Distributed Video Coding (DVC):

Motion estimation and DCT quantization in low complexity video compression

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op Woensdag 30 juni 2010 om 10.00 uur
door Stefan BORCHERT
Diplom Ingenieur in Information Technology, University of Rostock, Germany,
geboren te Rostock, Germany.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. R. L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. R. L. Lagendijk	Technische Universiteit Delft, promotor
Prof. dr.-ing. J. Östermann	Leibniz Universität Hannover
Prof. dr. ir. P. H. N. de With	Technische Universiteit Eindhoven
Prof. dr. K.G. Langendoen	Technische Universiteit Delft
Prof. dr. ir. I. Niemegeers	Technische Universiteit Delft
Prof. dr. M. C. Gastpar	University of California, Berkeley, United States
Ing. R. Klein Gunnewiek	Philips Research Laboratories, Eindhoven

Advanced School for Computing and Imaging

This work was carried out in the ASCI graduate school. ASCI dissertation series number 196.

Published by TU Delft Mediamatica.

ISBN: 978-94-90818-01-2

Distributed Video Coding (DVC):

Motion estimation and DCT quantization in low complexity video compression

Preface

The research for this thesis was conducted within the I-SHARE project. One of the main objectives of the I-SHARE project was to develop video compression techniques on resource constrained devices making use of distributed compression technologies. I-SHARE was funded by the Ministry of Economic Affairs, as a sub project of the FREEBAND project. FREEBAND Communication is supported by the Ministry of Economic Affairs through the BSIK program (BSIK 03025).

Freeband Communication co-operates with numerous international projects and institutes, both existing programs, as well as in the 6th framework program of the European Union. In the I-SHARE project itself, the following parties were involved: Delft University of Technology, Eindhoven University of Technology, University of Twente, Vrije Universiteit Amsterdam and Philips Research.

The research in this thesis was carried out during the period February 2006 - February 2010 in the Information and Communication Theory group at Delft University of Technology, Delft, The Netherlands. During this time the research was mainly conducted in the Digital Signal Processing and later the Video Processing Systems group at Philips Research, Eindhoven, The Netherlands.

S. Borchert, Eindhoven, February 2010.

Acknowledgments

First I would like to thank everyone who, in one way or another, helped me to complete this thesis. I know that many people have supported me during the Ph.D. In the following I would like to thank a few of them in particular.

I would like to thank my promotor Inald Lagendijk and my supervisor at Philips Rene Klein Gunnewiek, without whose support, encouragement and outstanding supervision this thesis would not have been possible. I also owe a lot to Chris Varekamp, who was my first supervisor at Philips and encouraged me to consider the PhD position with Inald and Rene. Inald contributed an enormous amount of expertise and experience and Rene a valuable viewpoint on practical aspects.

Even though I was only in Delft once a week, I met a lot of great people there. I would like to thank my fellow PhD students in the ICT group for making that one day a week memorable. Just to name a few (there are many more): Alper, Bartek, Cees, Hassan, Katerina, Jacco, Maarten, Stevan, Umut and Zeki. I am especially grateful to my research partner Ronald Westerlaken, with whom I had the honor of working together in the I-SHARE project. In addition I would like to acknowledge the support given by the following members of the ICT group: Anja, Ben, Robbert and Saskia.

Thanks go out to all my fellow (PhD) students at Philips Research for a great work atmosphere. Just to name a few: Alberto, Nico, Michael, Nico, Othmar, Tom and Tommy. For the sportive relief I would like to especially thank Olaf and Ruben. And of course the fellow Germans also deserve notice with Janto and Tobias. My gratitude also goes to all Philips employees, who provided professional support. I will also greatly miss the countless social events like poker nights, sailing trips.

Last but by no means least I would like to thank my family and friends in Germany, who always made me feel at home in Rostock. Your support made this thesis possible. And thanks to all of you every vacation was a special experience which helped to recharge the batteries.

Contents

Nomenclature

3DRS	3-D Recursive Search
3SS	Three-Step Search
AVC	Advanced Video Coding
AWGN	Additive White Gaussian Noise
B-DM	Bit plane-based Dependency Model
B-SWD	Bit plane-based Slepian-Wolf Decoder
BCH	Bose-Chaudhuri-Hocquenghem
BER	Bit Error Rate
BSC	Binary Symmetric Channel
CARS	Content-Adaptive Recursive Search
CBP	Constrained Baseline Profile
CIF	Common Intermediate Format
DCT	Discrete Cosine Transform
DS	Diamond Search
DSC	Distributed Source Coding
DVC	Distributed Video Coding
EM	Expectation-Maximization
EOB	End-of-Block
EPZS	Enhanced Predictive Zonal Search
FEC	Forward Error Correction
GOP	Group of Pictures

