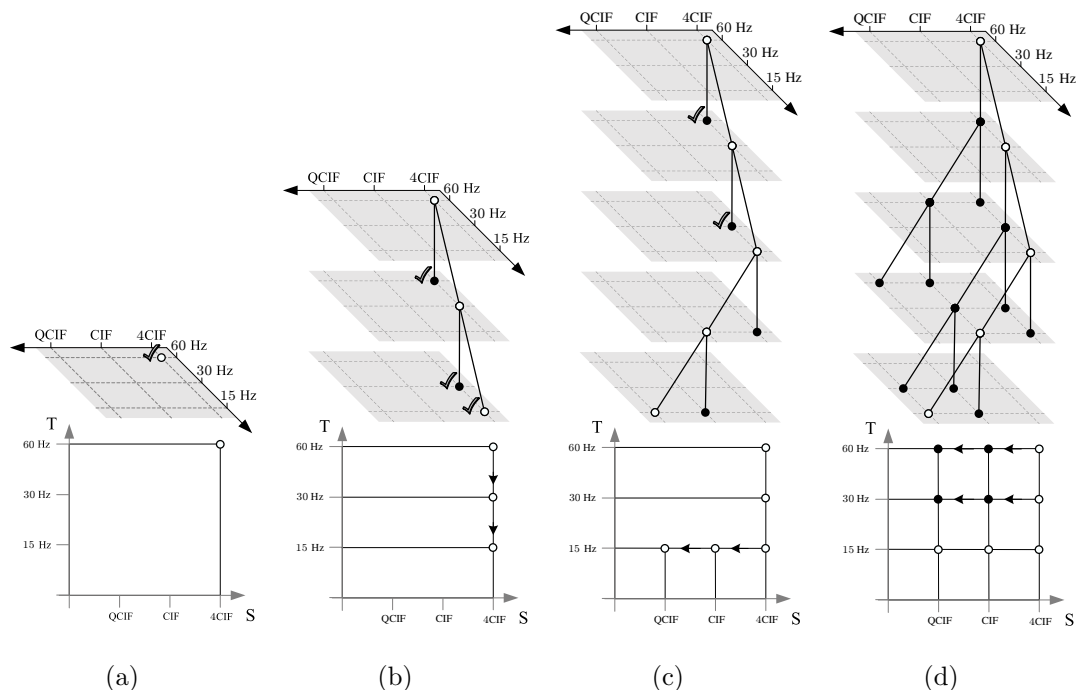


Figure 4.6: Example of the spatio-temporal decomposition corresponding to the $t+2D$ architecture.

The first two command steps perform two levels of temporal decomposition and are followed by two commands that specify two levels of spatial decomposition on the last created low-pass and high-pass nodes. The last command is applied on the high-pass subband created in the first temporal step. The whole process is illustrated in Fig. 4.6, for decomposition path corresponding to the $t+2D$ architecture, and where where $(L_S, L_T) = (3, 3)$. In the figure, the contents of the decomposition stack are marked with a check sign, and are as follows: Fig. 4.6(a), before decomposition, $\Lambda = \{\mathcal{N}_{2,L}^{2,L}\}$; Fig. 4.6(b), after the first two decomposition commands performed, $\Lambda = \{\mathcal{N}_{2,H}^{2,L}, \mathcal{N}_{1,H}^{2,L}, \mathcal{N}_{0,L}^{2,L}\}$; Fig. 4.6(c), after the third decomposition command performed, $\Lambda = \{\mathcal{N}_{2,H}^{2,L}, \mathcal{N}_{1,H}^{2,L}\}$; Fig. 4.6(d), after the last two commands have been performed on the nodes that remained in the stack, $\Lambda = \emptyset$.

Non-redundant open-loop architectures can be conveniently represented using the spatio-temporal decomposition trees. Examples illustrating possible realisa-

4.2 Spatio-Temporal Decomposition Tree

tions of spatio-temporal decomposition trees for $t+2D$ and $2D+t$ architectures are shown in Fig. 4.5(a) and Fig. 4.5(b), respectively. As it is shown in Fig. 4.5(a), in the $t+2D$ architecture the temporal decomposition is performed before the spatial one. In this particular example, where the original sequence is of 4CIF resolution and 60 Hz frame rate, temporal decomposition of two levels is performed, producing the lowest temporal subband sequence corresponding to the frame rate of 15 Hz. In the subsequent step a spatial decomposition of one level is performed, leading to the low-pass spatial subband corresponding to the CIF resolution. By reconstructing the sequence, *i.e.*, by applying the inverse transform, a video sequence can be represented on any of the spatio-temporal decomposition nodes that were visited in the encoder (white circles). Moreover, by combining the available subbands a different decomposition path can be chosen (dashed lines), producing the decoding points that were not present at the encoder (dashed circles). As discussed in Section 3.2, the drawback of this architecture is the spatio-temporal mismatch, as the nodes created at the decoder do not correspond to the ones that would be created at the encoder by choosing a direct decomposition path to that node. Even if the delta low-pass temporal filter is used, these artefacts may appear in the frames obtained by IMCTF at the resolution lower than the one at which the corresponding MVs were obtained. In the example shown in Fig. 4.5(b) corresponding to the $2D+t$ architecture, a spatial decomposition of two levels is first performed on the sequence, resulting in the low-pass subband at QCIF resolution. In the next step a temporal transform of one level applied to all spatial subbands gives the reduction of the temporal frame rate, from 60 Hz to 30 Hz.

In all the reported experiments the in-scale variant of the $2D+t$ decomposition path was used, as described in Section 3.4. Since the definition of the update step is ambiguous, only the temporal wavelets with delta low-pass filters were used. To enable in-scale $2D+t$ the required algorithmic modifications in the developed framework include buffering of the subbands prior to their spatial transform, that will be later used for ME on a particular resolution level. To give a concrete example, consider Fig. 4.5(b). If the ST-point (QCIF, 30Hz) is equivalent to $(0, 0)$, then the contents of the nodes $\mathcal{N}_{0,L}^{1,L}$ and $\mathcal{N}_{0,L}^{2,L}$ have to be buffered before the spatial transform takes place, in order they are available for the nodes $\mathcal{N}_{0,L}^{1,H}$

4.2 Spatio-Temporal Decomposition Tree

Table 4.1: Example of sets of decoding points. The ones in bold represent one set, and together with the other points they form an extended set.

Resolution	Frame rate [Hz]	Bit-rate [kbps]
QCIF	15	64 80 96 112 128
CIF	7.5	128 160 192 224 256
	15	192 224 256 320 384
	30	256 320 384 448 512
4CIF	15	512 640 768 896 1024
	30	768 896 1024 1280 1536
	60	1024 1280 1536 1792 2048

and $\mathcal{N}_{0,L}^{2,H}$ for ME and subsequent temporal transform. For the reconstruction the fact that delta-low pass filters are employed is used, so that the necessary low-pass temporal frames can be reconstructed directly by performing the inverse spatial transform on the required temporal resolution level.

The following example is used to demonstrate how the decomposition path can be optimised to mitigate the effects of the spatio-temporal mismatch. Test scenario for the sequence “City”, from [72], shown in Table 4.1, has been used. The set of decoding points is divided into two scenarios - a reduced set and an extended set, where the first is a subset of the second. If only the first set is observed, it can be seen that the corresponding spatio-temporal coordinates that cover these decoding points are $\Omega_0 = (4CIF, 60Hz)$, $\Omega_1 = (4CIF, 30Hz)$, $\Omega_2 = (CIF, 30Hz)$ and $\Omega_3 = (QCIF, 15Hz)$. The decomposition paths involving the least number of steps, and that connect these points are presented in Fig. 4.7.

For this particular example, two decomposition paths are possible. A preferred decomposition direction, can be set in the encoder. A criterion can be, for instance, minimising the visual artefacts caused by the spatio-temporal mismatch. Therefore, a path selection algorithm that is driven by some cost metric can be utilised to find an optimal one of all possible paths. For the extended set of condition points the corresponding decomposition paths are shown in Fig. 4.8. In Fig. 4.8(a) the spatial direction is preferred to the temporal, while in Fig. 4.8(b)

4.2 Spatio-Temporal Decomposition Tree

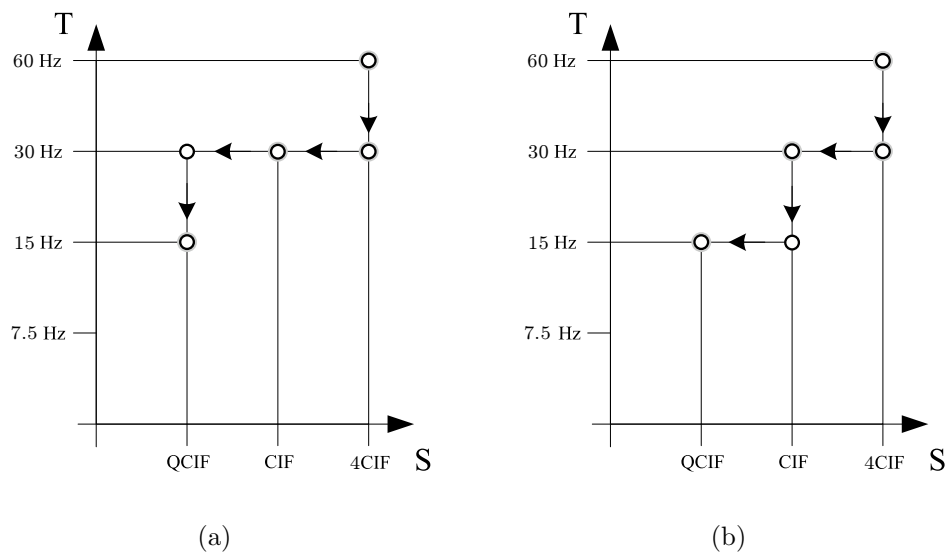


Figure 4.7: Two possible paths to support the reduced test set from Table 4.1. The spatio-temporal nodes corresponding to the set of specified decoding points are marked with a grey outer circle.

the case is the opposite.

The decomposition trees for these two cases are shown in Fig. 4.9, while the decomposition commands are displayed in Table 4.2.

4.2 Spatio-Temporal Decomposition Tree

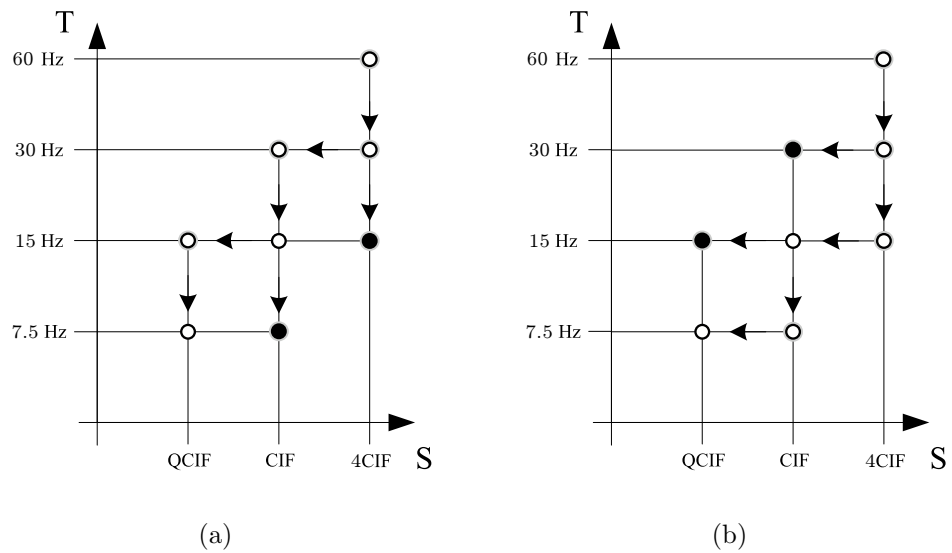


Figure 4.8: The decomposition paths for the extended set of condition points. Nodes containing high-pass subbands are shown in grey. (a) Spatial direction preferred. (b) Temporal direction preferred.

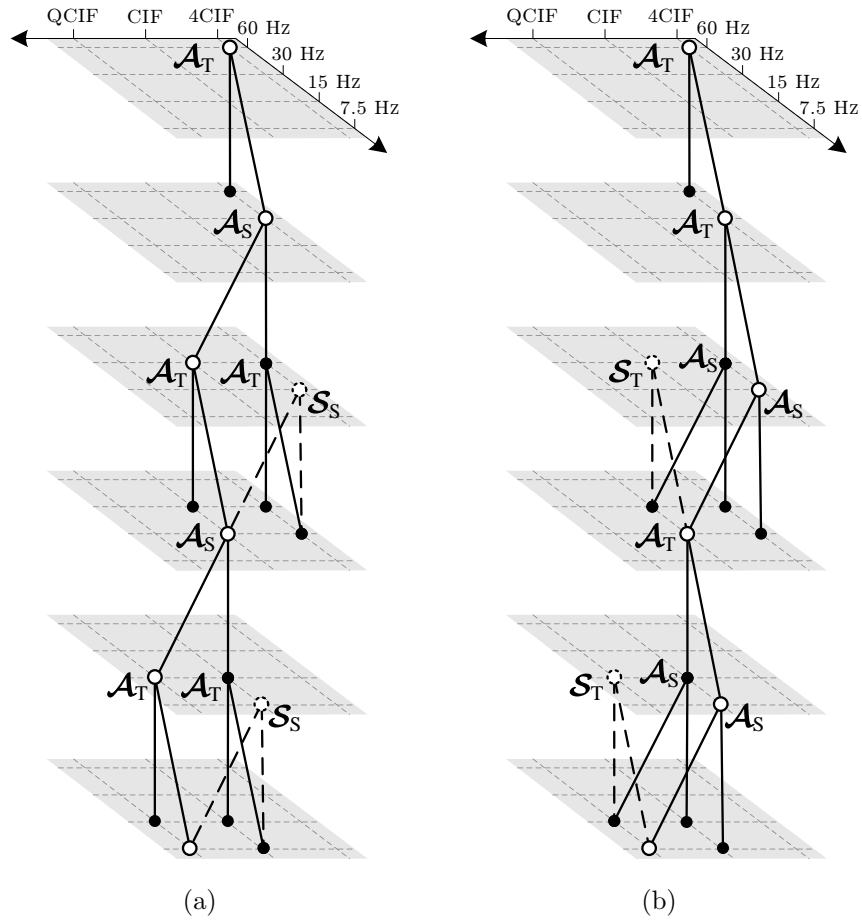


Figure 4.9: Decomposition trees for examples from Fig. 4.8. (a) Fig. 4.8(a). (b) Fig. 4.8(b).

4.3 Comparison of Architectures Using Decomposition Trees

Table 4.2: Decomposition commands.

Fig. 4.7(a)	Fig. 4.7(b)	Fig. 4.8(a)	Fig. 4.8(b)
<pre>L : T, 1;{ L : S, 1;{ L : S, 1;{ L : T, 1;} } } }</pre>	<pre>L : T, 1;{ L : S, 1;{ L : T, 1;{ L : S, 1;} } } }</pre>	<pre>L : T, 1;{ L : S, 1;{ L : T, 1;{ L : S, 1;{ L : T, 1;} } } H : T, 1;} } H : T, 1;} }</pre>	<pre>L : T, 1;{ L : T, 1;{ L : S, 1;{ L : T, 1;{ L : S, 1;} } } H : S, 1;} } H : S, 1;} }</pre>

4.3 Comparison of Architectures Using Decomposition Trees

To demonstrate the usefulness of the GSTS scheme, the experiment of comparison with $t+2D$ and $2D+t$ was conducted. Two sequences, “Football” and “Soccer”, were compressed once and decoding was performed at three ST-points, $\Omega = \{(CIF, 15Hz), (CIF, 7.5Hz), (QCIF, 7.5Hz)\}$, in the following corresponding bit-rates:

$$\begin{aligned}
 q_0 &\in \{384\text{kbps}, 448\text{kbps}, 512\text{kbps}, 640\text{kbps}\}, \\
 q_1 &\in \{192\text{kbps}, 280\text{kbps}, 384\text{kbps}, 448\text{kbps}\}, \\
 q_2 &\in \{96\text{kbps}, 128\text{kbps}, 160\text{kbps}, 192\text{kbps}\}
 \end{aligned}$$

The decomposition commands that were used are displayed in Table 4.3. Note that the decomposition path for $2D+t$ architecture specifies just one level of spatial decomposition before the temporal decomposition, while the additional spatial decomposition steps are performed only after the temporal ones. This is done to maximise the compression performance, as only one pre-temporal spatial decomposition level is required to cover the target ST-points. In all three cases, wavelets 1/3 and 9/7 were used in temporal and spatial decomposition, respectively. The detailed results of the experiment can be found in Fig. 4.10

4.3 Comparison of Architectures Using Decomposition Trees

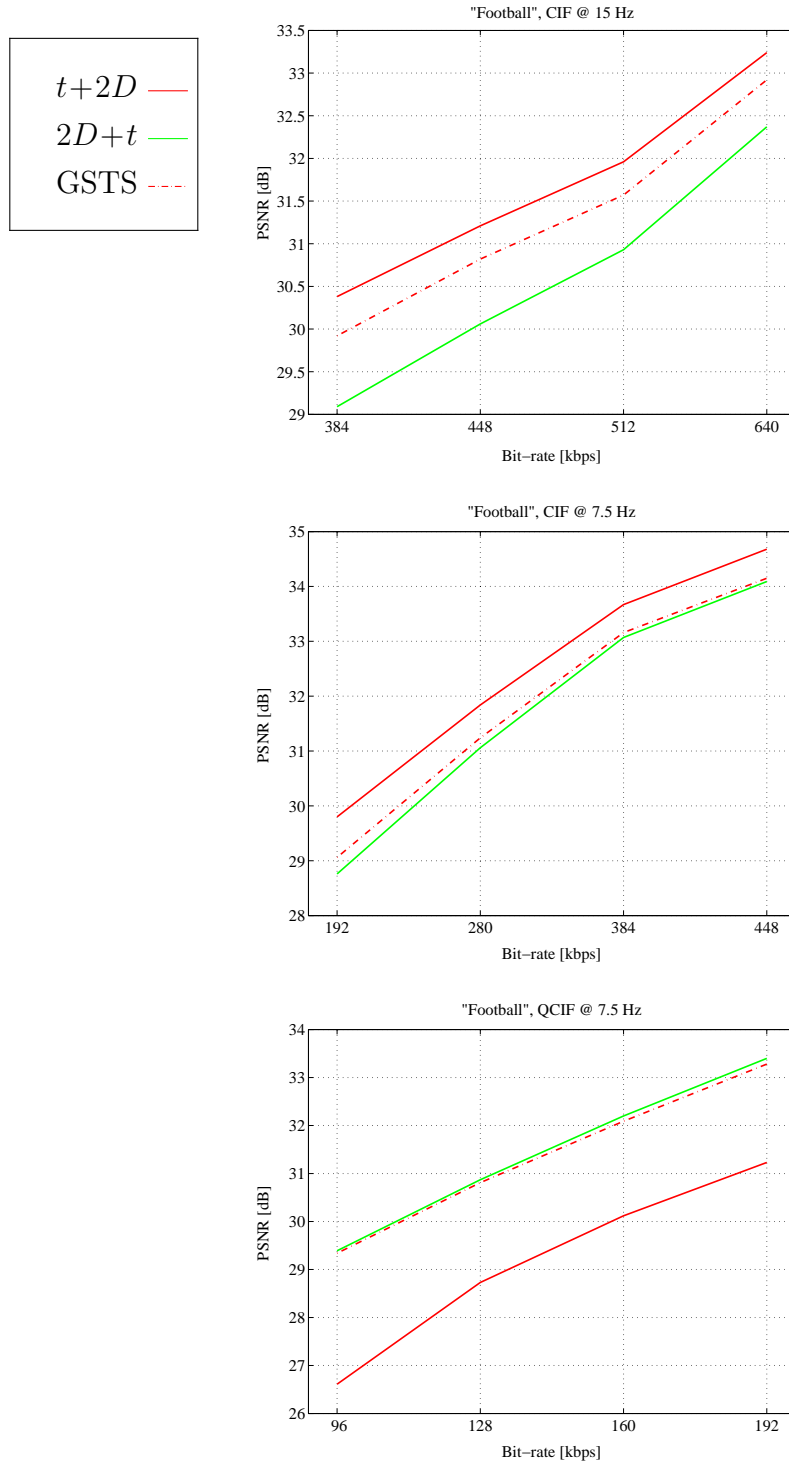


Figure 4.10: Results for different decompositions for test sequence "Football".

4.3 Comparison of Architectures Using Decomposition Trees

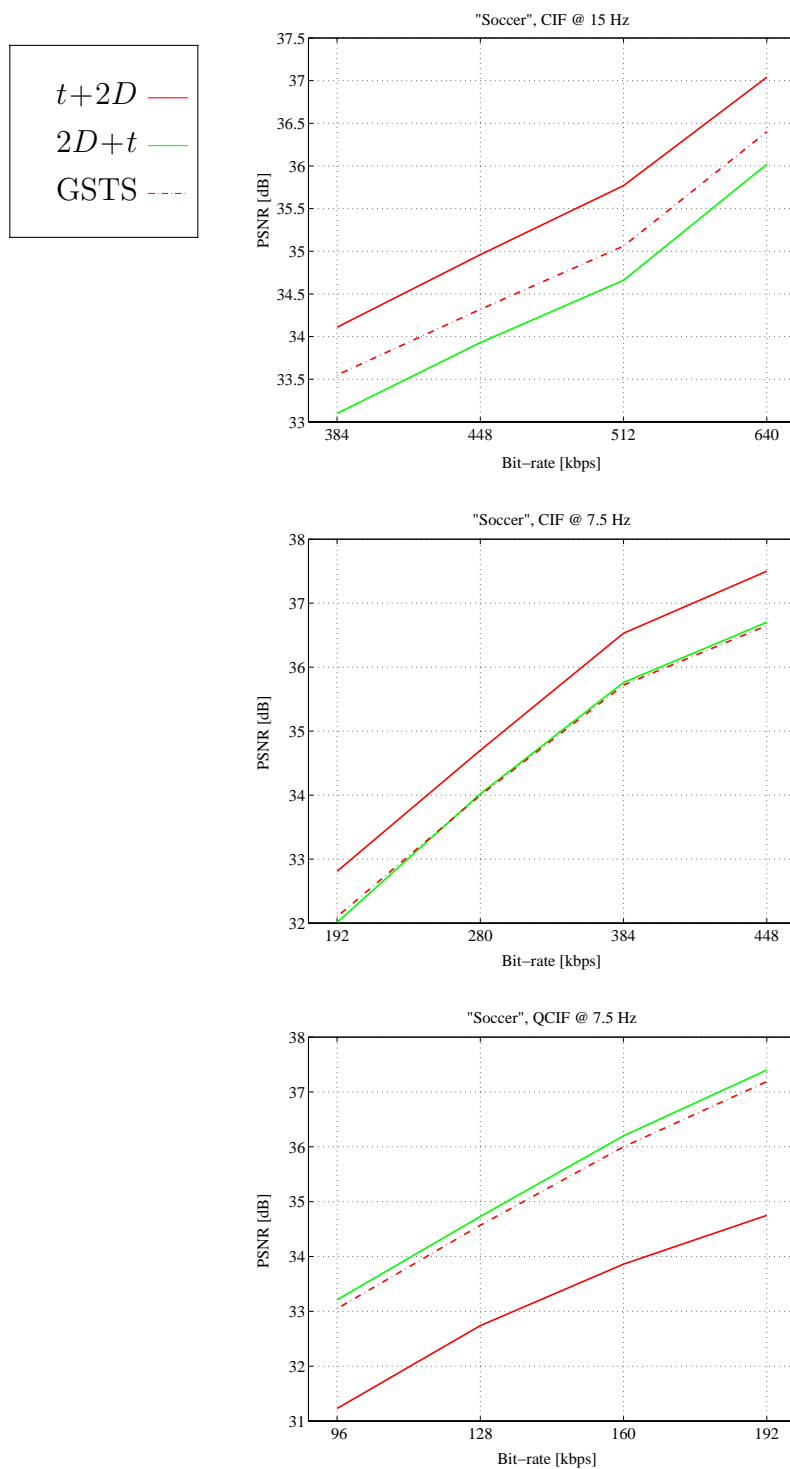


Figure 4.11: Results for different decompositions for test sequence "Soccer".

4.4 Post-Compression Rate-Distortion Optimisation

Rate-distortion optimisation is employed to achieve an optimal decoding performance at some target bit-rate, or conversely, to achieve a target distortion with a minimum number of bits. Rate-distortion optimisation techniques for wavelet based coding can be classified into two distinct groups. In the first are those that involve exhaustive search over a subset of possible combinations of quantisation parameters, the example being the Space-Frequency Quantization (SFQ) algorithm [73] and Joint Space-Frequency Segmentation using balanced wavelet packet trees for least-cost image representation [74]. However, the gain obtained with such solutions comes with the cost of significant computational complexity, and as the result of optimising for a certain decoding point the scalability property is often sacrificed or its performance is deteriorated [75]. Techniques from the second group offer a decrease in complexity by performing the distortion optimised bit-stream allocation only after the bit-stream has been compressed, commonly named Post-Compression Rate-Distortion (PCRD) optimisation techniques. An example is the EBCOT algorithm, which is an underlying algorithm in the JPEG-2000 standard for coding of still images [27]. This chapter introduces a PCRD technique applied to EZBC [24], which is the entropy coding algorithm employed in the developed scalable codec.

EZBC algorithm performs bit-plane coding of wavelet coefficients, which can be understood as embedded quantisation, where a series of progressively smaller quantisation steps is applied. Although the probability density function of wavelet coefficients is most accurately modelled with the Laplacian probability distribution function (pdf) [76; 77], within one quantisation step the distribution is nearly uniform, for small enough quantisation steps. Since using this assumption for all quantisation steps the deterioration in performance can be expected to be only negligible [78], this simplification is used in the rest of the discussion. Experiments devised to test this assumption for validity, of which the results are reported in the following, also support it. Although in the PCRD the exact distortion of each individual coefficient could be computed during encoding, this would involve a several additional floating point arithmetic operations per visited

4.4 Post-Compression Rate-Distortion Optimisation

significant coefficient. To reduce the complexity of the distortion estimation, a simplified statistical model is utilised, described in the following.