HDTV	High Definition Television
i.i.d.	independently and identically distributed
IDCT	Inverse Discrete Cosine Transform
JPEG	Joint Photographic Experts Group
LDPC	Low Density Parity Check
MAD	Mean Absolute Difference
MAP	Maximum *a posteriori*
MC	Motion Compensation
ME	Motion Estimation
MI	Motion(-compensated) Interpolation
MI-X	Motion(-compensated) Interpolation with a GOP-size of X
ML	Motion Learning
MO	Motion Oracle
MPEG-2	Moving Picture Experts Group 2
MSB	Most Significant Bit Plane
MSE	Mean Squared Error
MX	Motion(-compensated) Extrapolation
PDF	Probability Density Function
PMF	Probability Mass Function
PRISM	Power-efficient, Robust, High-compression, Syndrome-based Multimedia coding
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
RD	Rate Distortion

S-DM	Symbol-based Dependency Model
S-SWD	Symbol-based Slepian-Wolf Decoder
SAD	Sum of Absolute Differences
SEASON	Source Encoding with side-information under Ambiguous State of Nature
SISO	Soft-Input Soft-Output
SW	Slepian-Wolf
VDC	Virtual Dependency Channel
VLC	Variable Length Code
WZ	Wyner-Ziv

1. Introduction

1.1. Video compression

Since last decade we are witnessing a transformation in the way we communicate. Modern communication media allow for intense long-distance exchanges between large numbers of people. In this environment digital media have become an integral part of our lifestyle. At the same time, the use of digital media has become mobile, this follows the general trend of *ubiquitous computing*, where information processing is thoroughly integrated into everyday devices and activities. Key to this abundant, mobile media experience are modern compression algorithms, especially connectivity and video compression.

It is evident that visual information is of vital importance if people are to perceive, recognize and understand the surrounding world. However, video involves a huge amount of data. The purpose of video compression is to create a compact representation of video data. As compressed video data requires less storage space and smaller transmission bandwidth, video compression is an integral part of most video capture, storage, processing, communication, and display systems. Especially bandwidth is often a limiting factor for many applications. An example that also touches the boundaries of storage capacity nowadays is uncompressed High Definition Television (HDTV). To store an uncompressed 2 hours HTDV movie would require 80 Blu-Ray discs. In practice only one such Blu-Ray disc is needed to store a high quality HDTV movie. This simple example shows the importance of video compression as enabling technology.

But compression comes at a cost. First, an increase in compression decreases the visual quality. Second, the complexity of the algorithms increases with better compression. Furthermore, the best compression is also dependent on device, location and application. Still, a user should not be required to deal with complex configurations and choices regarding video and compression format. That is why there are video compression standards, able to deal with a multitude of different application scenarios. These standards, also referred to as video coding standards, are widely used and evolving continuously.

1.2. State of the art conventional video coding standards

Throughout the development of video coding standards a rule of thumb emerged. It indicates that a new video coding standard should yield a significant decrease in bit rate to be worthwhile. For instance, the latest video coding standard H.264 Advanced Video Codec (AVC) offers a bit rate saving of around 50% with respect to the previous Motion Picture Experts Group Video Coding 2 (MPEG-2) standard [102, 101]. However, these lower bit rates required to obtain the same quality are not without cost. The complexity of both encoder and decoder increase, usually by a factor larger than two. Encoder and decoder complexity are also not identical. The encoder is generally one to two orders of magnitude more complex than the decoder. A lightweight decoder is important when focusing on the main application these video codecs are designed for, the broadcasting case. In that case the movie is encoded once and then decoded by millions of users. Naturally, the main focus is on keeping the decoder complexity as low as possible.

While the main focus for the average user was solely video consumption in the past we observe a shift towards also producing (and sharing) video. That is not done with high end professional cameras but constrained media devices. For these devices the complexity is an important limitation. For that reason the codecs also include profiles to deal with these new requirements. For H.264 the Constrained Baseline Profile (CBP) has the lowest complexity. CBP is used primarily for low-cost applications. It is used widely in videoconferencing and mobile applications [107]. Meeting the real time constraint however is still difficult and requires heavy optimizations on the system, the algorithm and the instruction level [96]. Various algorithmic and implementation techniques are necessary to optimize such an H.264 video encoder [78].

The by far most complex part of the H.264 encoder is the Motion Estimation (ME), which occupies up to 70% of total encoding complexity [55]. To reduce this significant encoding complexity many techniques for motion estimation have been proposed in literature to replace the exhaustive full search. More efficient search methods include for instance the popular Three-Step Search (3SS) and the Diamond Search (DS) [55]. Another option is to ignore the motion and apply image coding like for instance JPEG [81]. To ignore the motion however incurrs a major loss in compression performance.