In bit-plane b the coefficients can be divided into three sets: I_b^1 are the coefficients that have the most significant bit (MSB) in the b -th bit-plane, I_b^0 are those that have MSB below the b -th bit-plane, *i.e.*, they remain insignificant, while S_b are those that have MSB above the b -th bit-plane, *i.e.*, they have been already found to be significant in the previous bit-planes. Regarding the EZBC algorithm, I_b^1 and I_b^0 correspond to the contents of the List of Insignificant Nodes (LIN), while S_b corresponds to the contents of the List of Significant Pixels (LSP), [24]. The algorithm performs bit-plane coding, which is effectively an *embedded quantisation* with a central dead-zone around zero, such that during encoding of the bit-plane b the coefficients below threshold $T_b = 2^b$ are quantised to zero, as shown in the example in Fig. 4.12. $Q_{b,i}$ represent the reconstruction values, where index i represents the i -th uncertainty interval. The quantisation step in the bit-plane b is q_b , and since uniform quantisation is employed, $q_{b-1} = q_b/2$. The fact that the pdf of coefficients is not uniform can be used on the decoding side, where inverse quantisation can reconstruct coefficient values to $Q_{b,i}$ that are determined by the estimated or transmitted pdf parameters. In [79] gains of up to 0.4 dB were reported for embedded coding of still images using pdf-adaptive reconstruction.

Using the previously introduced notation, the expected distortion contributed to the quantisation error of one wavelet coefficient, before the coding of the bit-plane b has commenced, can be expressed as:

$$\begin{aligned}
 \delta_{I_b^0} &= \frac{1}{q_b} \int_0^{q_b} c^2 dc = \frac{1}{3} q_b^2, \\
 \delta_{I_b^1} &= \frac{1}{q_b} \int_{q_b}^{2q_b} c^2 dc = \frac{7}{3} q_b^2, \\
 \delta_{S_b} &= \frac{1}{2q_b} \int_{-q_b}^{q_b} x^2 dx = \frac{1}{3} q_b^2,
 \end{aligned} \tag{4.3}$$

where $\delta_{I_b^0}$, $\delta_{I_b^1}$ and δ_{S_b} are distortion contributions per one coefficient from the sets I_b^0 , I_b^1 and S_b , respectively. Here variable x represents the quantisation error of a coefficient, while variable c represents its absolute value. In the computation

4.4 Post-Compression Rate-Distortion Optimisation

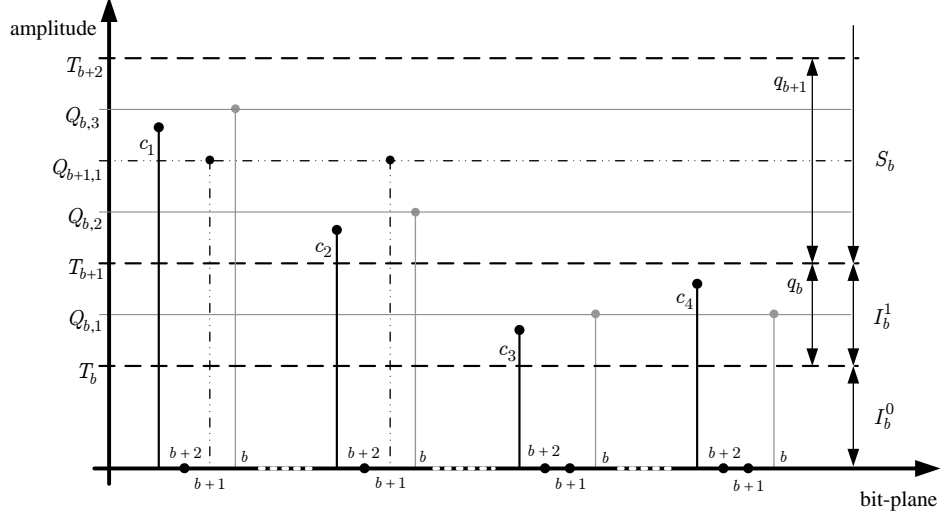


Figure 4.12: Embedded quantisation for an exemplary set of coefficients $\{c_1, c_2, c_3, c_4\}$.

of $\delta_{I_b^0}$ the probability distribution of the next lower bit-plane encoding pass is assumed to be unknown. The overall estimated distortion at the beginning of the b -th bit-plane coding pass is then:

$$\hat{D}_b = N(I_b^0)\delta_{I_b^0} + N(I_b^1)\delta_{I_b^1} + N(S_b)\delta_{S_b},$$

where $N(\cdot)$ represents the number of elements in the concerned set. Here \hat{D} stands for estimated distortion, while the real distortion is denoted with D . In the next lower bit-plane $b-1$, the set of significant coefficients is enlarged with the newly found significant coefficients $S_{b-1} = S_b \cup I_b^1$, *i.e.*, the number of significant coefficients has increased by $N(I_b^1)$. Also, it can be shown that $N(I_b^0)\delta_{I_b^0} = N(I_{b-1}^0)\delta_{I_{b-1}^0} + N(I_{b-1}^1)\delta_{I_{b-1}^1}$. This result readily follows for the computation of $\delta_{I_b^0}$ as in (4.3), which implicitly uses $N(I_{b-1}^1) = N(I_{b-1}^0) = N(I_b^0)/2$. Then it can be seen that the reduction of expected distortion after the b -th bit-plane has been encoded is determined with $\Delta\hat{D}_b = \hat{D}_b - \hat{D}_{b-1}$, which translates into:

$$\begin{aligned} \Delta\hat{D}_b &= N(I_b^1) \cdot (\delta_{I_b^1} - \delta_{S_{b-1}}) + N(S_b) \cdot (\delta_{S_b} - \delta_{S_{b-1}}) \\ &= \gamma_I N(I_b^1) q_b^2 + \gamma_S N(S_b) q_b^2, \end{aligned}$$

4.4 Post-Compression Rate-Distortion Optimisation

where $\gamma_I = 2.25$ and $\gamma_S = 0.25$. This result can be split into two terms, as the corresponding two sets are processed independently:

$$\begin{aligned}\Delta\hat{D}_{b,I} &= \gamma_I N(I_b^1) q_b^2 \\ \Delta\hat{D}_{b,S} &= \gamma_S N(S_b) q_b^2.\end{aligned}\tag{4.4}$$

The first term, $\Delta\hat{D}_{b,I}$, originates from the reduction in distortion caused by insignificant coefficients becoming significant in the b -th bit-plane, while the second term $\Delta\hat{D}_{b,S}$ is a consequence of encoding the b -th bit of significant coefficients. The fact that $\gamma_I > \gamma_S$ justifies the order of processing of the coefficients in most of the wavelet coders based on efficient encoding of significant and insignificant coefficients lists [27; 80], where it is assumed that encoding of the newly found significant coefficients conveys more information than encoding of the current bit-plane of the already significant ones.

In EZBC, one bit-plane is divided into *fractional* bit-planes that correspond to the levels of quadtrees representing blocks of wavelet coefficients in one subband. The number of fractional bit-planes depends on the related subband dimensions, and are denoted with $f = 0, \dots, f_\varsigma - 1$, where ς stands for a subband, and f_ς for the number of fractional bit-planes in that subband. Here $f = 0$ specifies the first processed fractional bit-plane, *i.e.*, the lowest quadtree level. The order of processing is from the lowest level to the highest as it has been found that it is optimal regarding the amount of bits spent per found significant coefficient. Since the parent subbands are not used in the context modelling for the entropy coder, all fractional bit-planes can be processed independently. Hence, the set of significant coefficients I_b^1 can be split into the subsets $I_{\varsigma,b,f}^1$, such that:

$$I_b^1 = \bigcup_{\varsigma \in \Theta} \left(\bigcup_{f=0}^{f_\varsigma} I_{\varsigma,b,f}^1 \right),$$

where Θ is a set of all spatio-temporal subbands, as defined previously. Thus, each $I_{\varsigma,b,f}^1$ set contains insignificant coefficients in subband ς belonging to a particular quadtree level f for bit-plane b . Similarly, the set S_b can be divided into independent sets, one for each subband, which is denoted with $S_{\varsigma,b}$. Since this set is encoded last in the bit-plane, it can be associated with an additional fractional bit-plane $f = f_\varsigma$, thus making the total number of fractional bit-planes $f_\varsigma + 1$.

4.4 Post-Compression Rate-Distortion Optimisation

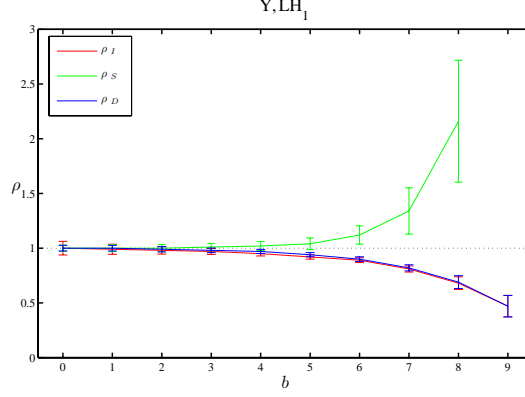


Figure 4.13: Distortion estimation accuracy for sequence “City”, 4CIF, 60 Hz, intra-only coding, 5 levels of spatial decomposition. Results for the subband LH_1 are displayed, averaged over 600 frames, with the confidence interval of two standard deviations indicated.

To examine the influence of non-uniform distribution of the wavelet coefficients within the quantisation steps, the estimated distortion \hat{D} in particular subband is compared here to the actual distortion D . For that purpose the following ratios are defined:

$$\begin{aligned}
 \rho_I &= \Delta D_{b,I} / \Delta \hat{D}_{b,I} \\
 \rho_S &= \Delta D_{b,S} / \Delta \hat{D}_{b,S} \\
 \rho_D &= \Delta D_b / \Delta \hat{D}_b.
 \end{aligned} \tag{4.5}$$

These ratios have been measured for two sequences and various subbands, and the results can be found in [Appendix B.1](#). Here only the result for one subband, showing a general trend, is shown in Fig. 4.13. It can be seen that the distortion estimation is accurate when the simplification assumption is valid, *i.e.*, the ratios are close to 1 when the quantisation step is small. Also, the total distortion ΔD_b and its estimated value $\Delta \hat{D}_b$ are determined mainly from the contribution of the I set, so that $\Delta D_b \approx \Delta D_{b,I}$ and $\rho_D \approx \rho_I$.

The number of spatio-temporal subbands that are jointly encoded into one bit-stream atom is determined by the corresponding leaf ST-node. As previously defined, the set of subbands belonging to a leaf ST-node Φ_κ is denoted with Ψ_κ . During encoding of the coefficients corresponding to a particular bit-plane b and

4.4 Post-Compression Rate-Distortion Optimisation

fractional bit-plane f the reduction in distortion is recorded into $\Delta D_{\Psi_\kappa,b,f}$ and the bits spent for encoding into $\Delta R_{\Psi_\kappa,b,f}$. As these two measures are additive, it follows that:

$$\begin{aligned}\Delta D_{\Psi_\kappa,b,f} &= \sum_{\varsigma \in \Psi_\kappa} \Delta D_{\varsigma,b,f}, \\ \Delta R_{\Psi_\kappa,b,f} &= \sum_{\varsigma \in \Psi_\kappa} \Delta R_{\varsigma,b,f}.\end{aligned}$$

The slope of the rate-distortion (R-D) curve $\lambda_{\Psi_\kappa,b,f}$ for the corresponding elementary bit-stream segment is:

$$\lambda_{\Psi_\kappa,b,f} = \frac{\Delta D_{\Psi_\kappa,b,f}}{\Delta R_{\Psi_\kappa,b,f}}.$$

This determines $\lambda_{\Psi_\kappa,b,f}$ as the finest granularity of the information on the slope of the R-D curve. To organise the bit-stream in the manner that achieves the optimum quality at all target bit-rates, it has to be sorted in a way that between all elementary bit-streams the ones that result in the steepest R-D curve are included. The sorting algorithm maintains the list of such elementary bit-streams, one for each Ψ_κ , such that in each time instance the one with the highest λ is chosen and placed into the output bit-stream. If the selected elementary bit-stream is not the last one in the encoded bit-stream of the subbands in a particular Ψ_κ , the next one with the next highest fractional bit-plane index, or the next highest bit-plane index, takes its place in the maintained list.

In aceSVC, the PCRD algorithm is implemented such that the distortion is optimised for a set of target bit-rates, where the bit-rate sets can be defined for different ST-points. Since the decoding performance is optimal only at the predefined set of bit-rates, truncating the bit-stream at some point between the quality layers will inevitably lead to suboptimal performance. This is demonstrated by the following example of fine-granular quality scalability, where the sequence ‘‘City’’ is compressed in two different ways. In the first only two target bit-rates are selected - {192, 1024} kbps, while for the second a total of nine are selected - {192, 256, 320, 384, 480, 576, 672, 768, 1024} kbps. The decoding for these two cases is performed in the range of 192–634 kbps in steps of 26 kbps, and is displayed in Fig. 4.14. Since only 7 first layers for the second case are within

4.4 Post-Compression Rate-Distortion Optimisation

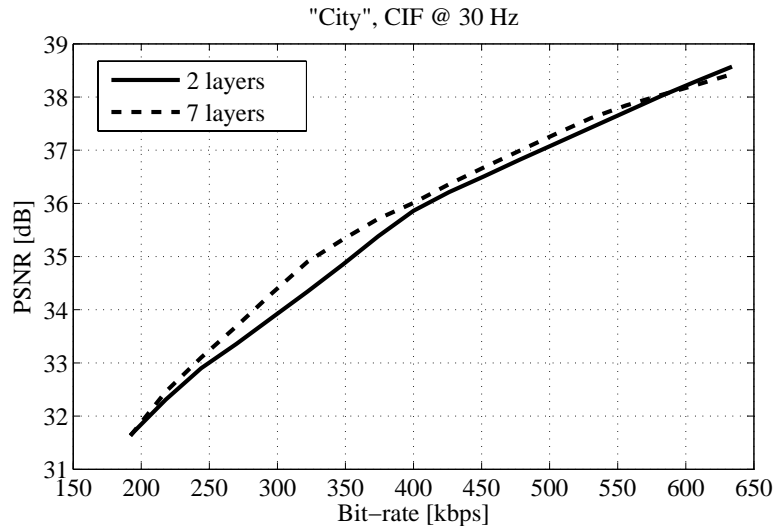


Figure 4.14: Decoding with different number of optimised bit-rate points.

the tested range, this value is indicated in the legend. The gap of up to 0.5 dB of performance loss can be observed for the case of 2 layers coding. Owing to the efficient embedding via fractional bit-planes, the truncated layer is embedded up to some degree, otherwise the gap would be even larger. The control data introduced that is needed to organise the bit-stream into many layers takes prevalence at the higher bit-rates over the gain of embedding, so the overall performance suffers and becomes lower than with 2 layer coding. It should be noted that in this case the bit-streams have been organised to support spatial and temporal scalability as well, all together supporting decoding at 25 ST-points. Therefore, the number of bits introduced per one quality layer has to be multiplied by approximately 25 to get the total number of control data bits, as in each decodable ST-point there is the target number of quality layers.

Ideally, the contribution to overall distortion of each wavelet coefficient should be weighted differently, a fact which has been already recognised [81; 82; 83]. The value of this weight is directly related to the synthesis waveform that a particular coefficient represents, *i.e.*, that reconstructs that coefficient directly to the signal domain. More precisely, if the signal representing the quantisation error is uncorrelated and of zero mean, the weight is equal to the concerned waveform energy, or its L^2 norm, given with (2.37) for 1D waveforms and with (2.38) for

4.4 Post-Compression Rate-Distortion Optimisation

2D waveforms. This is a direct consequence of Parseval's Theorem for discrete signals:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega = \sum_n h_n^2,$$

where h_n is the synthesis waveform and $H(\omega)$ its Fourier transform. As the spectrum of the quantisation error is flat and the convolution in the signal domain is translated into multiplication in the frequency domain, the energy gain factor is equal to the synthesis waveform energy. Note that this waveform can be multi-dimensional, although the notation used here is one-dimensional. Direct computation of these waveforms can be achieved by convolution of the employed wavelet filters in a sequence that reproduces the decomposition path used to generate the corresponding wavelet coefficient, similarly as it is done in (2.19) for the 1D case. However, this assumes a separable wavelet transform, as well as an infinite-length signal. Since the motion-compensated 3D spatio-temporal decomposition is generally non-separable and operates on sequences of finite dimensions, the final shape of the synthesis waveform is dependent on several factors. The non-separability factors are caused by the methods employed to improve the prediction in the temporal direction, which basically consists of modifying one-to-one pixel connections to one-to-many. To summarise, some of the following factors have to be taken into account when determining the exact synthesis waveform:

- interpolation in each lifting step of the MCTF for sub-pixel accurate compensation;
- overlapped block motion compensation [84];
- adaptive update step schemes, *e.g.*, [85];
- signal extension at the boundaries, *e.g.*, symmetric extension at frame boundaries for spatial transform, and at Group of Pictures (GOP) or sequence boundaries for temporal transform.

However, since these conditions vary from coefficient to coefficient, *e.g.*, a different interpolation filter is used in different motion blocks depending on the motion vector value, the exact values of coefficient-precise weights would be excessively

4.4 Post-Compression Rate-Distortion Optimisation

complex to compute. If the weights do not vary significantly within one subband, a mean value of the coefficients weights can be used instead for that subband. This weight is denoted with w_ζ . To determine the exact values of these mean weights, a simple measuring technique is employed here, described in the following. It should be noted that it is assumed that the non-linear methods, such as the adaptive update step, are not used, so the weights are not dependent on the processed signal values. This also means that the compression system is linear and that the error due to quantisation can be treated as a independent component of the reconstructed signal. The quantisation noise of the wavelet coefficients can be characterised as an uncorrelated signal with uniform distribution, *i.e.*, it is a noise with flat spectrum. This assumption holds only for sufficiently small quantisation steps, due to Laplacian distribution of the wavelet coefficients. In the measuring technique the sequence is firstly fully processed by the encoder, and at the decoder all of the subbands are set to zero values. The selected subband is filled with random values of uniform distribution and known overall energy, which simulates the quantisation error. By measuring the reconstructed signal energy, the mean gain factor w_ζ of the reconstruction for that particular subband is found. To provide an example, the results of measuring of the subband weights for one sequence are presented in [Appendix B.2](#).

Since in a practical implementation the two chrominance components are processed jointly and independently to the luminance component, to each a different distortion weighting factor can be associated. This enables different treatment of the chrominance components to the luminance component, if required. As the distortion of coefficients from different subbands is also multiplied with the factors stemming from the subband synthesis waveform energies, the final distortion computation for a particular colour component can be written as:

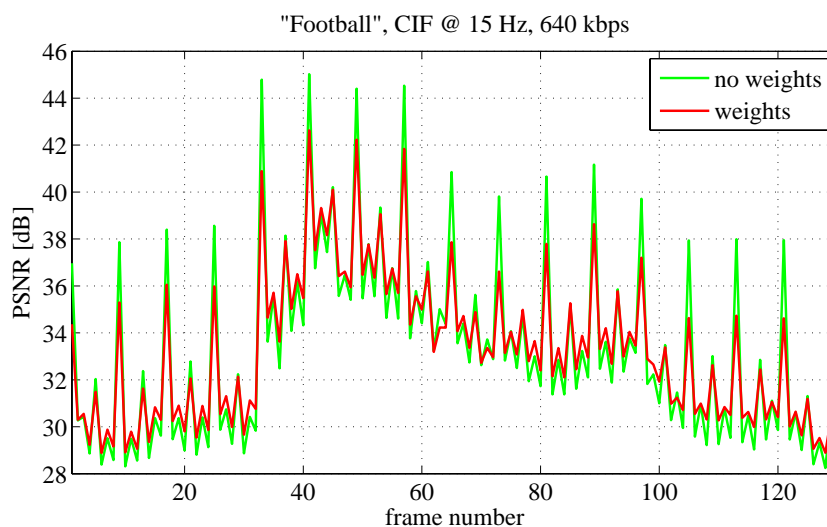
$$\Delta D_{\Psi_\kappa, b, f}^{comp} = w_{comp} \sum_{\zeta \in \Psi_\kappa} w_\zeta \Delta D_{\zeta, b, f},$$

where the component is denoted with $comp \in \{\text{luminance, chrominance}\}$, and w_{comp} is the weight associated with the particular component.

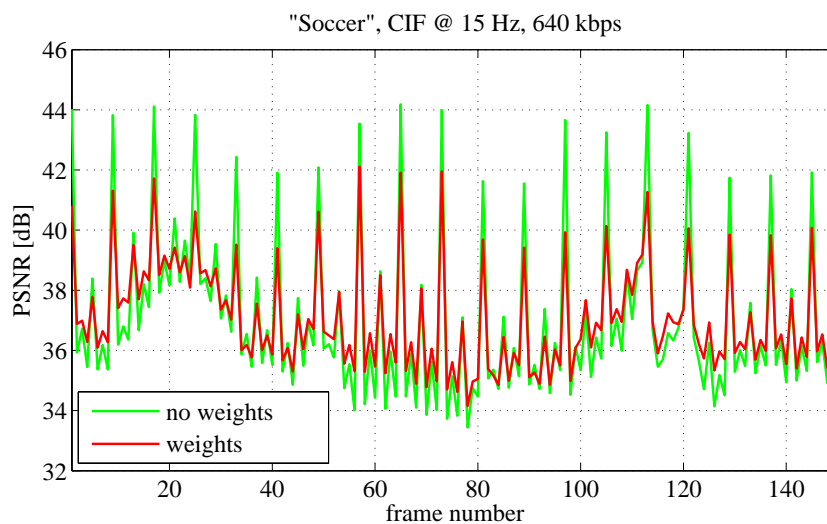
The sequences were processed using the obtained mean weighting factors. The result for two sequences are presented in [Fig. 4.15](#). It can be seen that the gain in PSNR is only marginal, but on the other hand the variation between

4.4 Post-Compression Rate-Distortion Optimisation

frames is considerably reduced, which can be confirmed by the measured standard variation.



(a) no weights - PSNR = 33.24 dB, standard deviation 3.89 dB; weights - PSNR = 33.26 dB, standard deviation 3.15 dB



(b) no weights - PSNR = 37.04 dB, standard deviation 2.67 dB; weights - PSNR = 37.06 dB, standard deviation 1.77 dB

Figure 4.15: Results with subband weights used in PCR D.

4.5 Comparison with Existing SVC Codecs

This section provides a comparison of compression result between the aceSVC and the two following codecs:

- H.264 / AVC scalable extension (JSVM software). For this test the version 5.8¹ has been used.
- test software used in MPEG's group for Wavelet Video Exploration (Vidwav) [32], originally provided by Microsoft Research Asia with additional modules approved by the Vidwav group participants [86].

All tested codecs consist of three main modules - encoder, extractor and decoder. While the JSVM codec employs the DCT-like block-based spatial transform, aceSVC and Vidwav reference software employ the frame-based wavelet transform. Regarding the underlying decomposition architecture all codecs use the $t + 2D$ decomposition, but while the JSVM software employs the closed-loop scheme, the aceSVC and the Vidwav reference software use an open-loop one. The other notable differences include the motion model, GOP organisation, quantisation, entropy coding and modelling of layers in scalable bit-stream. Because of these differences, in the following only the quality scalability performance for different codecs is compared. None of tested codecs guarantees the best achievable compression results using the provided software and default input settings. Therefore the chosen sequences and decoding points have for this test been derived from the MPEG experiments in scalable video coding in order to get representative results for the two codecs compared to aceSVC. Only the results for the Y component are displayed, as recommended in the tests used in MPEG. The chrominance components constitute a considerably smaller portion of the bit-stream so that large variations in PSNR of the chrominances have a consequence in only small variations of the luminance PSNR. Also, the Y component contributes the most to the overall subjective quality.

The results are summarised in Fig. 4.16 and Fig. 4.17. All three codecs give comparable results at wide range of bit-rates. The test sequence "City" contains slow camera panning that is well captured by the motion model employed in

¹available at CVS server garcon.ient.rwth-aachen.de, June 2006.

4.5 Comparison with Existing SVC Codecs

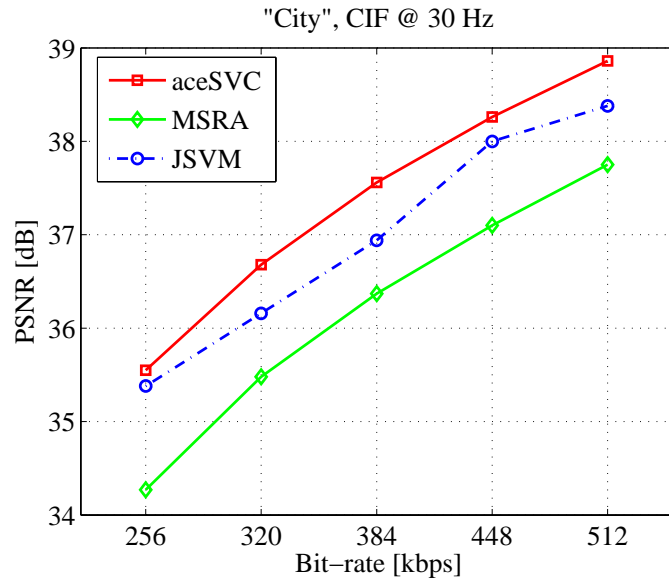


Figure 4.16: PSNR results for Y component of the “City” test sequence.

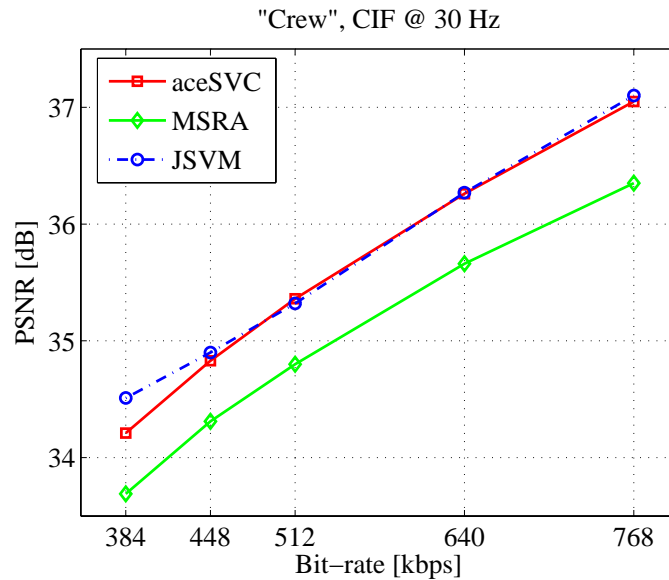


Figure 4.17: PSNR results for Y component of the “Crew” test sequence.

aceSVC. In this case aceSVC gives the best PSNR results at all test points. However, for sequences that consist of higher degree of activity, such as the “Crew” test sequence, JSVM software provides better results.

Chapter 5

Adaptive Spatial Transform for Scalable Coding

In the traditional approach to wavelet-based video coding a fixed spatial transform is used, *i.e.*, a non-adaptive signal-independent transform. Usually that is the DWT using the 9/7 biorthogonal wavelet, due to its excellent compression performance and relatively low complexity [22; 87]. However, it is a well established fact that with adaptive wavelet transforms a better energy compaction can be achieved than with transforms with a fixed set of basis functions [74; 77; 88; 89; 90], which then can lead to an excellent overall compression performance. But, to obtain gain there are generally two issues to be addressed. The first is the *bookkeeping*, where the bit-allocation algorithm must take into account the amount of additional (side) information necessary for the decoder to perform the inverse transform. There are adaptive schemes where no bookkeeping is required [45; 91], since in these schemes the adaptation depends on local characteristics of the signal. However, the embedded quantisation, as a standard tool for progressive coding, will inevitably introduce visually unpleasant non-linear artefacts. The reason is that the decoder, while performing the inverse transform, will not possess the same information that the encoder had used for adaptation. The second issue is that the adaptive transform does not necessarily result in a more compact representation of the input signal. The reason for this can be the application of an improper bit-rate allocation method that

misleads the process of choosing the transform basis or other adaptation parameters. Therefore the adaptation criteria and the amount of side information must be carefully selected.

The R-D performance of wavelets on 1D and 2D discontinuities has been investigated by many researchers and various solutions have been proposed [92; 93]. The results from the approximation theory show that for class of images containing r -times continuously differentiable smooth areas separated by smooth contours, the asymptotic R-D performance of the classical wavelet coders is limited to $D(R) \leq 1/R$, which is considerably above the optimal performance of $1/R^r$ for images without edge discontinuities [94]. Generally, adapting the signal representation for discontinuities has been proved to be more efficient than encoding its linear transform coefficients, and various techniques exploiting this fact have been proposed, to name just a few - curvelets [95], wedgelets [96], prune-join quadtree [97], matching pursuit [98], wavelet footprints [99]. These methods can be characterised as dictionary based, where the main objective is to design a dictionary $\mathfrak{D} = \{\varphi_i\}_{i \in I}$, spanning a certain space of signals \mathcal{V} , which will be used to describe a signal $x \in \mathcal{V}$. The goal here is that description is done via as small subset of \mathfrak{D} as possible, resulting in the optimal approximation \hat{x} in a rate-distortion sense. In a contrast to a transform basis, with an overcomplete dictionary \mathfrak{D} there will be not just one unique way to represent a signal x , but since φ_i are linearly dependent, there will be an infinite number of possible representations. However, the main problem of these techniques is the design of a fast search algorithm for finding the sparsest representation with a given \mathfrak{D} , since computing $\langle \varphi_i, x \rangle \forall i \in I$ would be too computationally intensive. Some of the overcomplete dictionary methods rely on the efficiency of estimation of locations of the discontinuities, which is a drawback that becomes evident when the estimation is suboptimal in regard to the human visual system, as it can cause visually annoying artefacts for large quantisation steps. Thus, the subjective performance is dependent on the edge detection method, whether it is done as a pre-processing step or on-the-fly during transform.

In this chapter, an adaptive spatial transform is introduced, designed to adapt the spatial transform on boundaries of signal segments. It is built using a novel multi-resolution framework called *connectivity-map decomposition*. This scheme

5.1 Properties of the Temporal High-pass frames

has been developed for the particular application case where the motion information is used as adaptation criteria - Motion-Driven Adaptive Transform (MDAT). MDAT exploits local properties of temporal frames, specifically high-pass temporal frames. Since the same motion information used in the encoder is also available in the decoder, the MDAT does not introduce additional bit-stream overhead. In contrast to methods that rely on locating image edges, the MDAT uses the locations of discontinuities that are artificially created by motion compensation and can be reproduced using the motion vector field. This can be likened to the so-called “oracle” scenario [97], where it is assumed that encoder possesses a perfect information on discontinuities.

5.1 Properties of the Temporal High-pass frames

As the work presented in this chapter primarily targets the spatial transform of the high-pass temporal frames in SVC, in this section the basic properties of these frames are described. The structure of content in the individual temporal frames is primarily the result of the motion compensation process that produces low- and high-pass temporal frames. The former correspond to the frames containing either a downsampled (delta low-pass filter wavelet) or a motion-compensated average of the sequence, and the latter to the frames consisting of the motion-compensated difference. To minimise this difference in the predicted frames and to obtain a compression gain, pixels in this frame are divided into different prediction mode areas. The pixels can be predicted from reference areas in the neighbouring frames or from neighbouring areas in the same frame. These areas are then treated differently in the subsequent compression steps. To reduce the computation complexity and the size of the bit-stream allocated to description of the frame partitioning, these areas are generally of regular rectangular or triangular shapes [60]. Such partitioning often causes discontinuities at the predicted area boundaries, or so-called *blocking artefacts*. There exists several strategies to reduce this effect, *e.g.*, deblocking filters [100] or Overlapped Block Motion Compensation (OBMC) [84]. Here a method that augments the existing ones is proposed, where such partitioned signals are treated as piecewise stationary, or segmented with the segments corresponding to various types of prediction areas.

5.1 Properties of the Temporal High-pass frames

Some other techniques relevant to this work, which also employ local signal adaptation by modifying the underlying transform, will be mentioned in the following. For instance, in [101] the signal is extended at both sides of the located discontinuity by extrapolation, so that filtering is applied on these extrapolated segments instead across the discontinuity. However, only a convolution implementation of filtering and disconnected segments are considered, which from the perspective of proposed connectivity-map restricts the connectivity values to the set consisting of only two elements - regularly connected and disconnected. The connectivity-map scheme provides a generalisation of this technique, as it allows continuous connectivity values. In [102] the wavelet transform is performed in a locally selected spatial direction. When the direction of maximum signal decorrelation is found, this direction is encoded and the result of transform is kept. This technique improves performance for textured images, but filtering across the edges is still performed for at least one spatial direction. In [103] a selection of the prediction orientation in lifting on a quincunx lattice [104] is used. Both approaches in [102] and [103] employ Lagrangian rate-distortion optimisation to encode the orientation map that defines the transform direction. This usually results in a coarse orientation map partitioned in rectangular blocks of varying sizes. The approach presented here is not concerned with efficient encoding of the map. Instead, it targets an accurate multi-resolution representation of the map while also defining its corresponding decomposition scheme.

The existing techniques for coding of intra blocks in high-pass temporal frames are mainly based on prediction of the intra areas by interpolation or extrapolation from the neighbouring pixels, before these neighbouring pixels are processed with the MCTF [56]. In [56] the intra blocks are called *directional I-blocks* and are selected as those for which the motion estimation (ME) fails to find a matching block in the neighbouring frames but still can be efficiently predicted from the neighbouring pixels in the same frame. This method works well for isolated intra blocks, and generally for small intra areas, so its usefulness is to a certain extent diminished by the application of the OBMC. The additional drawback of [56] is that it introduces a mismatch in the spatial scalability scenario, possibly causing a drift for decoding at lower bit-rates. This is due to the fact that the wavelet transform coefficients of the prediction error will generally not match the ones that

5.1 Properties of the Temporal High-pass frames

would be obtained by a prediction performed at lower resolutions. Due to these facts, the proposed method selects intra areas using different criteria. Specifically, those areas that cannot be efficiently predicted from any pixels, either in spatial or in temporal neighbourhood, are considered to be coded as intra.

In the $t+2D$, $2D+t+2D$ and multi-scale pyramid architectures the temporal decomposition is performed by applying MCTF in either original resolution spatial or low resolution spatial domains. Temporal frames belonging to different temporal subbands are then processed and compressed separately. Temporal scalability is achieved by decoding only a part of the bit-stream corresponding to the low-pass frames, the result being a low frame rate version of the original sequence. The low-pass frames are structurally similar to still images, as they generally contain piecewise smooth areas. On the other hand, the high-pass frames contain prediction errors that can be characterised as noise. However, due to the unavailability of good matches, they may also contain texture, edges and intra coded areas. The structure of this content mostly depends on the applied motion compensation and the sequence content. In this work, only schemes that use motion blocks as basic motion units are considered. The existence of differently compensated areas in the motion-compensated frames generally results in discontinuities between them. The magnitude and order of the discontinuity between two blocks in the predicted frames are related to the difference of their motion vectors, corresponding to different block displacements in the reference frame. To support these observations, the high-pass temporal frame resulting from one-level Haar MCTF for the first two frames of the 4CIF test sequence “Basket” is shown in Fig. 5.1. The area enclosed in a rectangle is used for examples that are presented later. The abrupt changes between intra and inter areas are clearly visible, and thus it is evident that these intra areas, if not treated differently, would create wavelet coefficients of large magnitudes, representing the boundaries between intra and inter areas. This would have a negative effect on the energy compaction property of the spatial transform, which is a key prerequisite for good compression performance.

To measure the differences between the motion vectors of neighbouring blocks, and to obtain the overall “degree” of segmentation in the high-pass frame, the magnitude of the motion vector field directional derivative is used. Let $\mathbf{z}_{m,n} =$

5.1 Properties of the Temporal High-pass frames

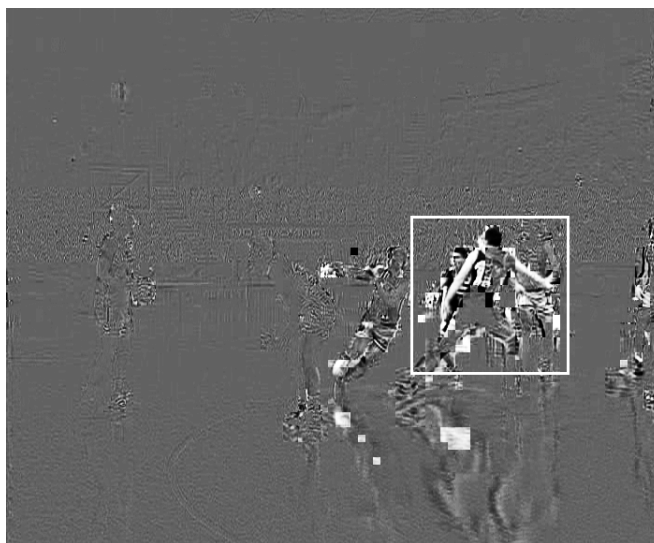


Figure 5.1: High-pass frame obtained by the MCTF of the first two frames of the sequence “Basket”, with intra blocks enabled.

$[u_{m,n}, v_{m,n}]$ be motion vector at the sampling position (m, n) , where u is the vertical displacement component (row-progression direction), and v is the horizontal displacement component (column-progression direction). As the bidirectional motion vector fields are also observed in this work, the same block can be assigned with two motion vectors. Taking into account the fact that these two vectors tend to be highly correlated, for the purpose of measuring the motion field it is sufficient to set $\mathbf{z}_{m,n}$ to the average value of these two vectors, compensated for opposite directions. For the sake of completeness, the motion vector for pixels inside intra blocks is defined as $\mathbf{z}_{m,n} = [D, D]$, where D is some large constant value far outside of the range of the potential motion vectors. The components of the directional derivative are defined as:

$$\begin{aligned} G_{m,n}^{(m)} &= \frac{\Delta \mathbf{z}_{m,n}}{\Delta m} = [u_{m+1,n} - u_{m,n}, v_{m+1,n} - v_{m,n}] \\ G_{m,n}^{(n)} &= \frac{\Delta \mathbf{z}_{m,n}}{\Delta n} = [u_{m,n+1} - u_{m,n}, v_{m,n+1} - v_{m,n}]. \end{aligned} \tag{5.1}$$

5.1 Properties of the Temporal High-pass frames

The magnitude of the motion vector field directional derivative is defined by:

$$\begin{aligned}
 \|G_{m,n}\| &= \left\| \left[\|G_{m,n}^{(m)}\|, \|G_{m,n}^{(n)}\| \right] \right\| \\
 &= \left((u_{m+1,n} - u_{m,n})^2 + (u_{m,n+1} + u_{m,n})^2 \right. \\
 &\quad \left. + (v_{m+1,n} - v_{m,n})^2 + (v_{m,n+1} + v_{m,n})^2 \right)^{\frac{1}{2}} \\
 &= \|\mathbf{z}_{m+1,n}\|^2 + \|\mathbf{z}_{m,n+1}\|^2 + 2\mathbf{z}_{m,n}^T (\mathbf{z}_{m,n} - \mathbf{z}_{m+1,n} - \mathbf{z}_{m,n+1})
 \end{aligned} \tag{5.2}$$

where $\|\cdot\|$ represents the L^2 norm of the vector of concern. As a contrast to divergence measure used in [82], and defined with:

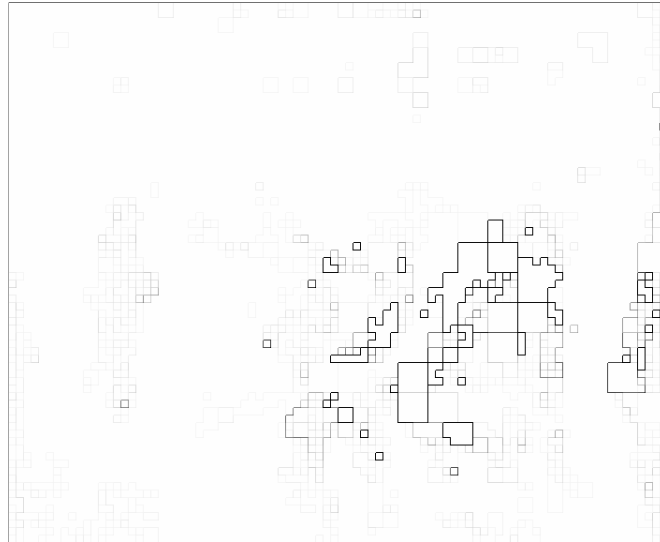
$$\begin{aligned}
 \nabla_{m,n} &= \nabla \cdot \mathbf{z}_{m,n} = \frac{\Delta u_{m,n}}{\Delta m} + \frac{\Delta v_{m,n}}{\Delta n} \\
 &= u_{m+1,n} - u_{m,n} + v_{m,n+1} - v_{m,n}
 \end{aligned}$$

the directional derivative is capable of detecting cases where $u_{m,n+1} \neq u_{m,n}$ and $v_{m+1,n} \neq v_{m,n}$.

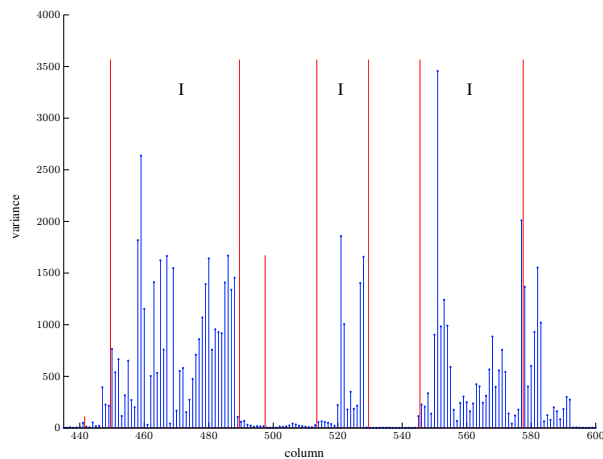
Fig. 5.2(a) displays the values of $\|G_{m,n}\|$ corresponding to the motion vector field used in Fig. 5.1. Obviously, $\|G_{m,n}\| = 0$ inside motion blocks, and $\|G_{m,n}\| \approx 2D$ for intra-inter borders. Although the neighbouring inter compensated areas of different displacements also contribute to the partitioning of high-pass temporal frames, their impact is less critical than that of the intra-inter borders. Fig. 5.2(b) illustrates the frame partition effect given by the differences between motion vectors of neighbouring blocks, as measured by the variance in the vertical direction of the coefficients in the selected portion of the frame, as highlighted in Fig. 5.1. The partitioning between different motion areas is marked with vertical lines, where the height of a line is proportional to the capped $\|G_{m,n}\|$. Generally, it can be observed that the magnitude of discontinuity in the high-pass temporal frame is strongly correlated with $\|G_{m,n}\|$. Therefore, the conclusion is that a simple measure for the motion vector field disparity, *e.g.*, (5.1) and (5.2), can efficiently capture the pattern of discontinuities between differently compensated areas.

These observations provide motivation to use motion information during the spatial transform in order to obtain a better representation of the motion-induced discontinuities, and to reduce the artefacts related to the wavelet low-pass filtering

5.1 Properties of the Temporal High-pass frames



(a)



(b)

Figure 5.2: (a) Magnitude of the motion vector field directional derivative $\|G_{m,n}\|$ corresponding to Fig. 5.1. The intensities are inversely proportional to the magnitude, and are capped such that $\|G_{m,n}\| > 8$ is displayed in black colour. (b) Variance in vertical direction of the pixels enclosed in the highlighted rectangle from Fig. 5.1. Letters “I” mark the positions of intra areas.

across these discontinuities. This conclusion holds even more strongly for MCTF of several levels of decomposition, as the frames in the lower temporal subbands

will be less correlated. In other words, it can be expected that on the lower temporal levels the ME will not be able to correctly capture the motion of an even larger portion of the frames.

5.2 A Joint Wavelet Connectivity-Map Decomposition

The concept of the signal samples connectivity is introduced here, that will be used in the lifting for adaptation to the local signal characteristics. The connectivity-map of a signal is defined as a structure that specifies neighbourhood relations for each of its samples. As the prediction operator \mathcal{P} in lifting basically performs a linear interpolation, the connectivity of a pixel whose value is being predicted can be regarded as weights applied to the neighbouring pixels. Using this concept a wavelet transform can be generalised for irregular grids. Then the connectivity-map is used to specify pixel weights in each lifting step. The lifting on regular grids, as in (2.46), is then just a special case, where the neighbouring pixels are equally relevant, and therefore they are taken with equal weights. The same concept of connectivity can be used for regular sampling, but in that case the connectivity corresponds to the relevance of neighbouring pixels for prediction. Relevance is derived from the local signal properties and not by the sampling locations.

Given a pixel x_k at sampling position k , its left and right connectivity values are denoted by $c_{k,-1}$ and $c_{k,1}$, respectively, or $c_k = \{c_{k,-1}, c_{k,1}\}$. The pixels in (2.46) are regarded as *regularly connected*, *i.e.*, their corresponding connectivity values are $c_k = \{1, 1\}$. However, in the general case the weights are not equal, $c_{k,-1} \neq c_{k,1}$. Using this concept, the general lifting step can be rewritten by replacing the sum in (2.46) with the weighted sum. Thus, a general lifting step becomes:

$$\tilde{x}_k = x_k + \lambda \cdot (c_{k,-1}x_{k-1} + c_{k,1}x_{k+1}), \quad (5.3)$$

where the lifting coefficient is denoted by λ , and the signal resulting from the lifting step by \tilde{x} . Here a simplified representation of a lifting step without prior

5.2 A Joint Wavelet Connectivity-Map Decomposition

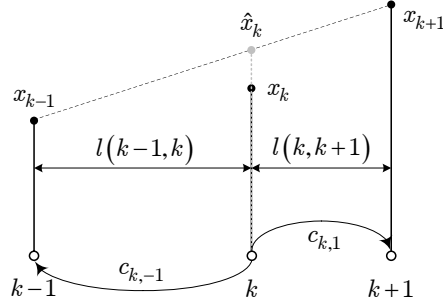


Figure 5.3: Geometrical interpretation of the connectivity. The corresponding interpolation relation is $\frac{1}{2}(c_{k,-1}x_{k-1} + c_{k,1}x_{k+1})$, producing the interpolated value \hat{x}_k .

lazy transform is used. Obviously, the connectivity values will depend on the sampling distance between the neighbouring pixels and the central one. If the distance between pixels x_i and x_j is denoted by $l(i, j)$, as in Fig. 5.3, then by geometric manipulation it follows that:

$$c_{k,t} = 2 \frac{l(k, k-t)}{l(k-t, k+t)}, \quad (5.4)$$

where $t \in \{-1, 1\}$ is used to specify the direction, so that an inversion of the sign specifies an opposite direction. It can be seen that the pixels x_k and x_{k+t} are *disconnected* when $l(k, k+t) \rightarrow \infty$. In that case $c_{k,-t} \rightarrow 2$ and $c_{k,t} \rightarrow 0$, *i.e.*, the corresponding lifting step can be viewed as applying symmetric extension on a boundary pixel x_k , thus replicating the x_{k-t} on the other side of the boundary instead of using x_{k+t} .

To a given signal $\mathbf{x}^{(j)}$ a connectivity-map $\mathbf{c}^{(j)} = \mathcal{C}(\mathbf{x}^{(j)})$ is assigned, where j denotes a particular scale, and $\mathcal{C}(\cdot)$ an operator of connectivity-map assignment. Fig. 5.4 shows the graphical representation of relations between the wavelet coefficients and their connectivity-maps. The wavelet analysis of the signal $\mathbf{x}^{(j)}$ produces a signal at the next coarser scale $\mathbf{x}^{(j-1)}$ (low-pass signal) and the detail signal $\mathbf{y}^{(j-1)}$ (high-pass signal). The connectivity-map is independent of the number of lifting steps, *i.e.*, on the filter length, so that for each lifting step the same connectivity-map is used.

To enable a resolution scalable representation of the signal and its connectivity-map, a process to derive coarser scale connectivity-maps is needed. This intro-

5.2 A Joint Wavelet Connectivity-Map Decomposition

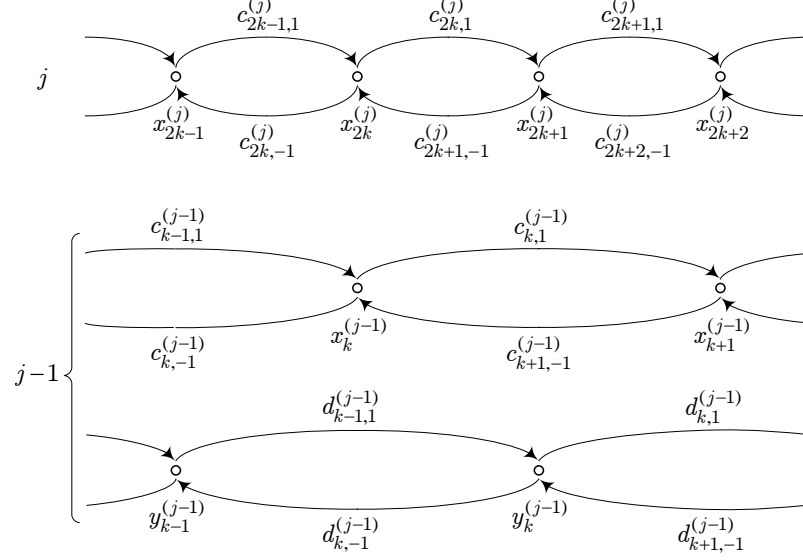


Figure 5.4: Connectivity-map analysis: $\mathcal{A}(\mathbf{c}^{(j)}) = \{\mathbf{c}^{(j-1)}, \mathbf{d}^{(j-1)}\}$; Wavelet analysis: $\mathcal{A}(\mathbf{x}^{(j)}) = \{\mathbf{x}^{(j-1)}, \mathbf{y}^{(j-1)}\}$.

duces the concept of *connectivity-map analysis*, as an analogous process to the wavelet analysis. The analysis of the connectivity-map $\mathbf{c}^{(j)}$ results in a low-pass map $\mathbf{c}^{(j-1)} = \mathcal{C}(\mathbf{x}^{(j-1)})$, associated with the low-pass signal $\mathbf{x}^{(j-1)}$, and a high-pass map $\mathbf{d}^{(j-1)} = \mathcal{C}(\mathbf{y}^{(j-1)})$, associated with the high-pass signal $\mathbf{y}^{(j-1)}$. The wavelet analysis of $\mathbf{x}^{(j)}$ is given by the lifting steps equations (2.46). In order to satisfy particular transform requirements the following two constraints are imposed on the connectivity-map. Firstly, the relation between left and right connectivity values of a signal sample at position k is defined as $c_{k,-1} + c_{k,1} = 2$. This constraint allows for preservation of the DC component of the signal. The second requirement poses an exception to the first one, where if $c_{k,t} = 0$ then $c_{k+t,-t} = 0$. This constraint eliminates one-sided disconnections. If one-sided disconnections are allowed this would translate into locally infinite sampling frequency, which is of no practical meaning. If (5.4) is imposed on both $\mathbf{c}^{(j)}$ and

5.2 A Joint Wavelet Connectivity-Map Decomposition

$\mathbf{c}^{(j-1)}$, the following analysis relation for $\mathbf{c}^{(j)}$ can be derived:

$$\begin{aligned} c_l &= c_{k,-1}^{(j)} c_{k-1,-1}^{(j)} \\ c_r &= c_{k,1}^{(j)} c_{k+1,1}^{(j)} \\ c'_{k,t} &= 2 \frac{c_{k,t}^{(j)} c_{k+t,t}^{(j)}}{c_l + c_r}, \text{ for } c_l \neq -c_r. \end{aligned} \tag{5.5}$$

Here \mathbf{c}' denotes a map having the map $\mathbf{c}^{(j-1)}$ for even samples and the map $\mathbf{d}^{(j-1)}$ for odd samples, *i.e.*, $c'_{2k} = c_k^{(j-1)}$ and $c'_{2k+1} = d_k^{(j-1)}$. In other words, the resulting connectivity-maps are given by the lazy transform of \mathbf{c}' . The case of $c_l + c_r = 0$, when (5.5) is not defined, is processed separately. If $c_l = c_r = 0$, this means that two disconnections appear on the distance of three pixels or less. In that case the resulting connectivity is defined as $c'_k = \{0, 0\}$, *i.e.*, an *isolated pixel* occurs. The occurrence of isolated pixels can be dealt with in two ways - by preserving the isolated pixels, or by merging them with the neighbouring segment, either on the left or right side, or both sides. By keeping the isolated pixels, the requirement for preserving the DC component cannot be fulfilled, but on the other hand merging can cause unwanted leaking of the energy from one segment to the other when quantisation is applied. If $c_l = -c_r$, this corresponds to the case when sampling frequency is locally negative, so the connectivity values can take negative values.

A dual relation to (5.5) can be derived from $l_k = l(k-1, k) / l(k, k+1)$. In this case the analysis relation becomes:

$$l'_k = l_k l_{k+1} \frac{l_{k-1} + 1}{l_{k+1} + 1}, \text{ for } l_{k+1} \neq -1.$$

The connectivity values needed for the lifting are then obtained as $c_{k,-1} = 2 / (l_i + 1)$ and $c_{k,1} = 2l_i / (l_i + 1)$. As this conversion is required, this whole process involves a larger number of arithmetic operations than in (5.5), rendering it less efficient. For this reason this dual transform is not further discussed here

A result similar to (5.5) can be found in [105] where a generalisation of the Haar transform called the *Unbalanced Haar Transform* is used for data defined on the intervals. The weights are in that case computed by integration over the interval corresponding to the support size of a coefficient. In contrast to that scheme, the proposed connectivity-map can be used in a general case of irregular sampling and is completely defined in a discrete setting. This guarantees

5.2 A Joint Wavelet Connectivity-Map Decomposition

preservation of the coefficients' positions in the analysis, implying that cross-scale disconnections are also preserved. Moreover, the weights are not associated with intervals between coefficients but with the coefficients themselves, where number and values of weights depend on the neighbouring coefficients. This fact appeals for an extension to non-separable lifting of higher dimension signals based on factoring of multivariate polynomials [106].