1.3. Low complexity video encoding with distributed video coding

This thesis deals with an alternative method of ignoring the motion at the encoder - potentially without losing any compression performance. In Distributed Video Coding (DVC) the motion estimation and its complexity is shifted from the encoder to the decoder. The focus is on a lightweight encoder, suited for a constrained device. In contrast, the decoder is assumed to be resource abundant. An example would be to capture video on a constrained device and later decode on the home computer without any time constraints. Another application example employs a transcoder. Transcoding is the direct digital-to-digital conversion of one encoding format to another. Such a transcoder approach would offer both lightweight encoding and lightweight decoding for the user. The computationally complex operations, i.e. DVC decoding and H.264 encoding, would be tackled by a powerful server in the network.

It is not the purpose of DVC to replace H.264 by providing better compression. In fact, the best DVC can be expected to do is to perform comparably. In conventional video coding, both the encoder and the decoder have access to the predicted frame. It is available at the encoder since both, motion estimation and Motion Compensation (MC), are executed there. The decoder needs to execute only the compensation step since it receives motion information from the encoder. Next to this motion information, the motion vectors, the decoder also receives the residual difference between the original frame and its prediction. With this information the decoder is able to reconstruct the original frame.

In DVC the information is distributed and only the decoder has access to the predicted frame. Consequently, the encoder can only send information about the original frame itself. Both encoding and decoding have to be redesigned to achieve a comparable compression efficiency to conventional video coding while maintaining a low encoder complexity.

1.4. Outline

In this thesis we focus on the inherent performance limitations of DVC and focus on three challenges. These challenges will be introduced in Chapter 2. They will then be analyzed separately in Chapter 3 (channel coding), Chapter 4 (motion estimation) and Chapter 5 (quantization). Since the challenges are not orthogonal it is also necessary to analyze possible interaction between them. Chapter 6 analyzes possible interactions between the channel coding and the motion estimation. Chapter 7 then compares how our latest DVC system compares against conventional video codecs in terms of compression efficiency and encoder complexity. Finally, Chapter 8 finishes the thesis with a discussion and an outlook into the future of DVC.

Chapter 2 focuses on the challenges in distributed video coding. First, we introduce the underlying theory. This will help to understand the challenges better. We then look at practical approaches from literature. After introducing the general components of such a DVC system we show challenges compared to state of art conventional video coding. Finally we give an overview of how the challenges are addressed in literature.

Chapter 3 focuses on our approach to one of the challenges, the channel coding, which replaces the source coding from conventional video coding. First we compare the two most widely used channel coders, turbo and Low Density Parity Check (LDPC) codes. After motivating our choice we continue with looking at LDPC exclusively. Since these codes rely heavily on an accurate channel model for the Virtual Dependency Channel (VDC) we then investigate which model is best suited for video data. The data itself can be encoded in two ways. First the symbols themselves can be encoded and secondly it is possible to split the symbol values into bit planes and to encode these. In the final part of this chapter we compare the two methods. To conclude we ascertain the channel coding choices we will adhere to for the rest of the thesis.

Chapter 4 focuses on our approach to another challenge, the motion estimation at the decoder. First we establish the differences between conventional ME at the encoder and ME at the decoder. These include availability of reference data, motion vectors processing and cost to send motion vectors. In this context, we also address the limitations of block-based motion estimation. For motion compensation at the decoder we look at two approaches, motion compensated inter- and extrapolation. We present our proposed extrapolation scheme, using three frames. Finally, we evaluate the prediction quality and the system Rate Distortion (RD) performance of inter- and extrapolation.

Chapter 5 shifts the focus from Mean Squared Error (MSE)/Peak Signal to Noise Ratio (PSNR) to the system RD performance. As such it is necessary to take spatial correlation into account. A spatially decorrelating transform like the Discrete Cosine Transform (DCT) increases the RD performance significantly. At the same time there is a new challenge with respect to DVC, namely how to quantize the DCT coefficients. We investigate three different quantization methods. Further, we propose a method to improve the motion estimation by using decoded coefficients to improve the remaining ones. The chapters up to this one focused on a single challenge or component separately. However, it is beneficial to not look at them as stand alone processes.

Chapter 6 then focuses on how to improve the VDC modeling by using motion information. The first observation made is that the VDC is non-stationary. We show that it is highly beneficial to take this non-stationarity into account and use distinct VDC models. We investigate a classification oracle and its sensitivity towards misclassification. We then focus on how to acquire helpful information from the motion estimation to make a reliable classification. We find it very difficult to achieve reliable classification.

Chapter 7 focuses on a comparison of the proposed DVC components and a state of the art video codec. We analyze how the methods derived in Chapter 3, 4 and 5 compare to their counterparts in H.263. We discuss the reason for observed RD performance differences. Finally, we look at the trade-off between low complexity encoding and RD performance for both conventional video coding and DVC. This benchmark is not limited to only our DVC codec but includes the DISCOVER codec [1] from literature.

Chapter 8 concludes the thesis with a discussion on the findings made with respect to DVC. We summarize and evaluate these findings. Furthermore we provide final considerations and an outlook into the near future with respect to DVC.

1.5. Contributions

The work presented in this thesis offers insights into different aspects of DVC.
With regard to channel coding we analyze which state of the art channel codes and models are most suited to DVC. Furthermore we propose a sophisticated bit plane-based coding scheme. This scheme is able to achieve a performance similar to the more complex symbol-based coding. Our contributions have been published in [63, 98, 97]

With regard to the motion estimation we show the utility of true motion estimation in combination with motion compensated extrapolation. We observe superiority over the still widely used motion compensated interpolation. For that reason we focus on improving an extrapolative approach. Our contributions have been published in [27, 24, 25]

With regard to the quantization of transform coefficients we show it is a more difficult problem to solve in DVC than in conventional coding. Furthermore it is a problem that seems to be under-investigated in the DVC community, even though it has a noticeable impact on the RD performance. We propose a method to improve the predicted frame by accessing partially decoded data. Our contributions have been published in [26, 100]

One of the main contributions is an investigation into how to best combine the motion estimation and the channel decoding. In this context we focus on how to handle the non-stationarity of the prediction errors. First, we show the performance for manually assigning classes and a classification oracle. Secondly, we show the performance when using information from the motion estimation. Finally, we show the improvement when extracting motion information during decoding. Our contributions have been published in [63, 24, 80]

We provide a benchmark of DVC against conventional video coding. Next to quantifying performance differences we look at ways to bring the performance of DVC closer to the one of conventional video coding. We focus on the complexity of the proposed solution and the trade-off between RD performance and encoder complexity. Our contributions on the performance differences have been published in [28].

1.6. Publications

R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, R.L. Lagendijk. Dependency Channel Modeling for a LDPC-based Wyner-Ziv Video Compression Scheme. *Proceedings 2006 IEEE International Conference on Image Processing,* pp. 277-280, Atlanta, Georgia, USA, October 8-11 2006.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. Motion compensated Prediction in Distributed Video Coding. *Proceedings of the fourteenth Annual Conference of the Advanced School for Computing and Imaging,* , pp. 230-234, Heijen, The Netherlands, June 11-13 2008.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. Motion compensated Prediction in Transform Domain Distributed Video Coding. *Proceedings of the tenth International Workshop on Multimedia Signal Processing,* pp. 332-336, Cairns, Queensland, Australia, October 08-10 2008.

R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, R.L. Lagendijk. Analyzing Symbol and Bit Plane-Based LDPC in Distributed Video Coding. *Proceedings 2007 IEEE International Conference on Image Processing,* , pp. II 17-20, San Antonio, Texas, USA, September 16-19 2007.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. On Extrapolating Side Information in Distributed Video Coding. *Proceeding of the 26th Picture Coding Symposium,* Lisbon, Portugal, November 07-09 2007.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. Analysis of performance losses in Distributed Video Coding. *Proceedings of the 27th conference on Picture Coding Symposium,* pp. 1-4, May 06-08, 2009, Chicago, IL, USA.

R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, R.L. Lagendijk. Finding a Near Optimal Dependency Channel Model for a LDPC-based Wyner-Ziv Video Compression Scheme. *Proceedings of the twelfth annual conference of the Advanced School for Computing and Imaging,* pp. 456-463, Lommel, Belgium, June 14-16 2006.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. Improving motion compensated Extrapolation for Distributed Video Coding. *Proceedings of the thirteenth annual conference of the Advanced School for Computing and Imaging,* pp. 291-297, Heijen, The Netherlands, June 13-15 2007.

R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, R.L. Lagendijk. On The Comparison Of Distributed Video Coding Using LDPC Codes On Bit Plane And Symbol Level. *Proceedings of the thirteenth annual conference of the Advanced School for Computing and Imaging,* pp. 457-462, Heijen, The Netherlands, June 13-15 2007.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. On the generation of side information for DVC. *Proceedings of the Twenty-eighth Symposium on Information Theory in the Benelux,* pp. 141-148, Enschede, the Netherlands, May 24-25 2007.

S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk. Analysis of performance losses in distributed video coding. *Proceedings of the fifteenth Annual Conference of the Advanced School for Computing and Imaging,* pp. 141-148, Enschede, the Netherlands, May 24-25 2009.