Following the analogy with wavelet analysis, the synthesis equations need to be established too. To achieve a perfect reconstruction of $\mathbf{c}^{(j)}$ first it is necessary to determine the conditions under which the equations (5.5) are reversible. For a signal of length K , the boundary conditions are $c_{0,1} = c_{K-1,-1} = 2$, since $c_{0,-1} = c_{K-1,1} = 0$. As $c_{k,-1} = 2 - c_{k,1}$, and if signal consists of only one segment, there are $K - 2$ unknown values of the $\mathbf{c}^{(j)}$ map. Using the same boundary conditions on $\mathbf{c}^{(j-1)}$ and $\mathbf{d}^{(j-1)}$, from these maps $K - 4$ values can be obtained. This means that critical sampling in (5.5) will lead to an irreversible transform. To obtain a reversible transform, at least 2 additional connectivity values need to be transmitted as side information. The exact contents of that side information can be deduced from the following synthesis equation, which follows directly from (5.5):

$$c_{k+1,1}^{(j)} = c_{k-1,-1}^{(j)} \frac{c_{k,-1}^{(j)} c'_{k,1}}{c_{k,1}^{(j)} c'_{k,-1}}, \text{ for } c_{k,1}^{(j)} c'_{k,-1} \neq 0. \quad (5.6)$$

This relation involves progression from left to right, which means that c_{k+1} can be reconstructed only if c_k and c_{k-1} have already been reconstructed. It can be observed that (5.6) cannot be applied for $k \in \{0, 1\}$ because of the boundary conditions. Consequently, $c_1^{(j)}$ and $c_2^{(j)}$ cannot be reconstructed using (5.6). Therefore, these connectivity values must be preserved as the side information during analysis, which for analysis on scale j is denoted with $\mathbf{c}_{side}^{(j)}$. Also, if disconnections occur within the signal, *i.e.*, if the signal is composed of multiple segments, then synthesis has to be “restarted” at the segment beginning. Depending on the length of a segment, at most two connectivity values from it have to be stored in $\mathbf{c}_{side}^{(j)}$. It should be noted that, using this method, the positions of disconnections, *i.e.*, the positions of starting pixels of the segments, do not have to be encoded explicitly, as they can be deduced from the decoding order.

5.2 A Joint Wavelet Connectivity-Map Decomposition

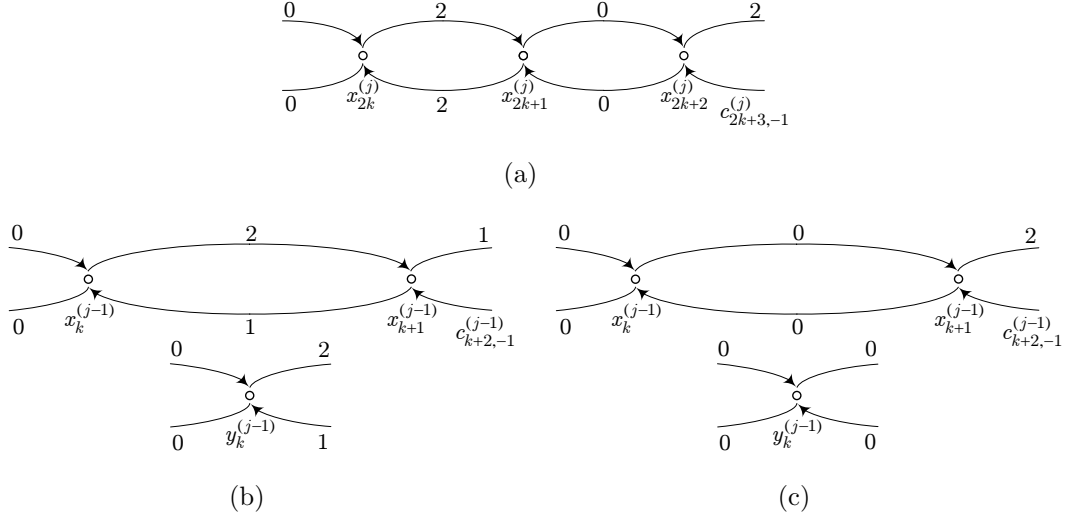


Figure 5.5: Connectivity-map analysis for a two-pixel segment $\{x_{2k}^{(j)}, x_{2k+1}^{(j)}\}$. (a) Original connectivity-map on scale j . Approximation and detail signal connectivity-maps on the next coarser scale $j - 1$ (b) obtained by merging of the isolated pixels; here merging on the right side is performed; (c) obtained by preserving the isolated pixels.

An example of analysis of the signal involving a signal segment of length 2, with two methods of treating the isolated pixels, is provided in Fig. 5.5. It is interesting to note that by using the connectivity-map in Fig. 5.5(a), the two pixels in the segment are effectively transformed using the Haar wavelet. The reason is that, due to wavelet property (2.29), which requires $H_0^\omega(\pi) = 0$, all biorthogonal wavelets become equivalent to the Haar wavelet when applied on a two pixels segment and using the symmetric extension at the boundaries.

Concerning the computational complexity of DWT using the connectivity-map, two more multiplications are necessary in the wavelet lifting, per lifting step. The connectivity-map analysis requires in average two multiplications, one addition, one division and three comparisons per connectivity value. The average requirement for the synthesis are three multiplications, one division and two comparisons. Generally, it can be said that the introduced complexity is reasonably low. The pseudocode for algorithm that performs the connectivity-map analysis with merging of the isolated pixels on the right side, is given by Algorithm 1. The

5.2 A Joint Wavelet Connectivity-Map Decomposition

variant of algorithm that preserves isolated pixels is similar. Note that an in-place

Algorithm 1 The connectivity-map analysis algorithm

```

procedure CMAP_ANALYSIS( $\mathbf{c}^{(j)}$ )
     $K \leftarrow \text{length}(\mathbf{c}^{(j)})$ 
     $\mathbf{c}' \leftarrow \text{initialise\_map}(K)$ 
     $\text{set\_left\_boundary}(\mathbf{c}')$ 
     $\mathbf{c}_{\text{side}}^{(j)} \leftarrow \text{push\_back}(c_1^{(j)})$ 
     $c_l \leftarrow 0$ 
    for  $k \leftarrow 2$  to  $K - 2$  do
        if  $((c_l = 0) \wedge (c_{k,-1}^{(j)} \neq 0) \vee (k \geq K - 3))$  then
             $\mathbf{c}_{\text{side}}^{(j)} \leftarrow \text{push\_back}(c_k^{(j)})$ 
        end if
         $c_l \leftarrow c_{k,-1}^{(j)} c_{k-1,-1}^{(j)}$ 
         $c_r \leftarrow c_{k,1}^{(j)} c_{k+1,1}^{(j)}$ 
         $c_\Sigma \leftarrow c_l + c_r$ 
        if  $c_\Sigma \neq 0$  then
             $c'_k \leftarrow \{2c_l/c_\Sigma, 2c_r/c_\Sigma\}$ 
        else
             $c'_k \leftarrow \{1, 1\}$  ▷ merging
        end if
        if  $c'_{k-2,1} = 0$  then ▷ check if merging correct
             $c'_k \leftarrow \{0, 2\}$ 
        else if  $c'_{k,-1} = 0$  then
             $c'_k \leftarrow \{1, 1\}$ 
        end if
    end for
     $\text{set\_right\_boundary}(\mathbf{c}')$ 
     $\{\mathbf{c}^{(j-1)}, \mathbf{d}^{(j-1)}\} \leftarrow \text{lazy}(\mathbf{c}')$ 
end procedure

```

variant of the algorithm is feasible, where using \mathbf{c}' is not necessary, however, for the sake of clarity here it is not presented. The corresponding pseudocode for algorithm that performs the connectivity-map synthesis is given by Algorithm 2.

5.2 A Joint Wavelet Connectivity-Map Decomposition

Algorithm 2 The connectivity-map synthesis algorithm

```

procedure CMAP_SYNTHESIS( $\mathbf{c}^{(j-1)}$ ,  $\mathbf{d}^{(j-1)}$ ,  $\mathbf{c}_{side}^{(j)}$ )
   $K \leftarrow \text{length}(\mathbf{c}^{(j-1)}) + \text{length}(\mathbf{d}^{(j-1)})$ 
   $\mathbf{c}' \leftarrow \text{inverse\_lazy}(\mathbf{c}^{(j-1)}, \mathbf{d}^{(j-1)})$ 
   $\mathbf{c}^{(j)} \leftarrow \text{initialise\_map}(K)$ 
   $\text{set\_left\_boundary}(\mathbf{c}^{(j)})$ 
   $c_1^{(j)} \leftarrow \text{pop\_front}(\mathbf{c}_{side}^{(j)})$ 
  for  $k \leftarrow 1$  to  $K - 3$  do
     $c_l \leftarrow c_{k,-1}^{(j)} c_{k-1,-1}^{(j)}$ 
     $c_r \leftarrow c_{k,1}^{(j)}$ 
    if  $c_r = 0$  then
       $c_{k+1}^{(j)} \leftarrow \{0, 2\}$ 
    else if  $c_l = 0$  then
       $c_{k+1}^{(j)} \leftarrow \text{pop\_front}(\mathbf{c}_{side}^{(j)})$ 
    else
       $c_{k+1,1}^{(j)} \leftarrow (c_l c'_{k,1}) / (c_r c'_{k,-1})$ 
       $c_{k+1,-1}^{(j)} \leftarrow 2 - c_{k+1,1}^{(j)}$ 
    end if
  end for
   $\text{set\_right\_boundary}(\mathbf{c}^{(j)})$ 
end procedure

```

To demonstrate the behaviour of the presented decomposition scheme on a sharp discontinuity, a 9/7 wavelet reconstruction of the signal representing the Heaviside step function is shown in Fig. 5.6. The anti-symmetrical setting with the central point at zero is chosen since the used wavelet is of linear phase. The step signal is defined with 512 samples, where $x_k = -1$ for $k = 0, \dots, 255$, central sample $x_{256} = 0$ and $x_k = 1$ for $k = 257, \dots, 511$. The connectivity at x_{256} is set to regular. Three different connectivity values between the central pixel and the neighbouring ones are tested, $c_{255,1} = c_{257,-1} = c$ for $c \in \{1, 0.1, 0.01\}$. The case $c = 1$ refers to regularly connected pixels and is equivalent to the non-adaptive wavelet decomposition. If the original scale is denoted with j_0 , it can be shown

5.2 A Joint Wavelet Connectivity-Map Decomposition

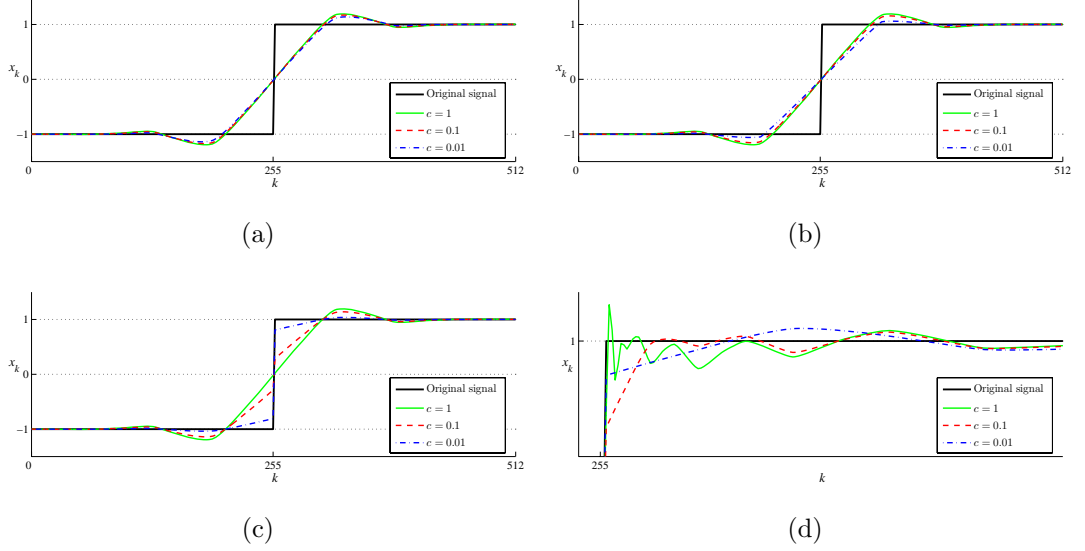


Figure 5.6: Wavelet 9/7 reconstruction of the step signal, decomposition of 5 levels is performed. Original scale signal is $\mathbf{x}^{(5)}$. (a) Reconstruction using $\mathbf{x}^{(0)}$. Connectivity-maps $\mathbf{c}^{(0)}$ and $\mathbf{d}^{(j)}$ are set to regular. (b) Reconstruction using $\mathbf{x}^{(0)}$. Subbands $\mathbf{y}^{(j)}$, $j = 0, \dots, 4$, are set to zeros. Connectivity-maps $\mathbf{d}^{(j)}$, $j = 0, \dots, 4$, inferred from $\mathbf{c}^{(0)}$. (c) Reconstruction using $\mathbf{x}^{(0)}$ and $\mathbf{d}^{(j)}$, $j = 0, \dots, 4$. Subbands $\mathbf{y}^{(j)}$, $j = 0, \dots, 4$ are set to zeros. (d) Magnified detail, reconstruction with quantised subbands and all connectivity-maps. Quantisation step is $q = 0.5$.

that the connectivity c on coarser scales is given by relation:

$$c^{(j)} = \frac{c \cdot 2^{j_0 - j}}{1 + c \cdot (2^{j_0 - j} - 1)}.$$

It can be observed that $c^{(j)} \rightarrow 1$ for $j \rightarrow -\infty$, *i.e.*, the connectivities converge to the regular ones at the coarser scales. This example shows that using the proposed connectivity-map enables better representation of the signals at discontinuities, with less “ringing” artefacts contributed to the quantisation.

In cases where the connectivity-map of the high-pass subband is not available, its approximation can be inferred from the connectivity-map of the low-pass subband. The high-pass subband connectivity-map inference relation can be deduced using (5.4), for instance by assuming that the connectivity-map of the samples at same positions on the next finer scale consists of regular connections. Then

5.2 A Joint Wavelet Connectivity-Map Decomposition

the missing high-pass subband connectivity-map is given with:

$$d_k^{(j)} = \left\{ 2 \frac{c_{k,-1}^{(j)}}{c_{k,-1}^{(j)} + c_{k+1,-1}^{(j)}}, 2 \frac{c_{k+1,-1}^{(j)}}{c_{k,-1}^{(j)} + c_{k+1,-1}^{(j)}} \right\}. \quad (5.7)$$

If a signal with even number of samples is observed, for the high-pass connectivity-map value at the right signal boundary the relation is slightly different:

$$d_k^{(j)} = \left\{ 4 \frac{c_{k,-1}^{(j)}}{3c_{k,-1}^{(j)} + c_{k,1}^{(j)}}, 2 \frac{c_{k,-1}^{(j)} + c_{k,1}^{(j)}}{3c_{k,-1}^{(j)} + c_{k,1}^{(j)}} \right\}. \quad (5.8)$$

The relations (5.7) and (5.8) were used to obtain the results shown in Fig. 5.6(b).

It can be easily verified that the wavelet transform on a fixed connectivity-map is linear. Furthermore, for the case of connectivity-map analysis using merging, the equivalent filterbank response at $\omega = 0$ and $\omega = \pi/2$ is not changed. This means that if the original signal is a constant, then all detail coefficients are zero, and conversely if the original signal spectrum consists of Nyquist frequency only, the approximation coefficients are zero.

The approximation power of a wavelet transform is measured in polynomial cancellation property, where the high-pass coefficients of polynomials up to certain degree are all zero, and is related with the number of vanishing moments property, as defined in Section 2.3. Connectivity-map can achieve preservation of the first-degree polynomial cancellation property for the high-pass filter in the case of an irregularly sampled signal. To verify this property let us consider a case where a segment of length l_{AB} is removed from a regularly sampled line. Now the signal can be seen as an irregularly sampled line with one irregularity, or as a line with a discontinuity that splits it into segments A and B . To preserve the first-degree polynomial cancellation, the connectivity-map values for the first pixel sampled on the right side of the discontinuity should satisfy:

$$\begin{aligned} c_{-1} &= \frac{2}{2 + d_{AB}} \\ c_1 &= \frac{2(1 + d_{AB})}{2 + d_{AB}}. \end{aligned} \quad (5.9)$$

For the pixel on the opposite side of the discontinuity, the connectivity-map values are merely mirrored from these. Using (5.9) the performance of analysis of

5.3 Motion-Driven Adaptive Transform (MDAT)

irregularly sampled first and higher degree polynomials is also enhanced, due to the differentiation properties of the wavelets. The examples for wavelet transform for polynomials of degrees 1 to 4 are shown in Fig. 5.7. It can be seen that when connectivity-map is used, the high-pass coefficients are zero in the case of degree 1 and are considerably smaller for higher degrees.

5.3 Motion-Driven Adaptive Transform (MDAT)

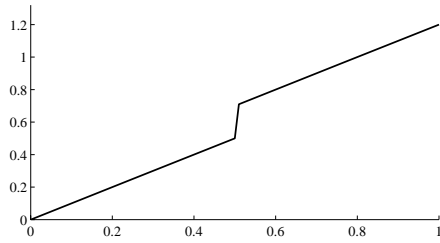
Since the borders between differently compensated areas in the high-pass temporal frames can be regarded as sharp edges, the non-adaptive spatial transform fails to concentrate most of the energy into the low-pass subbands, leaving high-amplitude edge coefficients in the high-pass subbands. To mitigate this problem, an adaptive transform based on the proposed connectivity-map can be used. It is assumed that high-pass temporal frames are divided into intra- and inter-coded areas, as defined by the corresponding motion information. All syntax elements of motion information related to the partitioning of frames for motion compensation are known by both encoder and decoder and therefore it can be used in spatial forward and inverse transforms. For the same reason, the proposed scheme does not add any side information or transmission overheads. Temporal scalability is also supported since the motion vector fields at the lower temporal scalability layers are not dependent to the ones at the higher levels. The process of adaptation is illustrated in Fig. 5.8, where the adaptation points are presented as borders dividing the input signal in different motion compensation areas.

The following part of this section is divided as follows. Subsection 5.3.1 introduces a method of specifying the connectivity-map for a given motion vector field, while Subsection 5.3.2 concentrates on a specific case where only information on intra and inter coded areas of a frame is used as a criterion for adaptation.

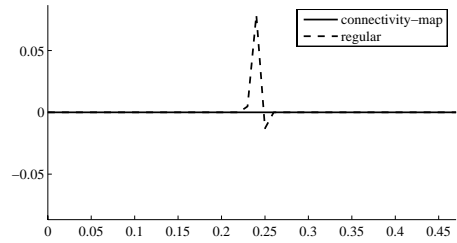
5.3.1 MDAT for General Motion Vector Fields

The relevance of the neighbouring pixels with respect to the one at the sampling position (m, n) can be expressed through the ratio of displacements $\rho_{m,n}$ of the neighbouring pixels on the left and right sides. Pixels are processed independently

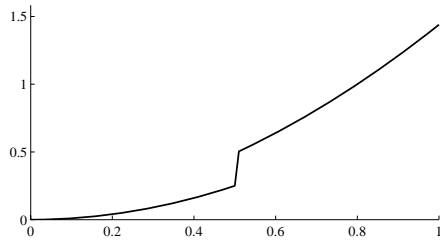
5.3 Motion-Driven Adaptive Transform (MDAT)



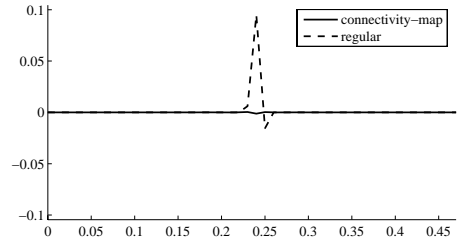
(a) degree 1



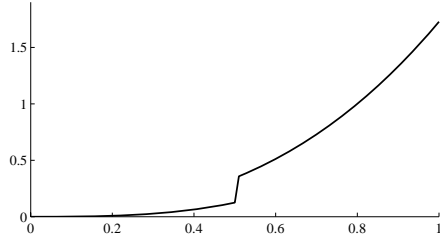
(b) wavelet high-pass coefficients for (a)



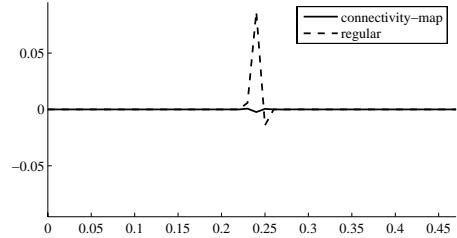
(c) degree 2



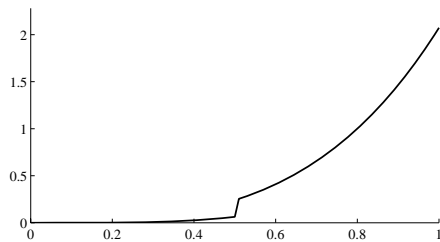
(d) wavelet high-pass coefficients for (c)



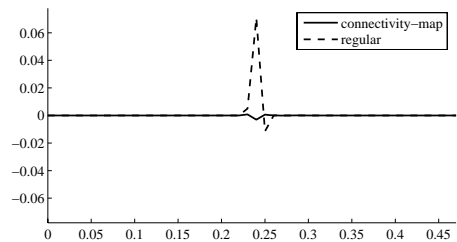
(e) degree 3



(f) wavelet high-pass coefficients for (e)



(g) degree 4



(h) wavelet high-pass coefficients for (g)

Figure 5.7: Polynomials of degrees 1 to 4 with one discontinuity and the resulting high-pass coefficients, using the non-adaptive wavelet transform and the joint wavelet-connectivity map decomposition.

5.3 Motion-Driven Adaptive Transform (MDAT)

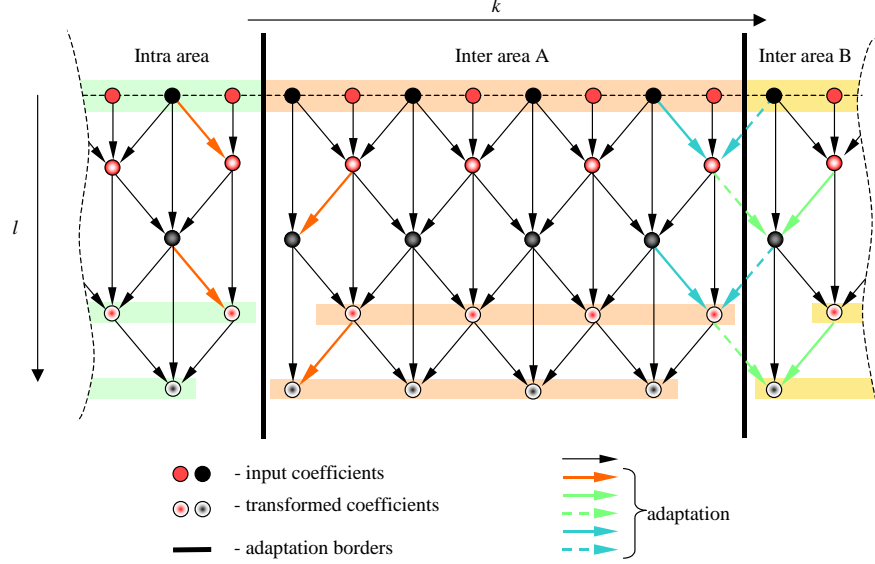


Figure 5.8: MDAT scheme. The direction of the lifting process is denoted as the direction of increase of the lifting step number l . The input and the resulting pixels are highlighted. The approximation $\mathbf{a}^{(l)}$ and detail $\mathbf{b}^{(l)}$ pixels are marked with dark and bright dots, respectively.

and identically for both spatial directions, thus the relations will be given just for one direction. For the horizontal direction, the ratio is given by:

$$\rho_{m,n} = \frac{1 + \mu \left\| G_{m,n}^{(n)} \right\|}{1 + \mu \left\| G_{m,n-1}^{(n)} \right\|}, \quad (5.10)$$

where μ is a tuning parameter, that is used to adjust the strength of the connections in the connectivity-map. The lower value of μ pushes connectivity values towards regularly-connected, while the higher value produces less connected pixels.

If the ratio (5.10) is denoted simply by ρ , one relation for connectivity values for a pixel at position (m, n) , that is an exponential function of ρ and that satisfies $c_{-1} + c_1 = 2$, such that $c_{-1}(\rho) = c_1(1/\rho)$, is:

$$c_1(\rho) = 2e^{-\rho} \frac{e^{\rho+1/\rho}}{e^{\rho} + e^{1/\rho}}. \quad (5.11)$$

5.3 Motion-Driven Adaptive Transform (MDAT)

It can be seen that if intra displacement D is defined as $D \rightarrow \infty$, which is denoted as D_∞ , the connectivity values between intra and inter pixels converge to zero. Thus, this setting for D is used to produce disconnections on the inter-intra borders. Moreover, it is defined that when $\mu = 0$ in (5.10) only adaptation on inter-intra borders is performed, while all other connections are regular.

To assess effectiveness of energy compaction of MDAT a simple experiment is devised where the proposed adaptive transform is compared to the non-adaptive, with intra blocks enabled. The results presented in Fig. 5.9 were obtained using the relations (5.10) and (5.11) to generate the connectivity-map of the frame portion displayed in Fig. 5.1. Fig. 5.9(a) and Fig. 5.9(b) display the connectivity-map for cases of $(D = D_\infty, \mu = 0)$ and $(D = D_\infty, \mu = 0.5)$, respectively. In Fig. 5.10 the resulting wavelet coefficients of the LH_1 subband (one level of DWT, horizontal low-pass, vertical high-pass) are shown, obtained by using these two different connectivity-maps, along with the result of the non-adaptive transform. It can be visually confirmed that the case of the full connectivity-map shows the best performance in minimising the energy of the wavelet coefficients around the motion compensation induced discontinuities. The energies of the coefficients in the shown portion are 3.8×10^5 , 2.7×10^5 and 2.4×10^5 for the non-adaptive, and the two cases from Fig. 5.9, respectively. This implies that more energy is concentrated in the lower subbands so that an improved compression performance can be expected for the case of MDAT. To illustrate the result of the decomposition of the connectivity-map itself, the lower resolution connectivity-maps of the one shown in Fig. 5.9(b) are displayed in Fig. 5.11.

5.3.2 MDAT for Intra-Inter Boundaries

Intra-coded blocks in high-pass frames possess properties that differ the most from those in other areas of the motion-compensated frames. As hinted by the energy of wavelet coefficients from Fig. 5.10, it can be expected that the adaptation is most beneficial on the inter-intra borders. Thus, the process of creating the connectivity-map and the transform itself can be significantly simplified in the special case of $\mu = 0$.