S. Sánchez, S. Borchert, R.P. Westerlaken, and R.L. Lagendijk. Non-stationary channel model based on unsupervised motion learning in distributed video coding. *Proceedings of the 30-th Symposium on Information Theory in the Benelux,* Eindhoven, the Netherlands, May 28-29 2009.

2. Challenges in distributed video coding[1]

In this chapter, we give an overview of a number of distributed video coding approaches from academic publications. First, we give a short introduction to the underlying theory to DVC, the Slepian-Wolf [86] (SW) and the Wyner-Ziv [112] (WZ) theorems. Then, we move from the theory to the application in video coding. After presenting two initial DVC approaches from Stanford and Berkeley and focusing on their differences, we present two state of the art approaches from literature. Finally, we extend the scope to the challenges and research questions considered in this thesis. These are channel coding, quantization and motion estimation at the decoder. Of the following three chapters of this thesis, each one will focus on our approach to a particular challenge.

2.1. Underlying theory: Slepian-Wolf and Wyner-Ziv

The Slepian-Wolf theorem addresses the independent encoding of two statistically dependent discrete random sequences, X and Y, which are each stochastically independently and identically distributed (i.i.d.). The independent encoding is radically different from the joint encoding as used in the largely deployed predictive coding solutions. The Slepian-Wolf theorem states that for joint decoding, the minimum rate to independently encode the two (correlated) sources is the same as the minimum rate for joint encoding, with an arbitrarily small error probability. The rate bounds for a vanishing error probability considering two i.i.d. sources are

$$\begin{aligned} R_X &\geq H(X \mid Y) \\ R_Y &\geq H(Y \mid X) \\ R_X + R_Y &\geq H(X, Y) \end{aligned} \tag{2.1}$$

which corresponds to the area identified in Figure 2.1. These bounds imply that the minimum coding rate is the same as for joint encoding (i.e., the joint entropy), provided that the individual rates for both sources are higher than the respective conditional entropies [71]. Since the reconstruction of the two sequences, X and Y, is perfect (neglecting the arbitrarily small probability of decoding error), Slepian-Wolf coding is generally referred to as lossless Distributed Source Coding (DSC) [71].

[1] A substantial part of this chapter is based on a book chapter written by Pereira *et al.* [71]

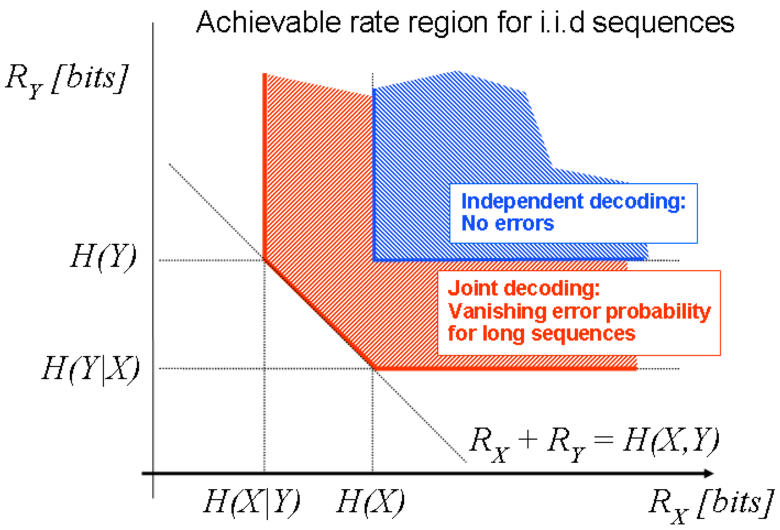


Figure 2.1.: Rate boundaries defined by Slepian-Wolf theorem [86].

The dependency between X and Y, computed at the decoder, is modeled as a virtual dependency channel. As X and Y are not identical, the VDC can be modeled by transition probabilities $P(X|Y)$. Since error-free transmission is desired, the data (virtually) transmitted over the VDC should be protected by error correcting codes. Indeed, in all proposals for DSC the information bits sent over the channel at rate $R_X \geq H(X \mid Y)$ are viewed as (the parity bits of) error-correcting codes. For that reason, DSC relies heavily on efficient channel codes [99]. The rationale is, the more X and Y are correlated, the less errors need to be corrected, i.e. more compression can be achieved. Channel capacity-achieving codes have been shown to reach the performance corresponding to the desired corner points of the Slepian-Wolf region [71] in Figure 2.1.

The work of Slepian and Wolf was later extended to the lossy case by Wyner and Ziv. The Wyner-Ziv theorem deals with lossy compression of source X associated with the availability of the Y source at the decoder, but not at the encoder. This is a particular case of distributed source coding and known as asymmetric coding. The asymmetry is between Y, which is independently encoded and decoded, and X, which is independently encoded, but conditionally decoded. In these conditions, Y is known as side information [71].