The motion block type parameter $MB_{m,n} \in \{\text{intra,inter}\}$ defines whether a

5.3 Motion-Driven Adaptive Transform (MDAT)

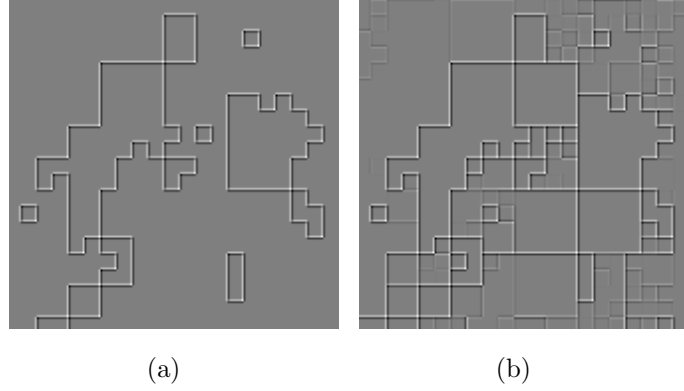


Figure 5.9: Connectivity-map for the selected portion of the frame in Fig. 5.1. (a) Intra-inter connectivity only ($D = D_\infty, \mu = 0$). (b) Full motion map connectivity ($D = D_\infty, \mu = 0.5$).

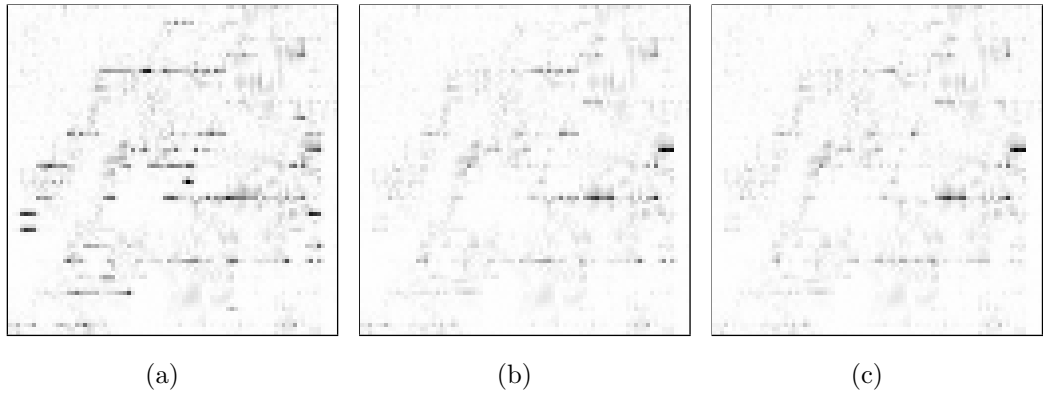


Figure 5.10: Detail of the LH_1 subband obtained with non-adaptive transform and MDAT. Absolute values of coefficients are displayed, where intensity is inversely proportional to magnitude. (a) Non-adaptive transform, (b) MDAT ($D = D_\infty, \mu = 0$), (c) MDAT ($D = D_\infty, \mu = 0.5$).

pixel $x_{m,n}$ at sampling position (m,n) is in an intra- or an inter-coded block. The transform based on this parameter is performed using a set of connectivity-map coefficients c_{m,n,t_n} and c_{m,n,t_m} that define connectivity in the horizontal and vertical direction, respectively. Here $t_m, t_n \in \{-1, 1\}$ and $c_t \in \{0, 1, 2\}$. For the

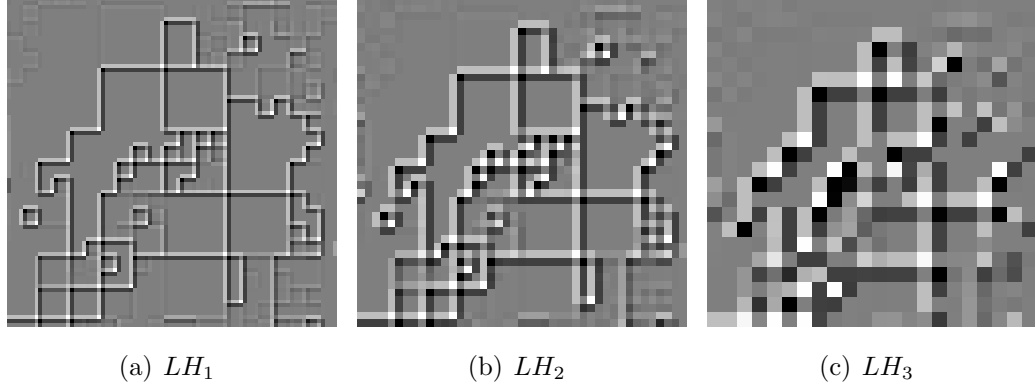


Figure 5.11: Example of the 2D connectivity-map decomposition for the selected portion of frame. The left-side connectivity values of pixels are added for horizontal and vertical directions, and normalised to the grayscale range.

transform of rows it follows:

$$c_{m,n,t_n} = \begin{cases} 0 & \text{if } MB_{m,n} \neq MB_{m,n+t_n} \\ 1 & \text{if } MB_{m,n} = MB_{m,n+t_n} = MB_{m,n-t_n} \\ 2 & \text{if } MB_{m,n} = MB_{m,n+t_n} \neq MB_{m,n-t_n} \end{cases}$$

Since a separable transform for both pixels and connectivity coefficients is used, a similar rule is obtained for the transform of columns. If a non-scalable motion structure is employed, the connectivity-map transform, as defined by (5.5), results in:

$$\begin{aligned} MB_{m,n}^{(j-r)} &= MB_{2^r m, 2^r n}^{(j)} \\ c_{m,n}^{(j-r)} &= c_{2^r m, 2^r n}^{(j)} \end{aligned},$$

where r is the corresponding reduction in spatial resolution. This relation holds as long as 2^r is smaller than the smallest employed motion block, otherwise the isolated pixel cases has to be accounted for as well. With weighting coefficients defined in this way, the transform of intra coded areas corresponds to their independent coding - borders between intra and inter coded areas are used as points for symmetrical extension of signals.

5.3.3 Postprocessing

In open-loop schemes the quality of motion-compensated frames at low bit-rates decreases because of information loss in both reference and residual frames. Specifically, the structure of motion units, in this case blocks, can become visible. Areas in a reconstructed high-pass frame that are most degraded are those around intra blocks and those where the motion cannot be captured efficiently, due to complicated motion trajectories. Moreover, these artefacts can be propagated from one reconstructed high-pass frame to other frames if an already synthesised high-pass frame is used as a reference frame at lower temporal decomposition levels. While the adaptive spatial transform reduces the spatial influence between differently compensated areas in high-pass frames, motion compensation at low bit-rates still may introduce blocking. The reduction of these visually disturbing artefacts can be prevented by application of the same deblocking filter used in non adaptive coding. The deblocking filter can be applied after each temporal reconstruction step so that the propagation of errors across temporal levels is reduced. This process is here referred to as “in-loop” deblocking filtering. The strength of such filtering should be adaptive because of the different nature of the artefacts. Firstly, the strength of a filter has to depend on the motion activity and motion block modes, *i.e.*, it has to be locally adaptive. As the motion activity and modes in the presented approach are already described by a connectivity-map that is needed for spatial decomposition and reconstruction, this information can be reused for in-loop filtering. Secondly, the strength of the deblocking filter has to depend on the overall bit-rate, *i.e.*, quality reduction, since at higher bit-rates the motion-related artefacts become less visible and therefore the filtering should be weaker. Overall quality in wavelet-based video coding depends on the number of preserved bit-planes which can be also used to adjust the deblocking strength.

The deblocking filter that is proposed here is based on the bilateral filter that has already been used in wavelet based coding [100; 107]. This filter provides local adaptation to the frame content, which additionally improves the selectivity of filtering. Although the 2D bilateral filter is a non-separable filter, the discussion that follows is based on 1D representation, as generalisation to the 2D case is trivial, and to comply with notation in the rest of the chapter.

5.3 Motion-Driven Adaptive Transform (MDAT)

In order to achieve filtering that is locally adaptive with respect to motion and motion mode information, it is applied on pixels that are near the boundaries of non-regularly connected blocks, *i.e.*, at positions k such that $k = m-b+1, \dots, m+b$ and $c_{m,1} < 1$, where b is the width of the filtered area on each side of the block boundary, and c_m is a connectivity-map value of the pixel on the left side of the block boundary. For every k the filtering is performed as:

$$x'_k = f_k^{-1} \sum_{l \in W} x_{k-l} s_l,$$

where x_k and x'_k are respectively original frame pixel and filtered pixel, W is the window with a centre in k from which the neighbouring pixels are taken into account for filtering and s_l are adaptive filter coefficients. f_k normalises the filtered pixel value and is defined as:

$$f_k = \sum_{l \in W} s_l.$$

s_l depends on the distance l of the neighbouring pixel x_{k-l} from the current pixel x_k . Besides the filtered content, s_l is a function of the decoding quality and connectivity information:

$$s_l = \exp \left(-\frac{l^2}{2\sigma_1^2} - \frac{(x_k - x_{k-l})^2}{2\sigma_2^2} \right).$$

In this expression σ_1 is a predefined value and σ_2 is a function of the number of discarded bit-planes b and of the connectivity value related to the motion block boundary. Parameter σ_2 is defined as:

$$\sigma_2 = \alpha 2^{b+c_{m,1}-1},$$

where parameter α is predefined and additionally controls the filter strength, and the nearest block boundary is between m and $m+1$. With the filter defined in this way stronger filtering is applied if the overall frame quality is low. On the other hand, the connectivity-map value $c_{m,1}$ decreases the filter strength in areas where the blocks are better connected ($c_{m,1}$ is close to 1) and provides the strongest filtering for unconnected blocks.

5.3.4 Experiments in SVC Framework

MDAT has been implemented in aceSVC. As the employed motion model is crucial for the detection of intra areas, it will be described briefly in the following. The basic motion units are blocks of variable size, where the macroblock partitioning is described by a tree model [108]. Rate-distortion criterion for tree splitting is used for the motion tree growth, which is performed in a greedy way or the tree is fully grown followed by pruning of the branches based on the coding cost criteria. The following four modes are available for the motion blocks: unidirectional forward and backward, bidirectional and intra. Intra blocks are selected based on error energy criteria; a block is intra if the error for all other three modes is higher than the predefined percentage of the variance of the predicted block. In this way, blocks that cannot be efficiently predicted from reference frames are detected. Also, the threshold for the minimum error energy is set, only above which that rule is applied. This prevents selecting smooth image areas as intra. Additionally, as some sequence parts contain fast camera movements, content of the corresponding frames cannot be compensated efficiently. These cases are detected by the percentage of area that intra blocks occupy in the frame. A threshold of the intra area portion is specified so that, when exceeded, the predicted frame is converted to an intra frame. Also, a simple morphological processing of motion blocks is done to remove small isolated intra and inter areas. As a result, frames that can be efficiently predicted will not contain any intra areas. The optimal setting for all the mentioned parameters are determined experimentally, and are applied in all tests. The reported results have been obtained for three sequences of CIF resolution - “Football”, “Soccer” and “Stefan”. These sequences are selected as they contain objects that are difficult to predict, as well as static or smoothly moving background.

The first experiment investigates the improvement obtained by MDAT with the intra blocks disabled. This mode of employing MDAT is referred to as “inter-MDAT”. This serves the purpose of discerning the gain coming from the introduction of the intra blocks from the gain where only adaptation on inter block boundaries is used. The sequence CIF “Football” at 7.5 Hz has been used, as at this frame rate for this sequence the ME produces motion vector fields having

5.3 Motion-Driven Adaptive Transform (MDAT)

high magnitudes of directional derivative. One level of temporal transform has been performed using a full search ME and with the GOP of 8 frames. GOPs have been encoded independently, meaning that unidirectional motion fields have been used at the GOP boundaries. The adaptive postprocessing filtering setting selected is $\alpha = 0.1$ when OBMC is not used, while no postprocessing is done when the OBMC is enabled. Note that adaptive filtering has been applied for both MDAT and the non-adaptive spatial transform, as in both cases a connectivity-map is created for that purpose. The parameter μ in inter-MDAT has been found experimentally for several sequences to be optimal in the range $0.05 - 0.1$, so the setting $\mu = 0.075$ has been used in this experiment. At this range a good trade-off is achieved between performance lost by not using the correlation that exists in pixels from neighbouring blocks, and the performance gained by avoiding large wavelet coefficients due to discontinuities at the block boundaries. Table 5.1 displays average, minimum and maximum gains over all GOPs for the case when OBMC is disabled, while for the results from Table 5.2 the OBMC has been enabled. The results are obtained for a wide range of bit-rates, in order to demonstrate relatively stable performance gap. From the results presented in Table 5.1 it can be concluded that inter-MDAT achieves an average gain of 0.1 dB. In Table 5.2 it can be seen that the gain that the OBMC alone introduces is quite significant; for some frames, gains of up to 1.5 dB have been observed, and in this case inter-MDAT does not offer improvement. This can be explained by the smoothing effect of the OBMC. The pixels at the boundaries of the blocks are compensated using the weighted contribution of reference frame pixels displaced by motion vectors from the neighbouring blocks, which can effectively smooth out the motion vector field, and consequently the blocking structure introduced by it. Similar behaviour has been observed for the other sequences, with the general conclusion that the OBMC diminishes the gain obtained by inter-MDAT. However, inter-MDAT can still be a suitable replacement for the standard non-adaptive transform when the OBMC is not feasible.

The following experiment has been performed to measure the performance of MDAT when intra blocks are enabled. OBMC has been included in all of the tests, as it has been shown that it introduces significant gain. The conclusion from the previous experiment is that a decrease in performance can be expected when

5.3 Motion-Driven Adaptive Transform (MDAT)

Table 5.1: “Football” CIF 7.5Hz, inter-MDAT, OBMC disabled

bit-rate[kbps]		128	192	256	384	512	768	1024
average PSNR [dB]	Y	28.36	30.16	31.48	33.58	35.34	38.11	40.48
	U	32.04	34.13	35.58	37.55	38.97	41.47	43.41
	V	35.41	36.88	37.84	39.48	40.61	42.72	44.23
average MDAT gain	Y	-0.02	+0.02	+0.04	+0.09	+0.10	+0.12	+0.10
	U	+0.11	+0.01	+0.07	+0.05	+0.05	+0.15	+0.09
	V	-0.01	+0.02	+0.11	+0.09	+0.08	+0.12	+0.07
maximum MDAT gain	Y	+0.01	+0.13	+0.10	+0.16	+0.36	+0.23	+0.26
	U	+0.76	+0.08	+0.23	+0.31	+0.29	+0.63	+0.24
	V	+0.09	+0.12	+0.33	+0.33	+0.28	+0.71	+0.27
minimum MDAT gain	Y	-0.05	-0.01	-0.01	+0.04	+0.02	+0.05	-0.12
	U	-0.03	-0.07	-0.06	-0.10	-0.15	-0.02	-0.06
	V	-0.10	-0.05	-0.04	-0.05	-0.02	-0.03	-0.05

Table 5.2: “Football” CIF 7.5Hz, inter-MDAT, OBMC enabled

bit-rate[kbps]		128	192	256	384	512	768	1024
average PSNR [dB]	Y	28.54	30.41	31.79	34.02	35.81	38.71	41.19
	U	32.35	34.47	36.01	38.13	39.57	42.14	44.03
	V	35.54	37.06	38.28	39.90	41.10	43.27	44.82
average MDAT gain	Y	-0.09	-0.10	-0.09	-0.10	-0.08	-0.12	-0.12
	U	-0.06	-0.10	-0.14	-0.14	-0.28	-0.09	-0.12
	V	-0.02	-0.05	-0.12	-0.09	-0.19	-0.09	-0.11

using MDAT coupled with OBMC on other boundaries except the intra-inter ones. Therefore, the MDAT has been disabled for boundaries between the inter areas with the setting $\mu = 0$. This mode of employing MDAT is referred to as “intra-MDAT”. In this case deblocking is used only for intra-inter block boundaries, again for both MDAT and non-adaptive transform. The results for the same sequence as from from the previous experiment are shown in Table 5.3. The gain

5.3 Motion-Driven Adaptive Transform (MDAT)

Table 5.3: “Football” CIF 7.5Hz, intra-MDAT, OBMC enabled

bit-rate[kbps]		128	192	256	384	512	768	1024
average MDAT gain [dB]	Y	+0.83	+0.65	+0.62	+0.59	+0.67	+0.69	+0.69
	U	+0.71	+0.48	+0.27	+0.41	+0.40	+0.60	+0.72
	V	+0.44	+0.26	+0.13	+0.36	+0.30	+0.51	+0.57

of up to 0.7 dB for intra-MDAT can be observed on all bit-rates. To justify the need for MCTF at this frame rate, the performance has been also compared to the case when motion compensation is not performed, *i.e.*, when the sequence is processed with the lazy wavelet in order to obtain a sequence at the lower frame rate of 3.75 Hz. The conclusion of the few experiments conducted is that MCTF with MDAT indeed achieves better performance on all bit-rates, and that PSNR gain over lazy wavelet increases up to 0.5 dB for all three components on bit-rates higher than 1 Mbps. Thus, MDAT combines the advantages of both choices, as parts of some frames can be efficiently compensated, while others contain very little correlation with the neighbouring frames and cannot be compensated.

The following tests have been obtained for full sequences of 15 Hz frame-rate. The “sliding window” MCTF has been used, meaning that the frames can be predicted from either neighbouring frame or from both, on all temporal levels regardless of their position in a GOP. All decoded bit-streams, for all reported decoding points, are obtained by extraction from the same compressed bit-stream. Fig. 5.13, Fig. 5.14 and Fig. 5.15 contain results for three decoding points for the Y component, each one represented with four bit-rates. In Fig. 5.13 the results for quality scalability only are shown. It can be seen that MDAT achieves significant gain in most of the decoding points, for the sequence “Football” of up to 0.78 dB for bidirectional prediction. This sequence is the most suitable for MDAT, as it contains the fastest motion, thus making the intra blocks necessary in most parts of the sequence. The gain obtained for the two other sequences is more modest, as the adaptation is done in fewer frames, since a large part of these sequences can be efficiently compensated. Temporal scalability results for adaptation to half the original frame rate are shown in Fig. 5.14, where even higher gains can be observed. To test the spatial scalability, the references have

5.3 Motion-Driven Adaptive Transform (MDAT)

been obtained by spatial downsampling with the 9/7 wavelet on the required frame rate. This avoids mismatch problems inherent in the $t+2D$ architectures caused by inverse MCTF on lower spatial resolutions. Therefore this represents a fair choice as neither of the tested approaches can achieve perfect reconstruction of the generated reference. The results for combined spatio-temporal scalability, for adaptation to QCIF resolution and half of the original frame rate, are shown in Fig. 5.15. Generally, MDAT achieves gain on all tested bit-rates, where more significant gains of more than 2 dB are observed on low bit-rates. The results for unidirectional motion model are also presented, from which it can be seen that in that case MDAT achieves slightly higher gains than for bidirectional motion model, as even larger parts of frames cannot be accurately predicted. It should be noted that in some cases the non-adaptive transform with intra blocks performs better than when intra blocks are not used, but using the adaptive transform always offers better performance when compared to these other two cases. The main drawback of the non-adaptive transform applied on intra blocks is that it causes serious ringing artefacts at the boundaries of intra and inter coded areas.

In Fig. 5.12 a visual comparison of a single decoded frame is shown. Here a low bit-rate is selected in order to make the artefacts more noticeable. It can be seen that employing the intra blocks helps in isolating areas that cannot be compensated efficiently, since when the intra blocks are not used the motion compensation results in annoying visual artefacts caused by high energy wavelet coefficients of poorly predicted pixels. The effect of MDAT is that there is no leaking of energy between these areas, each of which is processed separately. Also, by using MDAT coupled with adaptive postprocessing the boundaries between inter and intra areas are efficiently smoothed out and not visible.

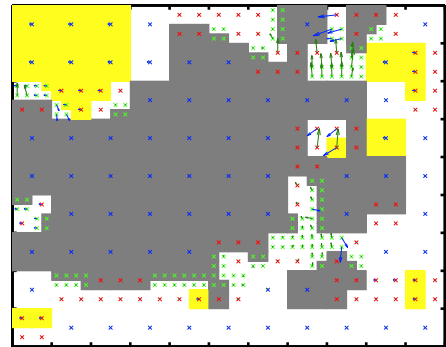
5.3 Motion-Driven Adaptive Transform (MDAT)



(a) non-adaptive transform, Y component PSNR of 26.07 dB



(b) MDAT, Y component PSNR of 26.97 dB



(c) MV field used in MDAT

Figure 5.12: Visual comparison of frame 58 in the decoded sequence CIF “Football” 7.5 Hz at 192 kbps. In the displayed motion vector field the gray areas represent intra blocks, while yellow represent bidirectional and white unidirectional (forward and backward). The centre of the motion block is marked with “x”, where the size of the block is colour-coded. Motion vectors belonging to the forward field are displayed in blue, while the ones belonging to backward field are displayed in green.

5.3 Motion-Driven Adaptive Transform (MDAT)

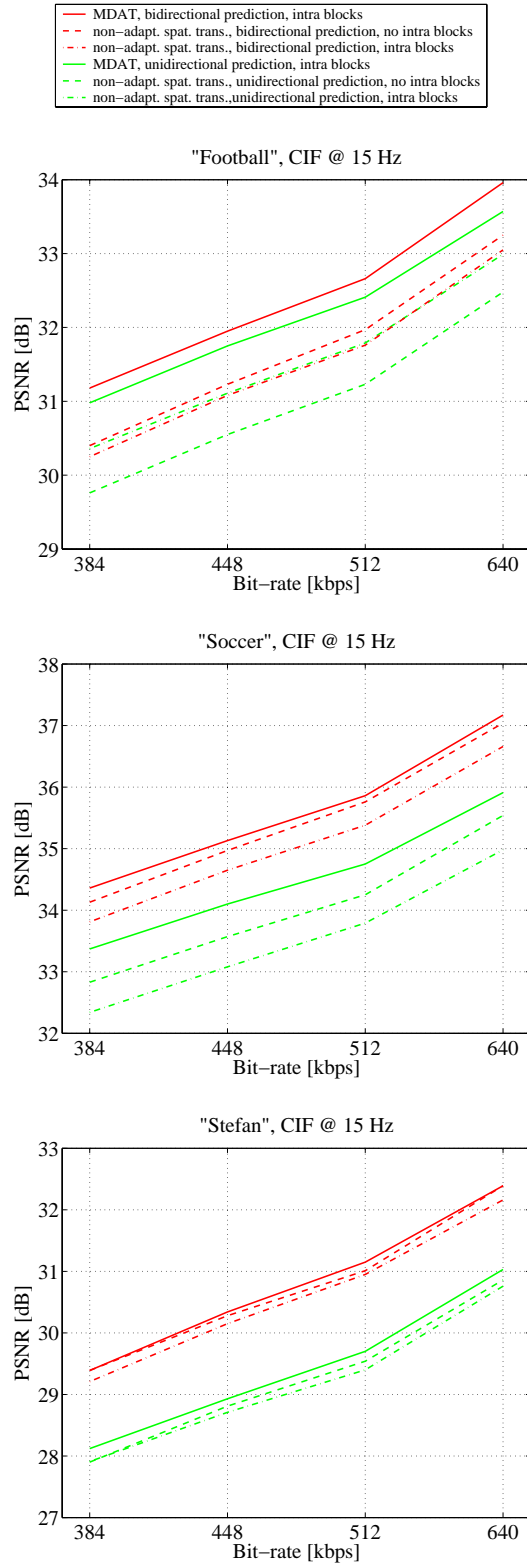


Figure 5.13: PSNR results for three sequences and decoding point (CIF, 15Hz).

5.3 Motion-Driven Adaptive Transform (MDAT)

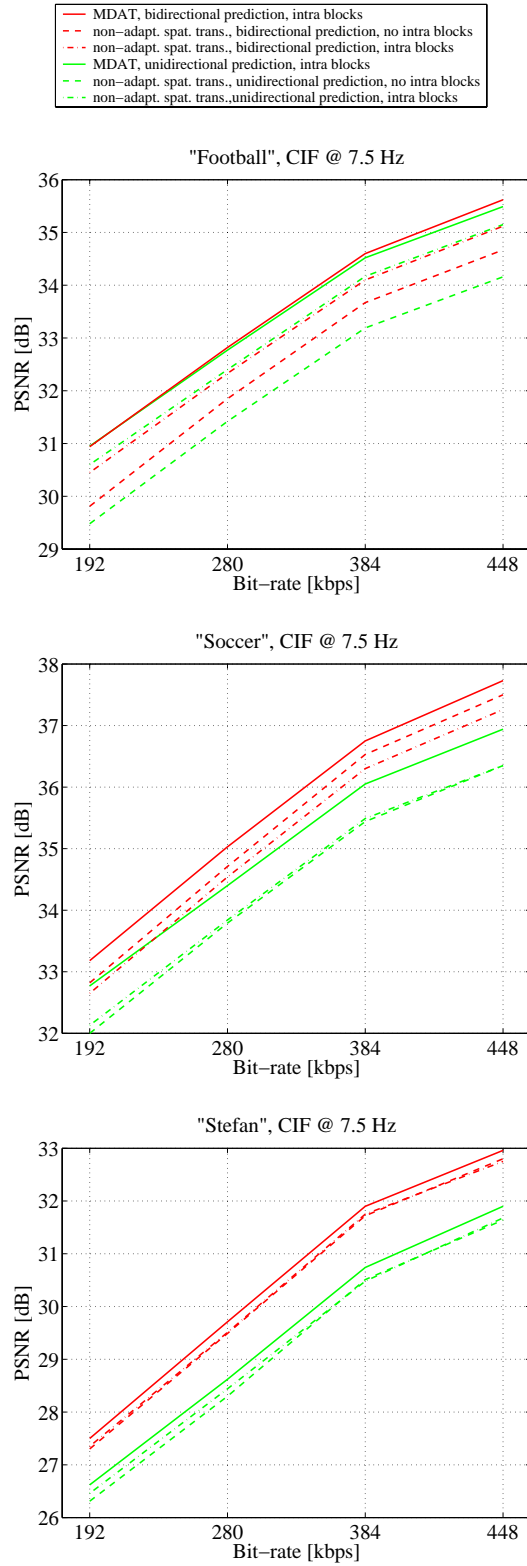


Figure 5.14: PSNR results for three sequences and decoding point (CIF, 7.5Hz).