The Wyner-Ziv theorem then states that when performing independent encoding with side information under certain conditions there is no coding efficiency loss with respect to the case when joint encoding is performed, even if the coding process is lossy. The conditions are that X and Y are jointly Gaussian, memoryless sequences and a MSE distortion measure is considered. Later, it was shown that only the $X - Y$ difference needs to be Gaussian [72].

Together, the Slepian-Wolf and the Wyner-Ziv theorems suggest that it is possible to compress two statistically dependent signals in a distributed way (separate encoding, joint decoding), approaching the coding efficiency of conventional predictive coding schemes (joint encoding and decoding). Based on these theorems, a new video coding paradigm, known as distributed video coding, has emerged. DVC does not rely on joint encoding of source X and side information Y. Thus, when applied to video coding, the side information needs only be present at the decoder. Not requiring the side information at the encoder typically results in the absence of the temporal prediction loop and hence reducing encoder complexity [71].

2.2. Video coding based on the DSC principles

In the context of video coding, the sequence X becomes the reference video frame and side information Y the motion compensated prediction. The two video frames are temporally correlated. In case of joint encoding, which implies conventional video coding, the correlation is exploited by the temporal prediction loop, i.e. motion estimation and compensation, at the encoder. Consequently, both encoder and decoder have access to the side information Y. Figure 2.2 shows such a predictive video coding system. The simplified block diagram emphasizes the conceptual differences with distributed video coding, depicted in Figure 2.3. A DVC coder can be thought to consist of a quantizer followed by a Slepian-Wolf encoder, as illustrated in Figure 2.3.

The quantization of original frame X yields quantized frame Q. The VDC in DVC is then modeled by the transition probabilities $P(Q|Y)$. After successful decoding, the obtained frame $\hat{Q}$ should be identical to Q, barring a vanishing error probability. In contrast, the reconstructed frame $\hat{X}$ suffers from possible quantization errors and is not identical to X.

In DVC, as opposed to conventional video coding, the temporal prediction loop is

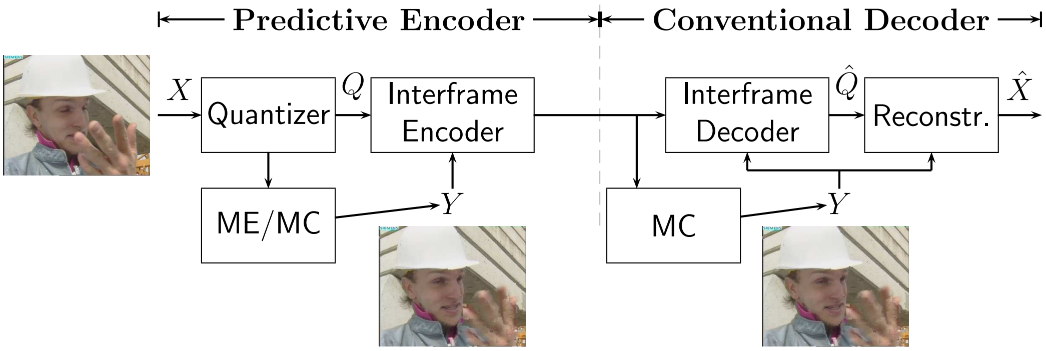


Figure 2.2.: Conceptual block diagram of the basic conventional video coding system.

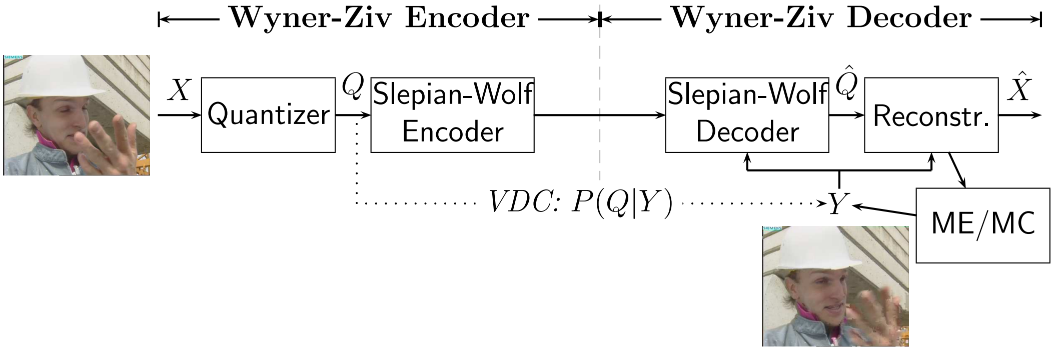


Figure 2.3.: Conceptual block diagram of the basic DVC system.

only present at the decoder. Hence, the encoder in DVC does not have access to Y. Without access to Y, the Slepian-Wolf encoder only takes the reference frame X into account, i.e. it only compresses information that is contained within X. This kind of video compression is called intra frame coding. In case of separate encoding and decoding, intra frame coding is inherently less efficient than inter frame coding, as only the latter exploits temporal correlation. In case of DVC the temporal correlation is to be exploited by means of the joint decoding of X and Y.

To enable joint decoding, the DVC decoder has to generate the side information Y without access to the reference frame X. For the purpose of motion estimation and compensation, the DVC decoder only has access to already decoded frames. In contrast, in conventional coding the motion estimation has access to the current frame. Since the derived motion information is sent to the decoder, both encoder and decoder have access to identical side information Y.

The inherent characteristics, i.e. intra frame encoding and motion estimation without access to the reference frame, constitute the main differences to conventional video coding with inter frame encoding. Replacing source coding in conventional coding with channel coding in DVC also introduces additional constraints to the encoding. For instance, LDPC codes being fixed-rate codes and only accepting a fixed-length input limits the flexibility of mode selection [59]. In conventional video coding, different properties of a video frame can be taken into account by different modes of operations.

2.3. Overview of early DVC systems

Practical design of DVC video codecs started around 2002, following important advances in channel coding technology, especially error-correcting codes with a capacity close to the Shannon limit, like turbo and LDPC codes. The first practical DVC video coding solutions emerged from Stanford University [10, 13, 12] and the University of California, Berkeley [75, 74]. The Stanford architecture is characterized by block-based coding with decoder motion estimation [71]; the Berkeley solution is characterized by frame-based Slepian-Wolf coding, typically using turbo codes, and a feedback channel to perform rate control at the decoder.

One approach to solve the rate control problem, adopted in the Stanford Codec [10, 13, 12], relies entirely on the decoder and feedback information. The decoder determines the optimal encoding rate and sends this information to the encoder. The encoder remains unchanged [47]. Another way to perform rate control is to allow some simple rate estimation at the encoder. For example, in the Berkeley scheme [75, 74], the encoder can store one previous frame to enable rate estimation.

The feedback channel approach from the Stanford WZ video coding solution has two significant drawbacks. Firstly, it requires the presence of a feedback channel. Secondly, it requires real time processing and low latency. To circumvent the need for the feedback channel the encoder needs to handle the rate control problem. As a result, the encoder needs to trade-off complexity versus accuracy. For the rate control to be accurate the encoder should perform motion estimation and compensation to get the side information. In contrast, a rate estimate, based solely on frame difference, will over-estimate the rate and decrease the RD performance in the majority of cases. For instance [29] reports a loss of up to 1.2dB between the pure encoder and decoder rate control solutions. A more complex approach, using machine learning at the encoder [62], still faces the problem of artifacts due to misclassification.

Both schemes adopt a spatial transform, which enables the codec to exploit the statistical

Table 2.1.: Functional differences between Stanford and Berkeley DVC systems

Function	Stanford [12, 11, 47]	Berkeley [75, 74]
encoding strategy	frame-based coding	block-based coding
rate control	decoder rate control	encoder rate control
complexity	simple encoder	smarter, more complex encoder
channel codes	sophisticated channel codes	simple channel codes
overhead data	no auxiliary data	hash codes sent by the encoder
error resilience	less intrinsically robust	higher resilience

dependencies within a frame, thus achieving better RD performance than a pixel domain scheme [47]. We highlight the functional differences between the two early DVC systems (Stanford versus Berkeley) [71] in table 2.1.