5.3 Motion-Driven Adaptive Transform (MDAT)

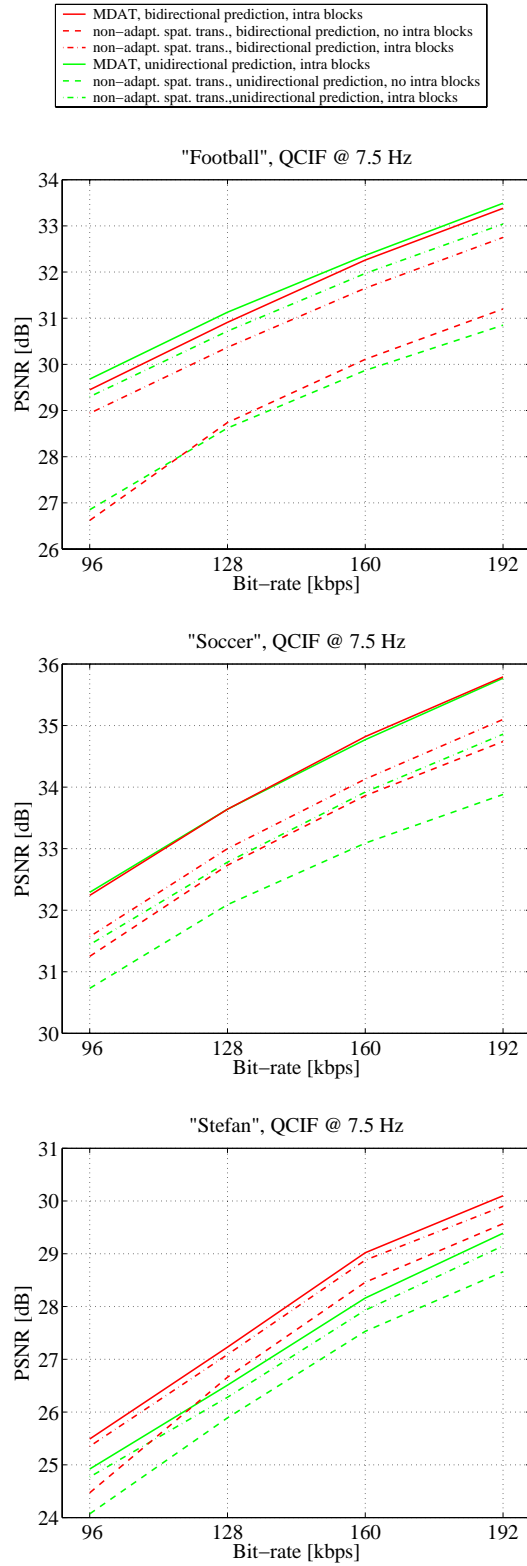


Figure 5.15: PSNR results for three sequences and decoding point (QCIF, 7.5Hz).

5.4 Object-based Coding Using Connectivity-Map Decomposition

In this section another potential application of the joint wavelet connectivity-map decomposition is presented, namely in the object-based coding. In that case, on the boundaries of objects the corresponding connectivity-map is set to disconnections. An example is provided in Fig. 5.16(b), which is a representation of connectivity-map of the image Fig. 5.16(a). This map has been obtained by a semi-automatic object detection tool in a popular image processing software “Gimp”. The result of analysis, shown in Fig. 5.16(c), is obtained by using relations (5.5). To assess the performance achievable with the image compression coder based on joint wavelet connectivity-map decompositions, a scenario that simulates a possible implementation has been devised. SPIHT was used for coding of wavelet coefficients [80], while the connectivity-map was compressed in a non-scalable way with the Portable Network Graphic (PNG) coder. PNG was chosen due to its efficient algorithm for lossless coding of binary images, although a much more efficient method could be used, that is based on arithmetic coding with adaptive context modelling, such as [109]. The image was compressed by encoding several most significant bit-planes, indicated with the equivalent quantisation step q . The obtained bit-rates R are expressed in *bits per pixel* (bpp). From the results shown in Fig. 5.17 and Fig. 5.18, it can be observed that the compression based on connectivity-maps, besides 0.1 – 0.4 dB higher PSNR, achieves a better subjective performance, as the ringing artefacts around the object’s silhouette do not appear.

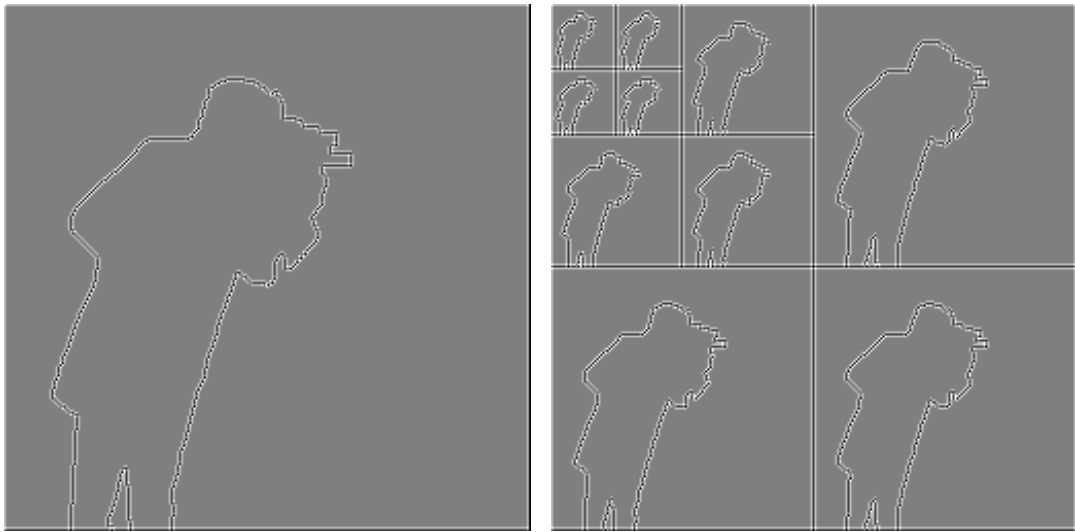
The connectivity-map that defines objects in a scene could be also indispensable in semantic analysis of images, as it would relieve the computational burden of the post-compression analysis steps. In that case the connectivity-map can be regarded as a metadata that is already embedded in the compressed bit-stream. This could be particularly beneficial when the analysed image has been previously adapted to a lower bit-rate, as in that case the information provided by the connectivity-map would be of much higher precision than any information that could be extracted from the low-quality version of the image itself.

There are numerous other techniques that have been proposed for wavelet

5.4 Object-based Coding Using Connectivity-Map Decomposition



(a) original image



(b) connectivity-map of the image

(c) result of the connectivity-map decomposition

Figure 5.16: Still image “Cameraman” and the corresponding connectivity-maps.

coding of arbitrary shapes, many of which are summarised in [110], where also a new technique is proposed, called Region-Based Discrete Wavelet Transform (RBDWT). The differences between them basically can be reduced to the different solutions for signal boundary treatment, that depend on symmetry properties of the employed wavelet, or to the different ways of translating the areas of

5.4 Object-based Coding Using Connectivity-Map Decomposition



(a) $q = 256$, $R = 0.08$ bpp, PSNR = 20.25 dB (b) $q = 128$, $R = 0.16$ bpp, PSNR = 22.98 dB



(c) $q = 64$, $R = 0.30$ bpp, PSNR = 26.43 dB (d) $q = 32$, $R = 0.55$ bpp, PSNR = 30.05 dB

Figure 5.17: Compression of still image “Cameraman” using non-adaptive wavelet transform.

arbitrary shapes to rectangularly shaped areas. The object-based coding has been recognised as an important factor in the future compression applications, and is incorporated in the MPEG-4 standard.

Given that the example provided here is just a model of the possible imple-

5.4 Object-based Coding Using Connectivity-Map Decomposition



(a) $q = 256$, $\Delta R = -0.00026$ bpp, PSNR = 20.65 dB (b) $q = 128$, $\Delta R = -0.006$ bpp, PSNR = 23.32 dB



(c) $q = 64$, $\Delta R = -0.0165$ bpp, PSNR = 26.69 dB (d) $q = 32$, $\Delta R = -0.028$ bpp, PSNR = 30.12 dB

Figure 5.18: Compression of still image “Cameraman” using joint wavelet connectivity-map decomposition. The bit-rate of the connectivity-map itself is 0.06 bpp, and is not included in the bit-rates shown. ΔR denotes the difference in the obtained bit-rates to the bit-rates shown in Fig. 5.17.

5.4 Object-based Coding Using Connectivity-Map Decomposition

mentation, with sub-optimal entropy coder that is not adjusted to this specific source, the conclusion is that connectivity-map provides a promising technique for scalable object-based wavelet still image and video coding.

Chapter 6

Conclusions

The research presented in this thesis is focused around the objective of developing a flexible scalable video coding framework. The framework has its practical application in the aceMedia system, which includes a media distribution component that requires high degree of content adaptability in order to enable wide range of portability options. At the initial stages of the development, wavelet transform was chosen as a main tool as it provides flexibility in design as well as high compression needed for content transmission and storage. Since its inception, the development of aceSVC has been progressing in parallel with the new concepts in wavelet based scalable video coding, often by adopting and extending newly proposed techniques. The aim of this thesis is to summarise different theoretical and practical aspects that have to be considered in the development of a wavelet based SVC framework, to present novel approaches and conclusions in this field and to present the developed aceSVC architecture. In this concluding chapter, a brief overview of the thesis is given, followed by a list of potential research topics that have been recognised as the most promising for further performance improvements and introduction of additional functionalities.

The thesis begins with an informative summary of some of the well-known concepts in wavelet theory, and concentrating on the wavelet properties and functionalities needed for their application in an efficient SVC system. Then the most recent innovations in SVC architectures are described, providing the motivation for techniques developed for aceSVC. More precisely, the concept of GSTS has been selected as a base for aceSVC. The formalised concept of the generalised

spatio-temporal decompositions was used to optimise the decomposition path to the required spatio-temporal decoding points. It has been shown that in this case improved averaged compression performance is achieved. Moreover, a special care has been taken in investigating the impact that the selection of the temporal wavelet filter has on the compression performance. With the devised post-compression optimisation method the distortion can be efficiently distributed between frames with the effect of the decreased PSNR variation across the reconstructed sequence. Compared to the other SVC state-of-the-art codecs, the aceSVC achieves remarkably good results.

A scheme for adapting the 2D wavelet transform accordingly to the available motion information - Motion-Driven Adaptive Transform (MDAT) has been proposed. The proposed scheme was motivated by the observation that temporal frames have spatially diverse characteristics as a result of motion compensation. Two different modes of MDAT, that represent special cases of possible motion vector fields, have been analysed: inter-MDAT, that performs adaptation only between inter areas, and intra-MDAT, that performs adaptation only on intra-inter boundaries. The intra-MDAT scheme offers better performance and adds only negligible complexity to the spatial transform. The proposed scheme has been tested in a challenging SVC environment and its superiority over non-adaptive transform has been confirmed. Another benefit of intra-MDAT is in the increased temporal scalability range, as it can detect whole frames and single frame areas where temporal filtering is not effective and adaptively perform filtering on the other areas thus efficiently producing a lower frame rate temporal subband. It should be noted that although it has been tested only in $t+2D$ framework the MDAT scheme with connectivity-map can be used in any other coding system based on MCTF and spatial wavelet transform, and also can be used together with scalable motion structures. However, in that case only the lowest layer of motion structure should be used for generating the connectivity-map. Another potential use of the connectivity-map is the object-based scalable still image or video compression. One possible approach for connectivity-map coding can be achieved by using discretised connectivity values, so a look-up table instead of using arithmetic operations would be used in transform, and finally a highly efficient context-based entropy coder could be employed for compressing the map.

In addition to the biorthogonal wavelets, a modification of the connectivity map for the compression-efficient orthogonal wavelets could be devised [111], based on the recently proposed signal extension methods [46].

Although the research on scalable video coding based on wavelets has produced many efficient techniques, there are still many areas for improvement as some issues have not been yet adequately resolved. While some of the issues are well-known from the previous research, some have become apparent only during the work on this thesis. To summarise, some of the key open topics include the following:

- Since the range of the embedded frame dimensions in the scalable video is determined by the employed wavelet transform, only the dyadic frame ratios have been used so far. This may cause a suboptimal performance in applications where the supported frame sizes are not dyadically related. Since the closest higher or lower available frame size would have to be transmitted instead, followed by downsampling or upsampling to the target device display size, this would result in inefficient utilisation of the bandwidth. A potential approach to solve this would be to employ M -band wavelets supporting synthesis at the rational frame size ratios [112].
- Wavelet filterbanks that are good for compression are not good for downsampling, and vice-versa, which has a direct negative consequence in spatial scalability performance. A possible solution would be to employ some of the redundant schemes, for instance the multi-scale pyramidal architecture, since they do not impose restrictions on the employed downsampling filter. Another approach could be to design a wavelet transform with variable downsampling rate, which will provide more degrees of freedom in the filterbank design and thus trade-off the controlled redundancy for improved downsampling performance. Possible approaches include the *frequency-warped* wavelets [113], or the already mentioned M -band wavelets coupled with the previously applied upsampling.
- Non-dyadic temporal scalability for achieving temporal scaling factors other than 2. For instance, in [114] a factor of 3 is achieved by employing three-

band filtering. However, there is no common framework for non-dyadic temporal decomposition scheme and arbitrary frame-rate scaling.

- Scalability of motion information, which is necessary for extending the operational bit-rate range of the scalable video coders. The non-separability of the 3D spatio-temporal decomposition poses a great difficulty in scalable coding of the motion vector field. Although there have been many proposed methods, for instance [115; 116; 117; 118; 119; 120], the main problem remains to accurately estimate the effect that motion scalability has on the overall distortion.
- The locally homogeneous motion models, like the block-based models, do not work well when coupled with the wavelet transform, which works globally on the whole frame. The difficulties arise from different connectivity states of the pixels, that decrease the coding gain of filtering with longer temporal filters and as well make the distortion optimised rate allocation more complex. Therefore new motion models, devised specifically for 3D wavelet coding should be developed to overcome this problem.

As a final note, the Matlab[®] software used to produce the wavelet-related figures presented in this thesis can be found on the author's web-site [121].

Appendix A - Wavelets

A.1 Wavelet and Scaling Functions and Filters

This section presents the wavelet filters used in this thesis. The filterbank coefficients are plotted, namely analysis low-pass h_0 , analysis high-pass h_1 , synthesis low-pass g_0 and synthesis high-pass g_1 . It should be noted that the downsampling lattice (e, e) was used in displaying of the filter coefficients, so the symmetric wavelets have the centre of symmetry at filter coefficient $n = 0$. Also the scaling functions, analysis $\tilde{\varphi}(t)$ and synthesis $\varphi(t)$, and wavelet functions, analysis $\tilde{\psi}(t)$ and synthesis $\psi(t)$, have been generated, using the procedure that can be obtained by using the basic recursion relations (2.4), (2.7), (2.11) and (2.12). For example, if in (2.4) we let that $j \rightarrow \infty$ then $\varphi_k^{(j)}(t) \rightarrow \delta(k)$, where $k \in \mathbb{R}$, so that scaling functions span $L^2(\mathbb{R})$. This also can be interpreted as if the scaling function is equivalent to the sampling function at some scale j_0 , and then by iterative substitution back into (2.4) scaling functions of the coarser scale can be obtained. This turns to be equivalent to (2.17), for $\varsigma = 0$. Similarly, using (2.7) the relation (2.17) for $\varsigma = 1$ follows. The dual relations (2.11) and (2.12) are then used in the same way to obtain (2.19). Specifically, in the following exactly 10 iterations were used to generate the scaling and wavelet functions, as this resolution is sufficient to represent the continuous plot. Wavelets with the delta low-pass filter are not included as the corresponding wavelet and scaling functions are trivial, *i.e.*, since $h_{0,n} = \delta_n$ they are identical at all scales.

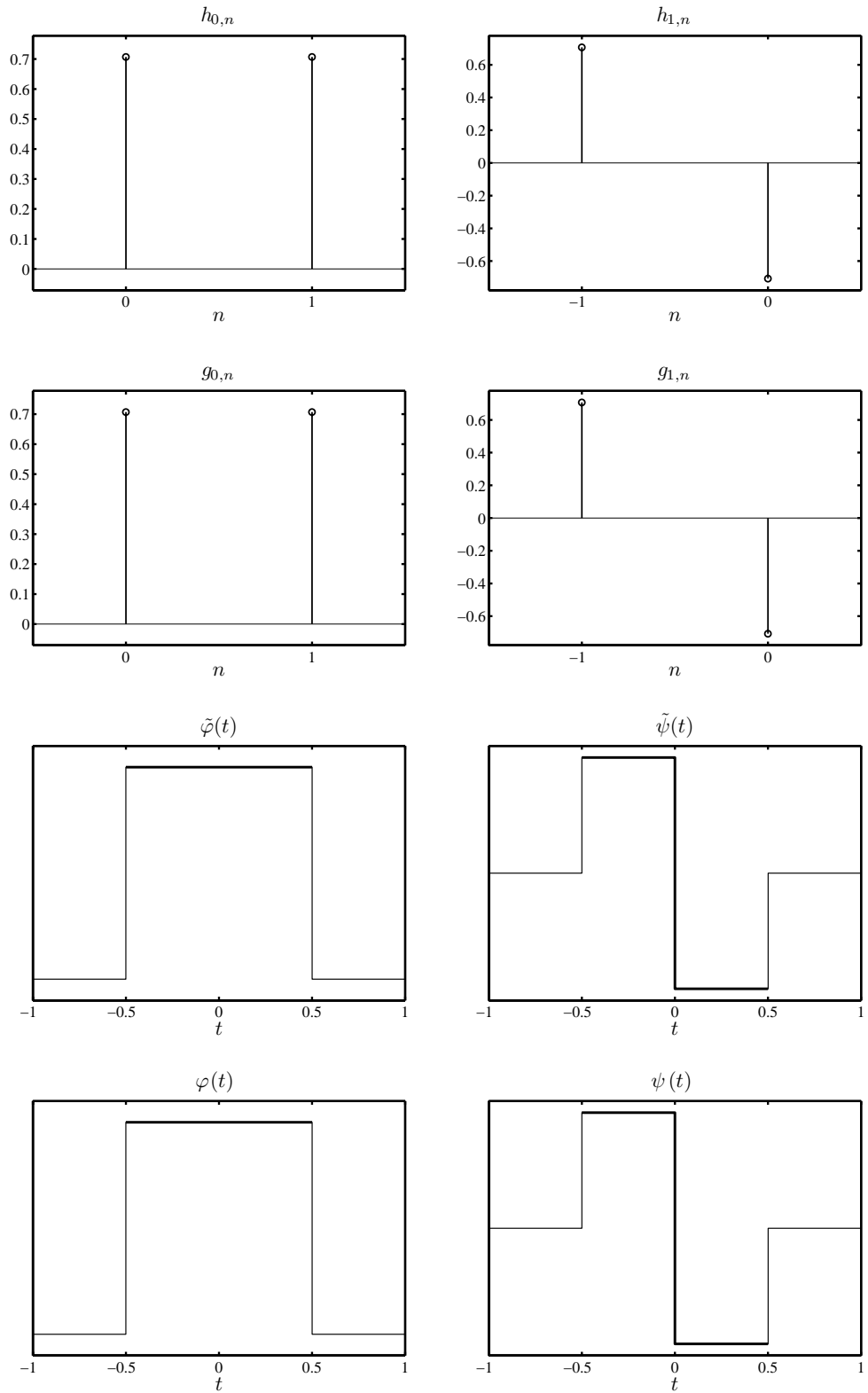


Figure 6.1: Haar 2/2 wavelet.

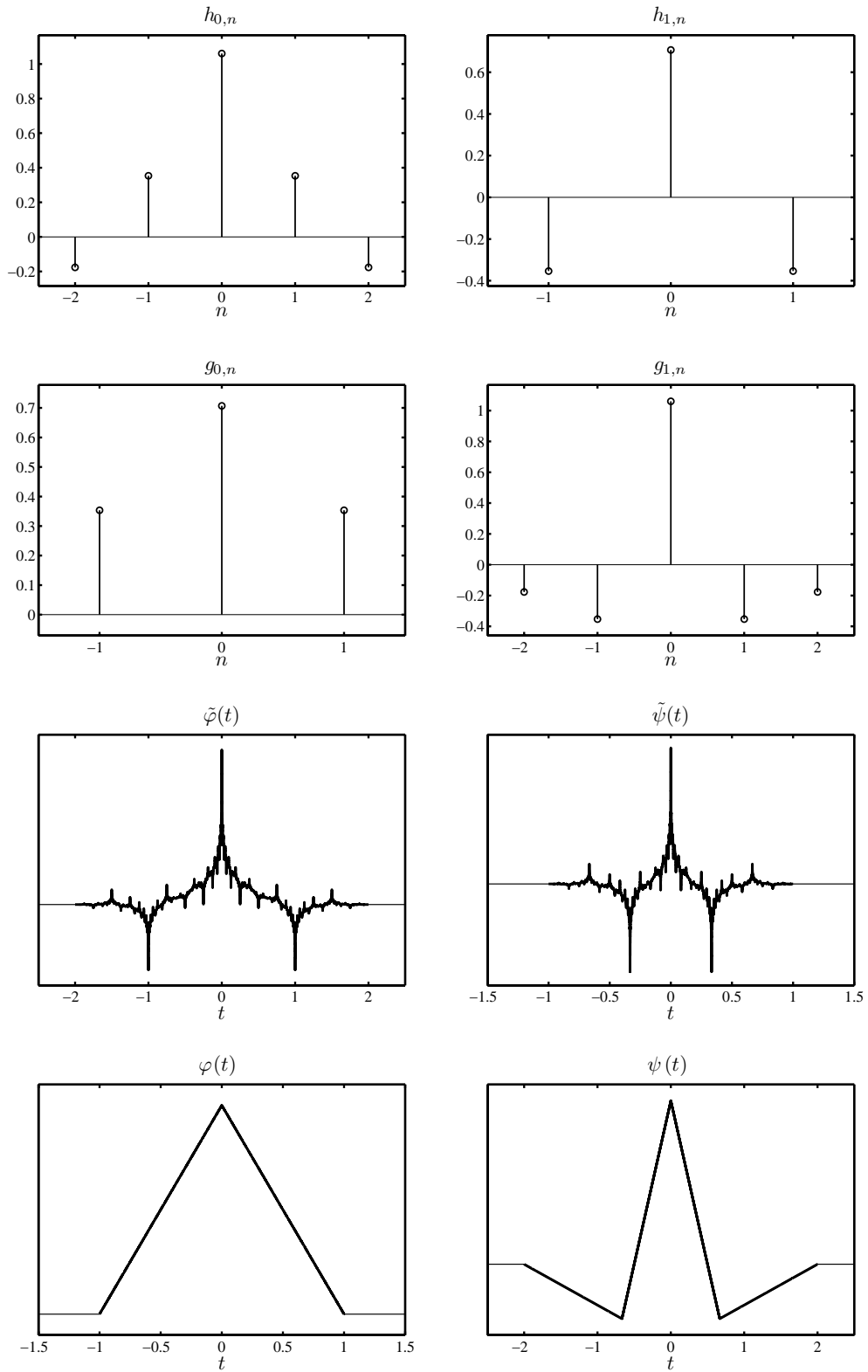


Figure 6.2: Le Gall 5/3 wavelet.

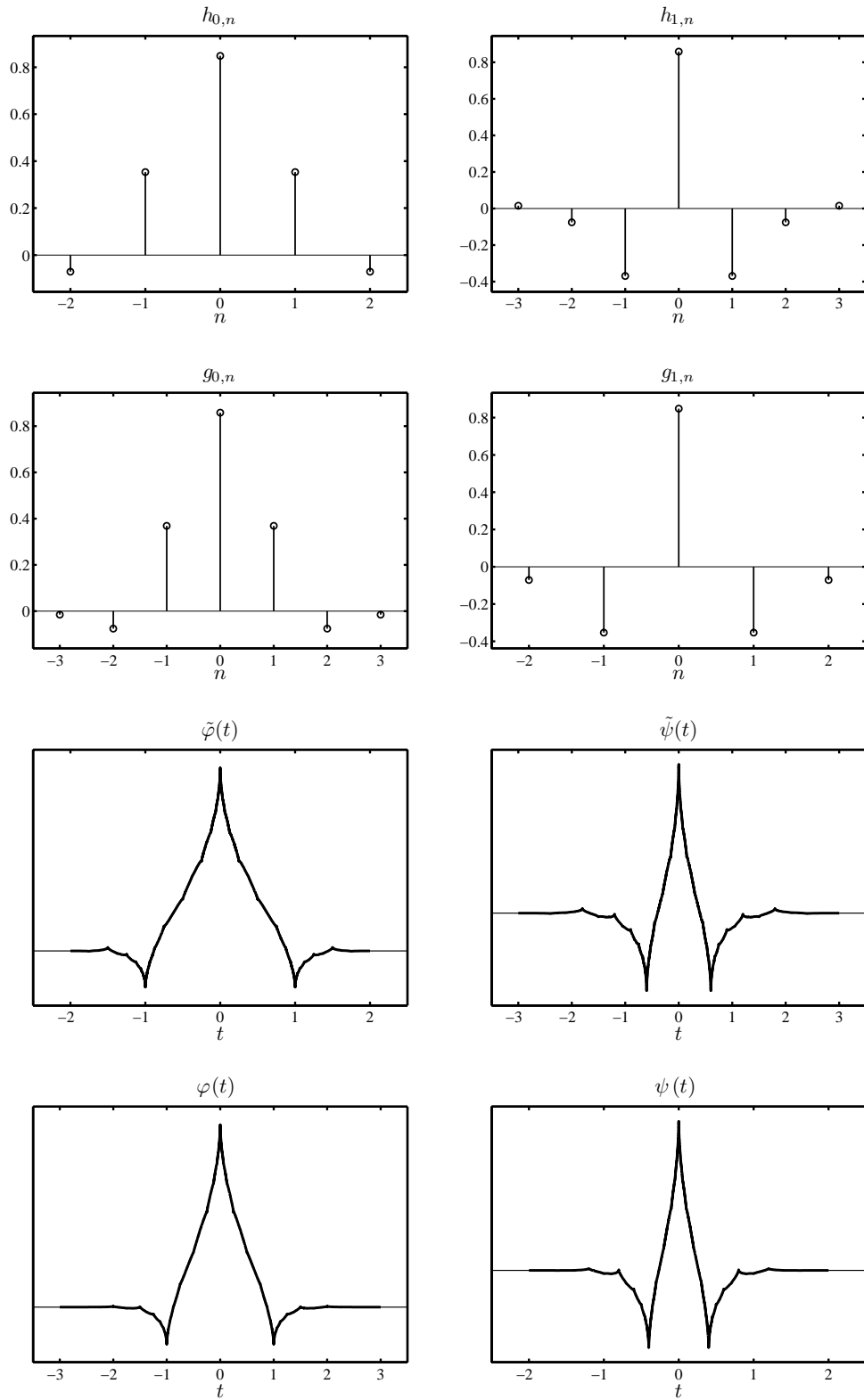


Figure 6.3: Burt-Adelson 5/7 wavelet.

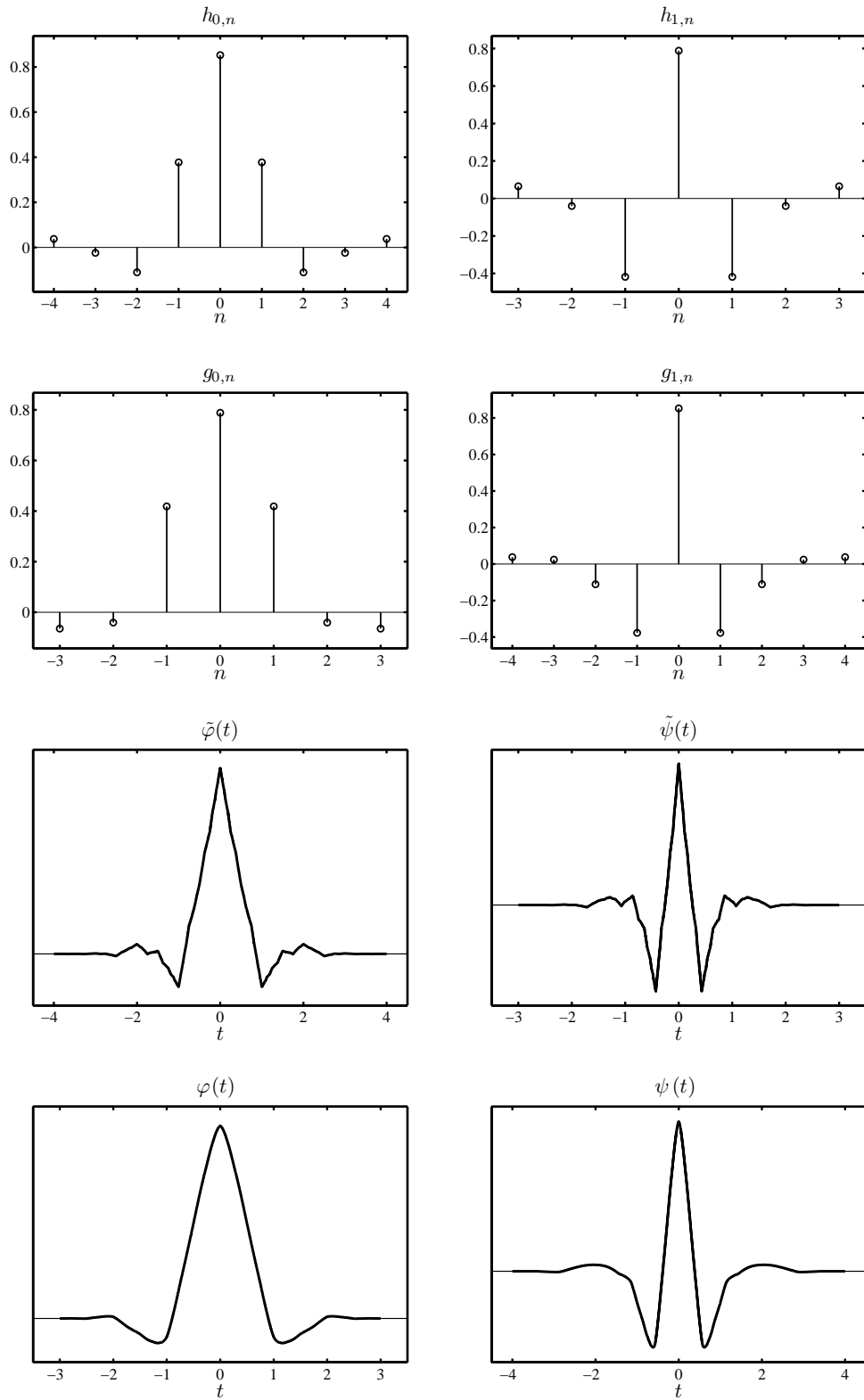


Figure 6.4: Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet.

A.2 Frequency and Phase Characteristics

This section presents the frequency and phase characteristics of the wavelets from the previous section. Frequency plot, besides the analysis low-pass and high-pass filters, represented with $|H_0(z)|$ and $|H_1(z)|$, respectively, shows the power spectrum density of the analysis filterbank, $S(z)$. It is defined as $S(z) = (|H_0(z)|^2 + |H_1(z)|^2)/2$, and since for orthogonal filters $S(z) = 1$, the closer the particular $S(z)$ is to 1 the closer is the corresponding wavelet to being orthogonal. Second plot shows the phase characteristics of the filters, specifically, it shows $\arg(H_0(z))$ and $\arg(H_1(z))$. It is interesting to note that the Haar wavelet is the only orthogonal wavelet with the linear phase, which is a characteristic of all symmetric wavelets.

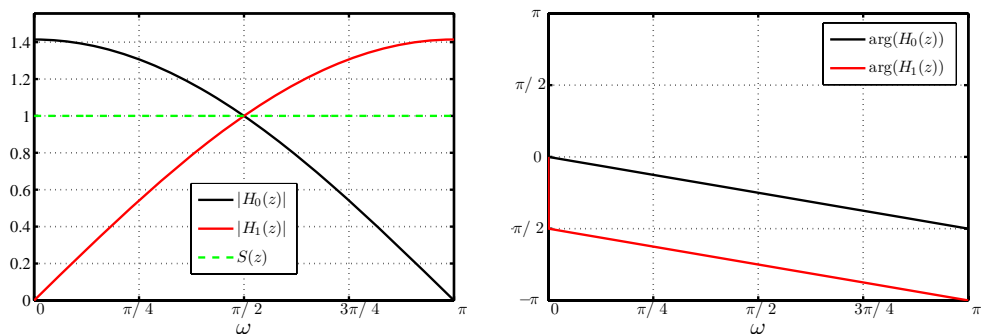


Figure 6.5: Haar - frequency and phase characteristic.

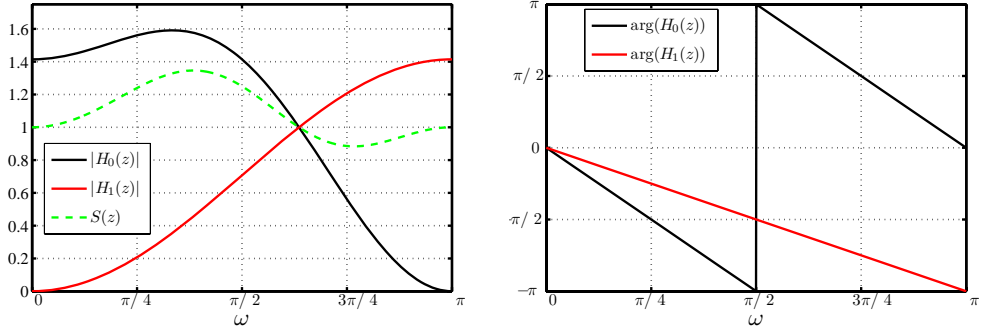


Figure 6.6: Le Gall 5/3 - frequency and phase characteristic.

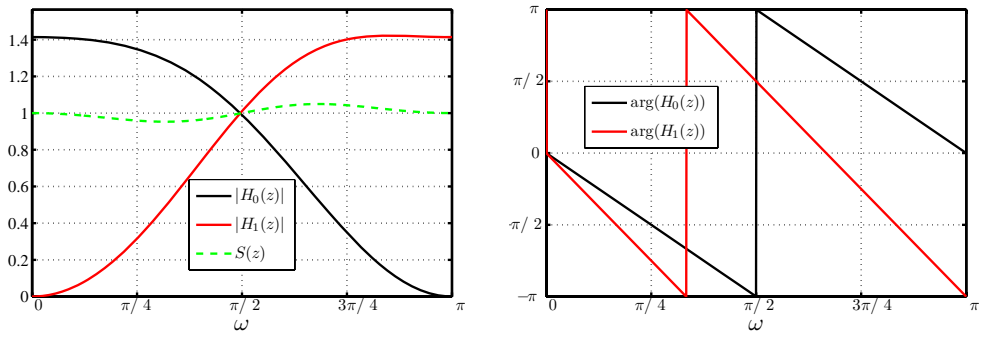


Figure 6.7: Burt-Adelson 5/7 - frequency and phase characteristic.

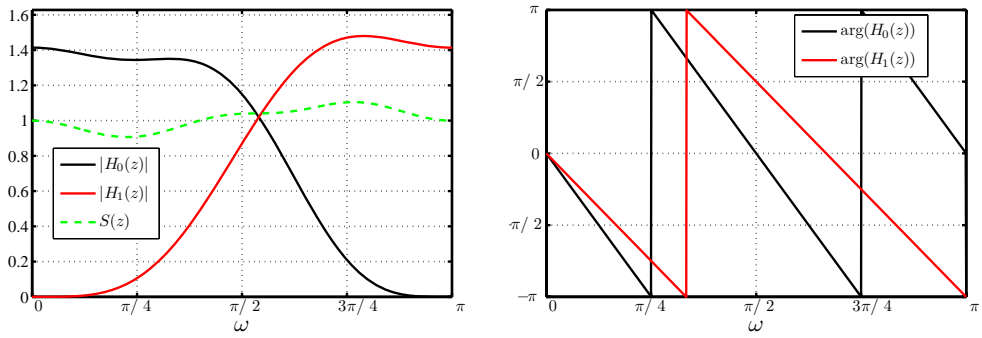


Figure 6.8: Cohen-Daubechies-Feauveau (CDF) 9/7 - frequency and phase characteristic.

A.3 Properties and Lifting Coefficients

This section presents the properties of used wavelets that are important for their application in SVC. Specifically, for each wavelet the following is computed:

- number of vanishing moments, (2.35);
- L_2 norms of the synthesis filters, (2.37);
- incremental BIBO gains of the analysis filters, (2.40);
- Peak CSRE Ratio, (2.43).

Haar wavelet is omitted as it is orthogonal and presents a trivial case, where for all j the L_2 norms and PCR are equal to 1, while the incremental BIBO gains are $\sqrt{2}$.

Table 6.1: Le Gall 5/3, $(V_{\tilde{\psi}}, V_{\psi}) = (2, 2)$

$l = j_0 - j$	$\ g_0^{(j)}\ $	$\ g_1^{(j)}\ $	$\Delta BIBO(L_j)$	$\Delta BIBO(H_j)$	$PCR(l)$
1	0.86603	1.19896	2.12132	1.41421	1.91667
2	0.82916	0.96014	1.53206	1.17851	2.33750
3	0.81968	0.89049	1.46861	1.19664	2.65104
4	0.81729	0.87221	1.42158	1.17524	2.81687
5	0.81670	0.86757	1.42286	1.17546	2.90717

Table 6.2: Burt-Adelson 5/7, $(V_{\tilde{\psi}}, V_{\psi}) = (2, 2)$

$l = j_0 - j$	$\ g_0^{(j)}\ $	$\ g_1^{(j)}\ $	$\Delta BIBO(L_j)$	$\Delta BIBO(H_j)$	$PCR(l)$
1	1.01048	0.98995	1.69706	1.77787	1.00363
2	1.01805	1.00324	1.43778	1.38391	1.00469
3	1.02110	1.01514	1.38755	1.31788	1.00562
4	1.02214	1.02011	1.41755	1.33072	1.00617
5	1.02246	1.02182	1.41476	1.32628	1.00657

In the following, the lifting steps of used wavelets are displayed. The lifting steps correspond to the definition of lifting as in (2.47) and (2.48). The

Table 6.3: Cohen-Daubechies-Feauveau 9/7, $(V_{\tilde{\psi}}, V_{\psi}) = (4, 4)$

$l = j_0 - j$	$\ g_0^{(j)}\ $	$\ g_1^{(j)}\ $	$\Delta BIBO(L_j)$	$\Delta BIBO(H_j)$	$PCR(l)$
1	0.99144	1.02002	1.95211	1.83513	1.22101
2	1.01519	0.98347	1.36549	1.34483	1.30426
3	1.02572	1.01962	1.38657	1.35364	1.37387
4	1.02882	1.03688	1.40998	1.33675	1.41392
5	1.02964	1.04204	1.41130	1.33649	1.44002

z-transforms of the even indexed lifting steps, $\lambda_{2k}(z)$, correspond to the prediction steps, while of the odd indexed, $\lambda_{2k+1}(z)$, correspond to the update steps. Therefore, $\lambda_{2k}(z)$ correspond to the polyphase signal component containing odd indexed signal samples, while $\lambda_{2k+1}(z)$ to the component containing even indexed signal samples. Since (e, o) downsampling lattice is used, and the signal polyphase components are mutually delayed by one sample, the z-transform representation of the neighbouring samples for the predicted polyphase component is $(1 + z)$, while for the updated component it is $(z^{-1} + 1)$. First term refers to the pixel on the “left” side of the lifted pixel, while the second to the pixel on the “right” side.

Table 6.4: Lifting coefficients

lift. step	Haar	5/3	5/7	9/7
$\lambda_0(z)$	-1	$-1 \cdot (1 + z)$	$-0.2 \cdot (1 + z)$	$-1.586134342 \cdot (1 + z)$
$\lambda_1(z)$	0.5	$0.25 \cdot (z^{-1} + 1)$	$0.357142857 \cdot (z^{-1} + 1)$	$-0.052980118 \cdot (z^{-1} + 1)$
$\lambda_2(z)$			$-0.21 \cdot (1 + z)$	$0.882911075 \cdot (1 + z)$
$\lambda_3(z)$				$0.443506852 \cdot (z^{-1} + 1)$

Appendix B - Subband Weighting Tests

B.1 Distortion estimation

In this section, the results of distortion estimation accuracy evaluation are presented for two sequences and selected subbands. Measured values are defined with relations (4.5), and are calculated for the range of bit-planes b , from the highest to $b = 0$. Results for selected spatio-temporal subbands are displayed, for each colour component (Y, U and V), averaged over all frames in the sequence, with the confidence interval of two standard deviations indicated. It should be noted that although in the developed codec the three spatial subbands HL, LH and HH are always processed jointly, and hence their distortions are accumulated into the same variable, here they are shown separately in order to demonstrate similarities as well as differences between their statistics.

1. “City”, 4CIF, 60 Hz, intra-only coding, 600 frames, 5 spatial levels

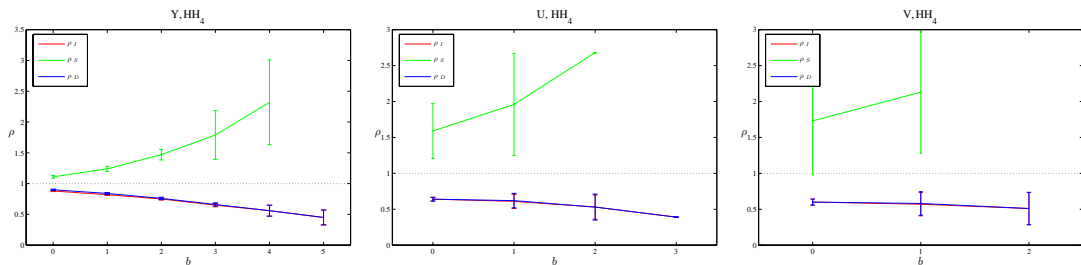


Figure 6.9: Accuracy of distortion estimation for subband HH_4 .

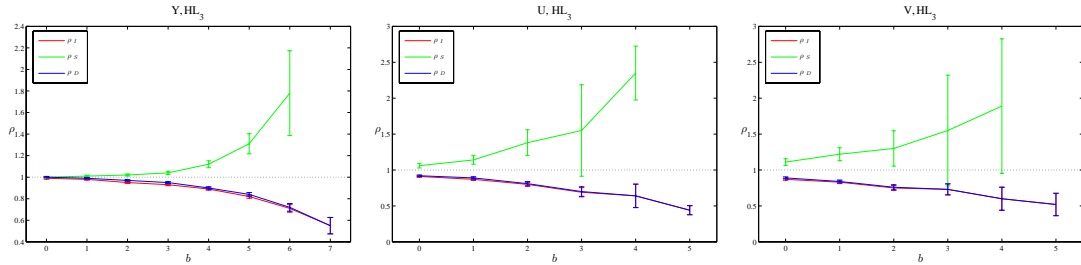


Figure 6.10: Accuracy of distortion estimation for subband HL₃.

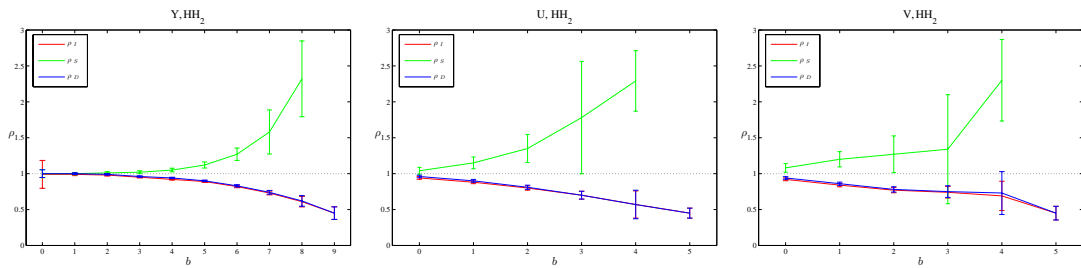


Figure 6.11: Accuracy of distortion estimation for subband HH₂.

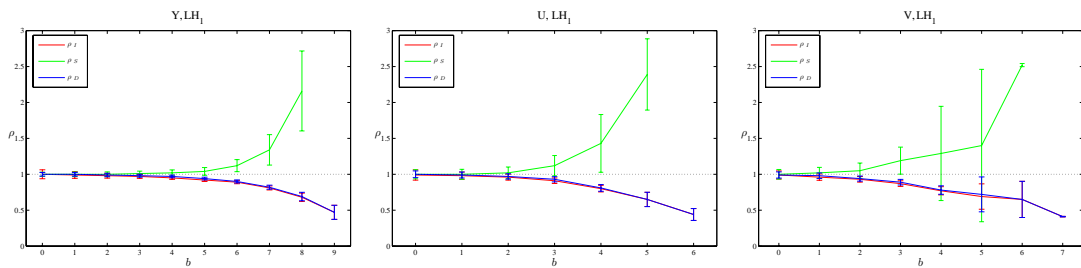


Figure 6.12: Accuracy of distortion estimation for subband LH₁.

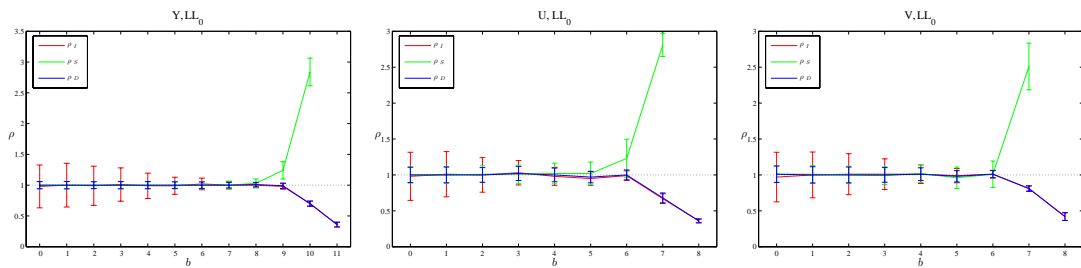


Figure 6.13: Accuracy of distortion estimation for subband LH₀.

2. “Crew”, CIF, 30 Hz, 96 frames

Employed decomposition commands:

```
L : T, 1;{
  L : S, 1;{
    L : T, 1;{
      L : T, 1;{
        L : S, 3;
        H : S, 3;}
      H : S, 3;}
    H : T, 2;}
  H : S, 4;}

```

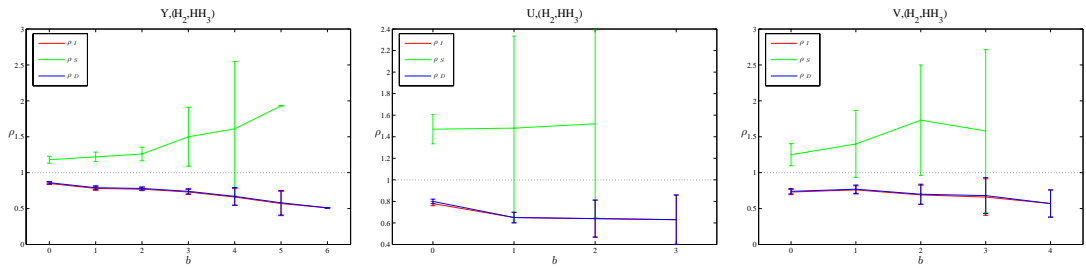


Figure 6.14: Accuracy of distortion estimation for subband (H_2, HH_3) .

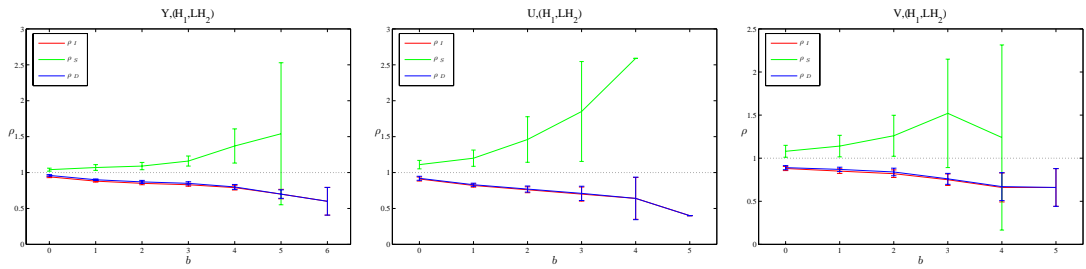


Figure 6.15: Accuracy of distortion estimation for subband (H_1, LH_2) .

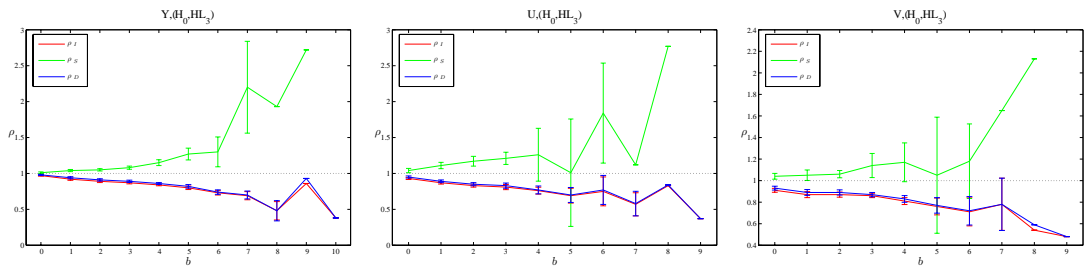


Figure 6.16: Accuracy of distortion estimation for subband (H_0, HL_3) .

B.2 Reconstructed Noise Energy Distribution

In this section the results of measuring the distribution of the reconstructed noise energy are presented. As described in Section 4.4, this procedure serves a purpose of determining the subband weighting factors, w_0 and w_1 , for the temporal low-pass and high-pass subbands, respectively. Noise signal of the unit energy is independently inserted into low-pass and the high-pass temporal subbands, prior to the inverse transform. Then the magnitude of energy is measured after the inverse transform is performed. The portion of the noise signal inserted into the low-pass temporal subband that reconstructs into the even frames is denoted with $G_{0,0}$, while the portion of the odd frames is $G_{0,1}$. Similarly, for the noise inserted into the high-pass temporal subband, the portions are $G_{1,0}$ and $G_{1,1}$, respectively. The subband weights, as defined in Section 4.4, then can be expressed with:

$$w_0 = \|g_0^{(j_0-1)}\| = G_{0,0} + G_{0,1}$$

$$w_1 = \|g_1^{(j_0-1)}\| = G_{1,0} + G_{1,1}.$$

These results also reveal the value of the peak cyclostationary reconstruction error ratio, since the error contributions to the even and odd frames can be expressed with:

$$\sigma_0^2 = G_{0,0} + G_{1,0}$$

$$\sigma_1^2 = G_{0,1} + G_{1,1},$$

so that $PCR(1) = \max(\sigma_0^2, \sigma_1^2) / \min(\sigma_0^2, \sigma_1^2)$. The results obtained for $PCR(1)$ are compared with the values given by the wavelet itself, which are presented in Table 6.5, along with the L_2 norms. However, since the perfect one-to-one

Table 6.5: $PCR(1)$, w_0 and w_1

	1/2	2/2	1/3	5/3	5/7	9/7
$PCR(1)$	5.0	1.0	4.5	1.917	1.003	1.221
w_0	1.0	1.0	0.866	0.866	1.011	0.991
w_1	1.414	1.0	1.414	1.199	0.990	1.020

pixel mapping is not preserved, there is a discrepancy between the measured and

the theoretical values. As mentioned in Section 4.4 the contributing factors are: interpolation used in the temporal prediction step, different connectivity state of the pixels and OBMC. In the results presented in the tables that follow, these factors have been selectively disabled or enabled, so that the influence of each can be observed separately.

Table 6.6: Haar with delta low-pass filter (1/2 wavelet), sequence “City”

	$G_{0,0}$	$G_{1,0}$	σ_0^2	$G_{0,1}$	$G_{1,1}$	σ_1^2	$PCR(1)$	w_0	w_1
integer precision MV, without OBMC									
Y	0.50	0.00	0.50	0.50	2.00	2.50	5.00	1.00	2.00
U	0.50	0.00	0.50	0.42	2.00	2.42	4.84	0.92	2.00
V	0.50	0.00	0.50	0.42	2.00	2.42	4.84	0.92	2.00
1/2-integer precision MV, without OBMC									
Y	0.50	0.00	0.50	0.41	2.00	2.41	4.82	0.91	2.00
U	0.50	0.00	0.50	0.42	2.00	2.42	4.84	0.92	2.00
V	0.50	0.00	0.50	0.42	2.00	2.42	4.84	0.92	2.00
integer precision MV, with OBMC									
Y	0.50	0.00	0.50	0.48	2.00	2.48	4.96	0.98	2.00
U	0.50	0.00	0.50	0.41	2.00	2.41	4.82	0.91	2.00
V	0.50	0.00	0.50	0.41	2.00	2.41	4.82	0.91	2.00
1/2-integer precision MV, with OBMC									
Y	0.50	0.00	0.50	0.40	2.00	2.40	4.80	0.90	2.00
U	0.50	0.00	0.50	0.41	2.00	2.41	4.82	0.91	2.00
V	0.50	0.00	0.50	0.41	2.00	2.41	4.82	0.91	2.00

Table 6.7: Haar (2/2 wavelet), sequence “City”

	$G_{0,0}$	$G_{1,0}$	σ_0^2	$G_{0,1}$	$G_{1,1}$	σ_1^2	$PCR(1)$	w_0	w_1
integer precision MV, without OBMC									
Y	0.50	0.49	0.99	0.50	0.53	1.03	1.05	1.00	1.02
U	0.50	0.49	0.99	0.42	1.06	1.48	1.50	0.92	1.55
V	0.50	0.49	0.99	0.42	1.06	1.48	1.50	0.92	1.55
1/2-integer precision MV, without OBMC									
Y	0.50	0.49	0.99	0.41	1.11	1.52	1.53	0.91	1.60
U	0.50	0.49	0.99	0.42	1.53	1.95	1.97	0.92	2.02
V	0.50	0.49	0.99	0.42	1.53	1.95	1.97	0.92	2.02
integer precision MV, with OBMC									
Y	0.50	0.49	0.99	0.48	0.56	1.04	1.05	0.98	1.05
U	0.50	0.49	0.99	0.41	1.07	1.48	1.50	0.91	1.56
V	0.50	0.49	0.99	0.41	1.07	1.48	1.50	0.91	1.56
1/2-integer precision MV, with OBMC									
Y	0.50	0.49	0.99	0.40	1.12	1.52	1.53	0.90	1.61
U	0.50	0.49	0.99	0.41	1.53	1.94	1.96	0.91	2.02
V	0.50	0.49	0.99	0.41	1.53	1.94	1.96	0.91	2.02

Table 6.8: Le Gall with delta low-pass filter (1/3 wavelet), sequence “City”

	$G_{0,0}$	$G_{1,0}$	σ_0^2	$G_{0,1}$	$G_{1,1}$	σ_1^2	$PCR(1)$	w_0	w_1
integer precision MV, without OBMC									
Y	0.50	0.00	0.50	0.34	2.00	2.34	4.68	0.84	2.00
U	0.50	0.00	0.50	0.29	2.00	2.29	4.57	0.79	2.00
V	0.50	0.00	0.50	0.29	2.00	2.29	4.57	0.79	2.00
1/2-integer precision MV, without OBMC									
Y	0.50	0.00	0.50	0.27	2.00	2.27	4.54	0.77	2.00
U	0.50	0.00	0.50	0.27	2.00	2.27	4.55	0.77	2.00
V	0.50	0.00	0.50	0.27	2.00	2.27	4.55	0.77	2.00
integer precision MV, with OBMC									
Y	0.50	0.00	0.50	0.33	2.00	2.33	4.65	0.83	2.00
U	0.50	0.00	0.50	0.28	2.00	2.28	4.56	0.78	2.00
V	0.50	0.00	0.50	0.28	2.00	2.28	4.55	0.78	2.00
1/2-integer precision MV, with OBMC									
Y	0.50	0.00	0.50	0.26	2.00	2.26	4.52	0.76	2.00
U	0.50	0.00	0.50	0.27	2.00	2.27	4.53	0.77	2.00
V	0.50	0.00	0.50	0.27	2.00	2.27	4.53	0.77	2.00

Table 6.9: Le Gall (5/3 wavelet), sequence “City”

	$G_{0,0}$	$G_{1,0}$	σ_0^2	$G_{0,1}$	$G_{1,1}$	σ_1^2	$PCR(1)$	w_0	w_1
integer precision MV, without OBMC									
Y	0.50	0.31	0.81	0.34	1.09	1.43	1.76	0.84	1.40
U	0.50	0.31	0.81	0.29	1.40	1.69	2.08	0.79	1.71
V	0.50	0.31	0.81	0.29	1.40	1.69	2.08	0.79	1.71
1/2-integer precision MV, without OBMC									
Y	0.50	0.31	0.81	0.27	1.43	1.70	2.10	0.77	1.74
U	0.50	0.31	0.81	0.27	1.62	1.89	2.34	0.77	1.93
V	0.50	0.31	0.81	0.27	1.62	1.89	2.34	0.77	1.93
integer precision MV, with OBMC									
Y	0.50	0.31	0.81	0.33	1.11	1.43	1.77	0.83	1.42
U	0.50	0.31	0.81	0.28	1.41	1.68	2.08	0.78	1.72
V	0.50	0.31	0.81	0.28	1.41	1.68	2.08	0.78	1.72
1/2-integer precision MV, with OBMC									
Y	0.50	0.31	0.81	0.26	1.44	1.70	2.10	0.76	1.75
U	0.50	0.31	0.81	0.27	1.62	1.89	2.33	0.77	1.93
V	0.50	0.31	0.81	0.27	1.62	1.89	2.33	0.77	1.93

References

- [1] A. Vetro, C. Christopoulos, and H. Sun, “Video transcoding architectures and techniques: An overview,” *IEEE Signal Processing Magazine*, vol. 20, pp. 18–29, Mar. 2003. [10](#)
- [2] S. Verdu, “Fifty years of Shannon theory,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 2057–2078, Oct. 1998. [11](#)
- [3] “Advanced video coding for generic audiovisual services,” Tech. Rep. Recommendation H.264 / 14496-10 AVC, ITU-T and ISO/IEC JTC1, 2003. [11](#)
- [4] V. K. Goyal, “Multiple description coding: Compression meets the network,” *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, Sept. 2001. [11](#)
- [5] A. Krishnamurthy, T. D. C. Little, and D. Castañón, “A pricing mechanism for scalable video delivery,” *ACM Multimedia Syst.*, vol. 4, no. 6, pp. 328–337, 1996. [11](#)
- [6] D. Taubman, “Remote browsing of JPEG2000 images,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, pp. 229–232, Sept. 2002. [11](#)
- [7] S. Deshpande and W. Zeng, “Scalable streaming of JPEG2000 images using hypertext transfer protocol,” in *ACM international conference on Multimedia*, pp. 372–381, 2001. [11](#)
- [8] J. Meessen, L.-Q. Xu, and B. Macq, “Content browsing and semantic context viewing through JPEG 2000-based scalable video summary,” *IEE Proc.-Vis. Image Signal Process.*, vol. 153, pp. 274–283, June 2006. [11](#)

-
- [9] D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 282–300, 2001. [12](#)
- [10] K. Ramchandran, A. Ortega, K. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 6–23, Jan. 1993. [12](#)
- [11] A. Ortega and K. Ramchandran, "Rate-distortion techniques in image and video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, Nov. 1998. [12](#), [45](#)
- [12] P. G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Comm.*, vol. 46, pp. 1555–1559, Dec. 1998. [13](#)
- [13] A. A. Alatan, M. Zhao, and A. N. Akansu, "Unequal error protection of SPIHT encoded image bit streams," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 814–818, June 2000. [13](#)
- [14] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bit streams," *IEEE Signal Processing Magazine*, vol. 22, pp. 91–107, Nov. 2005. [13](#)
- [15] I. V. Bajić and J. W. Woods, "EZBC video streaming with channel coding and error concealment," in *Proc. SPIE Visual Communications and Image Processing*, vol. 5150, pp. 512–522, July 2003. [13](#)
- [16] Y. Shan, S. Kalyanaraman, J. W. Woods, and I. V. Bajić, "Joint source - network error control coding for scalable overlay video streaming," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2005. [13](#)
- [17] N. Šprljan, M. Mrak, and E. Izquierdo, "A fast error protection scheme for transmission of embedded coded images over unreliable channels and fixed packet size," in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, (Philadelphia, USA), Mar. 2005. [13](#)

REFERENCES

- [18] I. Kompatsiaris, Y. Avrithis, P. Hobson, and M. G. Strintzis, “Integrating knowledge, semantics and content for user-centred intelligent media services: The aceMedia project,” in *Proc. Int. Work. on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Apr. 2004. [13](#)
- [19] D. S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, “Complexity scalable motion compensated wavelet video encoding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 982–993, Aug. 2005. [15](#)
- [20] H. M. Radha, M. van der Schaar, and Y. Chen, “The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP,” *IEEE Trans. Multimedia*, vol. 3, pp. 53–68, Mar. 2001. [15](#)
- [21] N. Šprljan, D. Djordjevic, and E. Izquierdo, “Scalability evaluation of still image coders,” in *Proc. Int. Work. on Image Analysis for Multimedia Interactive Services (WIAMIS)*, (Lisabon, Portugal), p. 88, Apr. 2004. [17](#)
- [22] D. Taubman and M. Marcellin, *JPEG2000 Image Compression: Fundamentals, Standards and Practice*. Boston, MA, USA: Kluwer Academic Publishers, 2002. [17](#), [23](#), [31](#), [42](#), [95](#)
- [23] H. Danyali and A. Mertins, “Highly scalable image compression based on SPIHT for network applications,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, pp. 217–220, Sept. 2002. [17](#)
- [24] S.-T. Hsiang, “Embedded image coding using zeroblocks of sub-band/wavelet coefficients and context modeling,” in *Data Compression Conference (DCC)*, (Snowbird, USA), pp. 83–92, 2001. [17](#), [83](#), [84](#)
- [25] M. D. Adams, “The JPEG-2000 still image compression standard,” Tech. Rep. N2412, ISO/IEC JTC 1/SC 29/WG1, Dec. 2002. [17](#)
- [26] M. Gormish, “JPEG 2000: worth the wait?,” in *42nd Midwest Symposium on Circuits and Systems*, vol. 2, pp. 766–769, 1999. [17](#)
- [27] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, July 2000. [17](#), [83](#), [86](#)

REFERENCES

- [28] W. Li, J.-R. Ohm, M. van der Schaar, H. Jiang, and S. Li, “MPEG-4 video verification model version 19.0,” Tech. Rep. M10431, ISO/IEC JTC1/SC29 WG11, Hawaii, USA, Dec. 2003. [17](#), [59](#)
- [29] J. Reichel, H. Schwarz, and M. Wien, “Joint scalable video model JSVM-5,” Tech. Rep. N7796, ISO/IEC JTC1/SC29 WG11, Bangkok, Thailand, Jan. 2006. [18](#), [57](#)
- [30] N. Mehrseresht and D. Taubman, “A flexible structure for fully scalable motion compensated 3D-DWT with emphasis on the impact of spatial scalability,” *IEEE Trans. Image Processing*, vol. 15, pp. 740–753, 2006. [19](#), [47](#), [55](#)
- [31] P. Chen and J. W. Woods, “Bidirectional MC-EZBC with lifting implementation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 1183–1194, Oct. 2004. [19](#)
- [32] R. Xiong, X. Ji, D. Zhang, and J. Xu, “Vidwav wavelet video coding specifications,” Tech. Rep. M12339, ISO/IEC JTC1/SC29 WG11, Poznan, Poland, July 2005. [19](#), [46](#), [61](#), [93](#)
- [33] M. Mrak, N. Šprljan, T. Žgaljić, N. Ramzan, S. Wan, and E. Izquierdo, “Performance evidence of software proposal for wavelet video coding exploration group,” Tech. Rep. M13146, 76th MPEG Meeting, ISO/IEC JTC1/SC29/WG11/MPEG2005, Montreux, Switzerland, Apr. 2006. [19](#)
- [34] K. Ramchandran, M. Vetterli, and C. Herley, “Wavelets, subband coding and best bases,” *Proc. of IEEE*, vol. 86, pp. 541–560, Apr. 1996. [23](#)
- [35] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1998. [23](#)
- [36] T. Sarkar, C. Su, R. Adve, M. Salazar-Palma, L. Garcia-Castillo, and R. Boix, “A tutorial on wavelets from an electrical engineering perspective, part 1: Discrete wavelet techniques,” *IEEE Antennas and Propagation Magazine*, vol. 40, pp. 49–68, Oct. 1998. [23](#)

REFERENCES

- [37] T. Sarkar and C. Su, “A tutorial on wavelets from an electrical engineering perspective, part 2: The continuous case,” *IEEE Antennas and Propagation Magazine*, vol. 40, pp. 36–49, Dec. 1998. [23](#)
- [38] M. Unser and T. Blu, “Wavelet theory demystified,” *IEEE Trans. Signal Processing*, vol. 51, pp. 470–483, Feb. 2003. [23](#)
- [39] B. Usevitch, “Optimal bit allocation for biorthogonal wavelet coding,” in *Data Compression Conference (DCC)*, pp. 387–395, Mar. 1996. [29](#)
- [40] F. M. de Saint-Martin, P. Siohan, and A. Cohen, “Biorthogonal filterbanks and energy preservation property in image compression,” *IEEE Trans. Image Processing*, vol. 8, pp. 168–178, Feb. 1999. [35](#), [36](#)
- [41] M. Unser, “Approximation power of biorthogonal wavelet expansions,” *IEEE Trans. Signal Processing*, vol. 44, pp. 519–527, Mar. 1996. [37](#)
- [42] A. Signoroni and R. Leonardi, “Modeling and reduction of PSNR fluctuations in 3D wavelet coding,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 3, (Thessaloniki), pp. 812–815, 2001. [38](#)
- [43] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps,” *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, 1998. [39](#)
- [44] W. Sweldens, “The lifting scheme: A construction of second generation wavelets,” *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998. [39](#)
- [45] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk, “Nonlinear wavelet transforms for image coding,” in *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 662–667, 1997. [41](#), [95](#)
- [46] V. Silva and L. de Sá, “General method for perfect reconstruction subband processing of finite length signals using linear extensions,” *IEEE Trans. Signal Processing*, vol. 47, pp. 2572–2575, Sept. 1999. [42](#), [137](#)

REFERENCES

- [47] G. Karlsson and M. Vetterli, “Three dimensional subband coding of video,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, (New York, NY), pp. 1100–1103, 1988. [45](#)
- [48] D. Taubman and A. Zakhor, “Multirate 3D subband coding of video,” *IEEE Trans. Image Processing*, vol. 3, pp. 572–589, Sept. 1994. [45](#)
- [49] J.-R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Trans. Image Processing*, vol. 3, pp. 559–571, Sept. 1994. [45](#)
- [50] N. Mehrseresht and D. Taubman, “Spatial scalability and compression efficiency within a flexible motion compensated 3D-DWT,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2005. [47](#)
- [51] C. Ong, S. Shen, M. Lee, and Y. Honda, “Wavelet video coding - generalized spatial temporal scalability (GSTS),” Tech. Rep. M11952, ISO/IEC JTC1/SC29 WG11, Busan, Korea, Apr. 2005. [47](#), [61](#)
- [52] S.-T. Hsiang and J. W. Woods, “Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank,” *Signal Processing: Image Communications*, vol. 16, pp. 705–724, May 2001. [48](#)
- [53] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, “Improved update operators for lifting-based motion-compensated temporal filtering,” *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 146–149, 2005. [48](#)
- [54] B. Girod and S. Han, “Optimum update for motion-compensated lifting,” *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 150–153, 2005. [48](#)
- [55] M. van der Schaar and D. S. Turaga, “Unconstrained motion compensated temporal filtering (UMCTF) framework for wavelet video coding,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 3, pp. 81–84, 2003. [49](#)
- [56] Y. Wu and J. Woods, “Directional spatial I-blocks for the MC-EZBC video coder,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 3, pp. 129–132, May 2004. [49](#), [98](#)

-
- [57] N. Bozinovic, J. Konrad, W. Zhao, and C. Vazquez, “On the importance of motion invertibility in MCTF/DWT video coding,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 2, pp. 49–52, 2005. [51](#)
- [58] S.-J. Choi and J. Woods, “Motion-compensated 3-D subband coding of video,” *IEEE Trans. Image Processing*, vol. 8, pp. 155–167, Feb. 1999. [51](#)
- [59] S.-T. Hsiang, J. W. Woods, and J.-R. Ohm, “Invertible temporal subband/wavelet filter banks with half-pixel-accurate motion compensation,” *IEEE Trans. Image Processing*, vol. 13, pp. 1018–1028, Aug. 2004. [52](#)
- [60] A. Secker and D. Taubman, “Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression,” *IEEE Trans. Image Processing*, vol. 12, pp. 1530–1542, Dec. 2003. [52](#), [97](#)
- [61] R. Xiong, J. Xu, F. Wu, S. Li, and Y.-Q. Zhang, “Spatial scalability in 3D wavelet coding with spatial domain MCTF encoder,” in *Proc. Picture Coding Symp. (PCS)*, (San Francisco, USA), Dec. 2004. [54](#)
- [62] N. Božinović and J. Konrad, “Modeling motion for spatial scalability,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 2, pp. 29–32, 2006. [54](#)
- [63] N. Adami, M. Brescianini, M. Dalai, R. Leonardi, and A. Signoroni, “A fully scalable video coder with inter-scale wavelet prediction and morphological coding,” in *Proc. SPIE Visual Communications and Image Processing*, vol. 5960, (Beijing, China), July 2005. [56](#)
- [64] M. Wien, “Variable block-size transforms for H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 604–613, July 2003. [57](#)
- [65] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis, “Fully-scalable wavelet video coding using in-band motion compensated temporal filtering,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 3, pp. 417–420, 2003. [57](#)

-
- [66] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis, “Complete-to-overcomplete discrete wavelet transforms for scalable video coding with MCTF,” in *Proc. SPIE Visual Communications and Image Processing*, vol. 5150, pp. 719–731, July 2003. 58
- [67] R. Xiong, J. Xu, F. Wu, and S. Li, “In-scale motion aligned temporal filtering,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2006. 58
- [68] D. Mukherjee, A. Said, and S. Liu, “A framework for fully format-independent adaptation of scalable bit streams,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 1280–1290, Oct. 2005. 62
- [69] I. Burnett, R. V. de Walle, K. Hill, J. Bormans, and F. Pereira, “MPEG-21: goals and achievements,” *IEEE Multimedia*, vol. 10, no. 4, pp. 60–70, 2003. 62
- [70] A. Vetro and C. Timmerer, “Digital item adaptation: Overview of standardization and research activities,” *IEEE Trans. Multimedia*, vol. 7, pp. 418–426, June 2005. 62
- [71] T. Žgaljić, N. Šprljan, and E. Izquierdo, “Scalable video adaptation based on bitstream syntax description,” in *Proc. Workshop on Immersive Communication and Broadcast Systems (ICOB)*, (Berlin, Germany), Oct. 2005. 62, 65
- [72] J. Xu and R. Leonardi, “Exploration experiments in wavelet video coding,” Tech. Rep. N7333, ISO/IEC JTC1/SC29 WG11, Poznan, Poland, July 2005. 75
- [73] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Space-frequency quantization for wavelet image coding,” *IEEE Trans. Image Processing*, vol. 6, pp. 677–693, May 1997. 83
- [74] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, “Joint space-frequency segmentation using balanced wavelet packet trees for least-cost

- image representation,” *IEEE Trans. Image Processing*, vol. 6, pp. 1213–1230, Sept. 1997. [83](#), [95](#)
- [75] K. K. Lin and R. M. Gray, “Rate-distortion optimization for the SPIHT encoder,” in *Data Compression Conference (DCC)*, pp. 123–132, 2001. [83](#)
- [76] N. Farvardin and J. W. Modestino, “Optimum quantizer performance for a class of non-gaussian memoryless sources,” *IEEE Trans. Inf. Theory*, vol. 30, pp. 485–497, May 1984. [83](#)
- [77] F. G. Meyer, A. Z. Averbuch, and J.-O. Strömberg, “Fast adaptive wavelet packet image compression,” *IEEE Trans. Image Processing*, vol. 9, pp. 792–800, May 2000. [83](#), [95](#)
- [78] J. Li and S. Lei, “Rate-distortion optimized embedding,” in *Proc. Picture Coding Symp. (PCS)*, (Berlin, Germany), pp. 201–206, Sept. 1997. [83](#)
- [79] T. Strutz and E. Mueller, “Image data compression with pdf-adaptive reconstruction of wavelet coefficients,” in *Proceedings of SPIE*, vol. 2569, (San Diego, USA), pp. 747–758, July 1995. [84](#)
- [80] A. Said and W. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996. [86](#), [130](#)
- [81] A. Munteanu, Y. Andreopoulos, M. van der Schaar, P. Schelkens, and J. Cornelis, “Control of the distortion variation in video coding systems based on motion compensated temporal filtering,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 3, pp. II–61–64, Sept. 2003. [89](#)
- [82] K. Hanke, J.-R. Ohm, and T. Rusert, “Adaptation of filters and quantization in spatio-temporal wavelet coding with motion compensation,” in *Proc. Picture Coding Symp. (PCS)*, (St. Malo, France), pp. 49–54, Apr. 2003. [89](#), [101](#)
- [83] A. Golwelkar, *Motion Compensated Temporal Filtering and Motion Vector Coding Using Longer Filters*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, Sept. 2004. [89](#)

REFERENCES

- [84] H. Watanabe and S. Singhal, “Windowed motion compensation,” in *Proc. SPIE Visual Comm. Image Proc.*, pp. 582–589, Nov. 1991. [90](#), [97](#)
- [85] N. Mehrseresht and D. Taubman, “Adaptively weighted update steps in motion compensated lifting based scalable video compression,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 2, pp. 771–774, 2003. [90](#)
- [86] R. Leonardi, T. Oelbaum, and J.-R. Ohm, “Status report on wavelet video coding exploration,” Tech. Rep. N8043, ISO/IEC JTC1/SC29/WG11, Montreux, Switzerland, Apr. 2006. [93](#)
- [87] M. Unser and T. Blu, “Mathematical properties of the JPEG2000 wavelet filters,” *IEEE Trans. Image Processing*, vol. 12, pp. 1080–1090, Sept. 2003. [95](#)
- [88] F. G. Meyer, “Image compression with adaptive local cosines: A comparative study,” *IEEE Trans. Image Processing*, vol. 11, pp. 616–629, June 2002. [95](#)
- [89] N. M. Rajpoot, R. G. Wilson, F. G. Meyer, and R. R. Coifman, “Adaptive wavelet packet basis selection for zerotree image coding,” *IEEE Trans. Image Processing*, vol. 12, pp. 1460–1472, Dec. 2003. [95](#)
- [90] N. Šprljan, S. Grgic, and M. Grgic, “Modified SPIHT algorithm for wavelet packet image coding,” *Real-Time Imaging*, vol. 11, pp. 378–388, Oct.-Dec. 2005. [95](#)
- [91] G. Piella and H. J. A. M. Heijmans, “Adaptive lifting schemes with perfect reconstruction,” *IEEE Trans. Signal Processing*, vol. 50, pp. 1620–1630, July 2002. [95](#)
- [92] M. Vetterli, “Wavelets, approximation, and compression,” *IEEE Signal Processing Magazine*, vol. 18, pp. 59–73, Sept. 2001. [96](#)
- [93] V. Chappelier, *Codage progressif d’images par ondelettes orientées*. PhD thesis, Université de Rennes, 2005. [96](#)

-
- [94] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, “Wavelet-domain approximation and compression of piecewise smooth images,” *IEEE Trans. Image Processing*, 2005. [96](#)
- [95] E. Candés and D. Donoho, “Curvelets – a surprisingly effective nonadaptive representation for objects with edges,” tech. rep., Saint-Malo: Vanderbilt University Press, 1999. [96](#)
- [96] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, “Image compression using an efficient edge cartoon + texture model,” in *Data Compression Conference (DCC)*, (Snowbird, Utah, USA), Apr. 2002. [96](#)
- [97] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, “Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images,” *IEEE Trans. Image Processing*, vol. 14, pp. 343–359, Mar. 2005. [96](#), [97](#)
- [98] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, vol. 41, pp. 3397–3415, Dec. 1993. [96](#)
- [99] P. Dragotti and M. Vetterli, “Wavelet footprints: Theory, algorithms and applications,” *IEEE Trans. Signal Processing*, vol. 51, pp. 1306–1323, May 2003. [96](#)
- [100] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *International Conference on Computer Vision*, (Bombay), pp. 839–846, 1998. [97](#), [119](#)
- [101] T. F. Chan and H. M. Zhou, “Adaptive ENO-wavelet transforms for discontinuous functions,” CAM Report 22, 21, Dept. of Math, UCLA, June 1999. [98](#)
- [102] W. Ding, F. Wu, and S. Li, “Lifting-based wavelet transform with directionally spatial prediction,” in *Proc. Picture Coding Symp. (PCS)*, (San Francisco, USA), Dec. 2004. [98](#)

REFERENCES

- [103] V. Chappelier and C. Guillemot, “Oriented wavelet transform on a quincunx pyramid for image compression,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Sept. 2005. [98](#)
- [104] G. Uytterhoeven and A. Bultheel, “The red-black wavelet transform,” Tech. Rep. 271, Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan 200A B-3001 Heverlee (Belgium), Dec. 1997. [98](#)
- [105] W. Sweldens and P. Schröder, “Building your own wavelets at home,” in *Wavelets in Computer Graphics*, pp. 15–87, ACM SIGGRAPH Course notes, 1996. [106](#)
- [106] J. Kovacevic and W. Sweldens, “Wavelet families of increasing order in arbitrary dimensions,” *IEEE Trans. Image Processing*, vol. 9, pp. 480–496, Mar. 2000. [107](#)
- [107] M. Beermann and M. Wien, “Joint reduction of ringing and blocking,” Tech. Rep. M12640, ISO/IEC JTC1/SC29 WG11, Nice, France, Oct. 2005. [119](#)
- [108] M. Mrak, N. Šprljan, and E. Izquierdo, “Motion estimation in temporal subbands for quality scalable motion coding,” *Electronics Letters*, vol. 41, pp. 1050–1051, Sept. 2005. [121](#)
- [109] M. Mrak, D. Marpe, and T. Wiegand, “A context modeling algorithm and its application in video compression,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, (Barcelona, Spain), Sept. 2003. [130](#)
- [110] P. Chen, *Fully Scalable Subband/Wavelet Coding*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, May 2003. [131](#)
- [111] S. Rout and A. Bell, “Narrowing the performance gap between orthogonal and biorthogonal wavelets,” in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, 2004. [137](#)
- [112] G. Pau, A. Pesquet-Popescu, and G. Piella, “Modified M-band synthesis filter bank for fractional scalability of images,” *IEEE Signal Processing Letters*, vol. 13, pp. 345–348, June 2006. [137](#)

-
- [113] G. Evangelista and S. Cavaliere, “Frequency-warped filter banks and wavelet transforms: a discrete-time approach via Laguerre expansion,” *IEEE Trans. Signal Processing*, vol. 46, pp. 2638–2650, Oct. 1998. 137
- [114] C. Tillier and B. Pesquet-Popescu, “3D, 3-band, 3-tap temporal lifting for scalable video coding,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 2, pp. 779–82, Sept. 2003. 137
- [115] S. Tsai and H. M. Hang, “Motion information scalability for MC-EZBC,” *elsp*, vol. 19, pp. 675–684, Aug. 2004. 138
- [116] A. Secker and D. Taubman, “Highly scalable video compression with scalable motion coding,” *IEEE Trans. Image Processing*, vol. 13, pp. 1029–1041, Aug. 2004. 138
- [117] G. Boisson, E. François, and C. Guillemot, “Accuracy-scalable motion coding for efficient scalable video compression,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 2, pp. 1309–1312, Oct. 2004. 138
- [118] R. Xiong, J. Xu, F. Wu, S. Li, and Y.-Q. Zhang, “Layered motion estimation and coding for fully scalable 3D wavelet video coding,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, (Singapore), Oct. 2004. 138
- [119] M. Mrak, G. Abhayaratne, and E. Izquierdo, “On the influence of motion vector precision limiting in scalable video coding,” in *Proc. Int. Conf. Signal Processing (ICSP)*, pp. 1143–1146, Aug. 2004. 138
- [120] T. Žgaljić, M. Mrak, N. Šprljjan, and E. Izquierdo, “An entropy coding scheme for multi-component scalable motion information,” in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc. (ICASSP)*, 2006. 138
- [121] N. Šprljjan, “Matlab image & video compression depot - <http://www.sprljjan.com/nikola/matlab/>,” 2005. 138