2.3.1. Stanford DVC

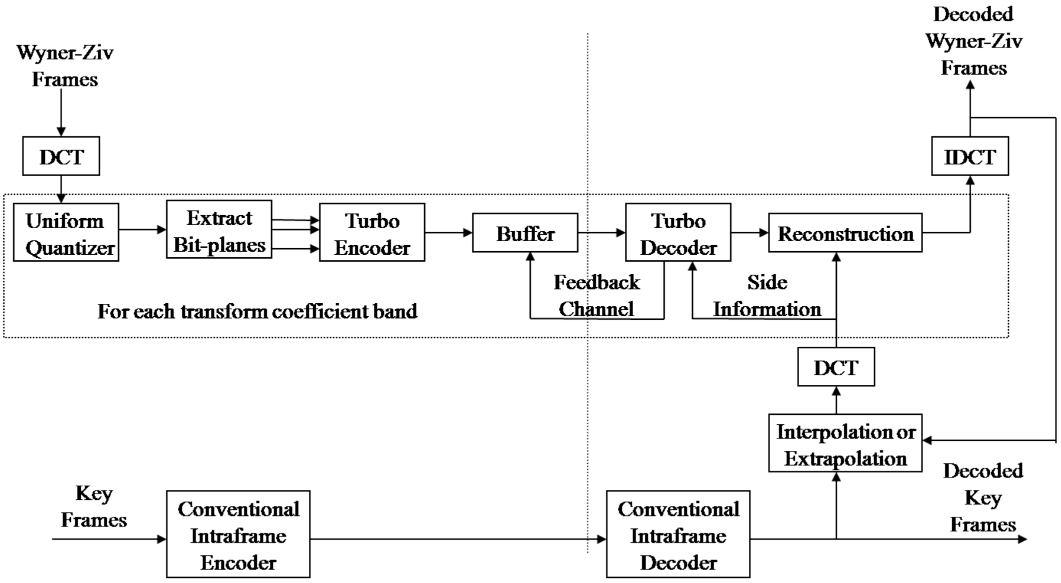


Figure 2.4.: Block diagram of Stanford transform domain DVC video codec [12].

The Stanford video coding architecture was first proposed for the pixel domain [10] and later extended to the transform domain [12]. The transform domain Stanford DVC codec, shown in Figure 2.4, works as follows [71]:

At the encoder

Frame classification The video sequence is divided into WZ frames, that is frames encoded by DVC, and key frames. The key frames are intra coded, using for instance the H.263+ intra or H.264/AVC intra standards. The key frames are periodically inserted, determining the Group of Pictures (GOP) size. It should be noted that although conventional video coding standards like for instance MPEG-2 are flexible, typically at most every 15th frame is made into a key frame [111].

Transform A block-based transform, typically a DCT, is applied to each WZ frame. The DCT coefficients of the entire WZ frame are then grouped together, forming DCT coefficient bands.

Quantization Each DCT band is uniformly quantized. For a given band, bits of the quantized symbols are grouped together, forming bit planes, which are then separately turbo encoded.

Turbo Encoding The turbo encoding of each DCT band starts with the Most Significant Bit Plane (MSB). The parity information generated for each bit plane is then stored in the buffer and sent in chunks upon decoder requests, via the feedback channel.

At the decoder

Side Information Creation The decoder creates the side information for each WZ frame by performing a motion compensated frame interpolation (or extrapolation) using the closest already decoded frames.

Correlation Noise Modeling The residual statistics between corresponding DCT coefficients in the WZ frame and the side information are assumed to be modeled by a Laplacian distribution whose parameter was estimated using an off-line training phase.

Turbo Decoding Once the side information DCT coefficients and the residual statistics for a given DCT coefficient band are known, each bit plane is turbo decoded. The turbo decoder receives from the encoder successive chunks of parity bits following the requests made through the feedback channel, until the decoder uses a request stopping criterion.

Reconstruction After turbo decoding, all the bit planes associated with each DCT band are grouped together to form the decoded quantized symbol stream. Once all decoded quantized symbols are obtained, it is possible to reconstruct all the DCT coefficients with the help of the corresponding high frequency side information coefficients. That is, if no WZ bits were transmitted, the corresponding DCT bands of the side information are used.

Inverse Transform After all DCT bands are reconstructed, an inverse discrete cosine transform (IDCT) is performed, yielding the decoded WZ frame.

Frame Reordering Finally, to get the decoded video sequence, decoded key frames and WZ frames are put in the correct order.

2.3.2. Berkeley DVC codec

The DVC approach from Berkeley is known in literature as PRISM - from Power-efficient, Robust, High-compression, Syndrome-based Multimedia coding [75, 74]. Contrary to the Stanford codec, which is frame-based, the Berkeley approach is block-based. Each block is classified into one of several predefined classes depending on its correlation with the predictor block. Such a predictor can be either a co-located block, or a motion compensated block [74]. The classification stage decides the coding mode for each block. The modes are: no coding, traditional intra frame, or syndrome coding. The coding modes are then transmitted to the decoder as header information.

Since Puri *et.al* have small block-lengths at their disposal (64 samples for an 8x8 block), they use the relatively simple Bose-Chaudhuri-Hocquenghem (BCH) block codes [104] which work well even at reasonably small block-lengths (unlike more sophisticated channel codes such as LDPC codes [46], and turbo codes [22]). In the context, the syndrome is the result of a parity check. If the syndrome is an all-zeros vector, a valid code word has been received. If there are detectable errors, the syndrome will have some nonzero value [85].

The assumption for blocks that fall in the syndrome coding class is that the most significant bits can be inferred from the side information. Thus, only the least significant bits of the quantized DCT coefficients are syndrome encoded. Within the least significant bits, the lower part is source coded. The upper part of the least significant bits is coded using a BCH channel code. In addition, for each block, the encoder sends a 16-bit cyclic redundancy check (CRC) checksum as a signature of the quantized DCT coefficients. The checksum is used at the decoder to select the best candidate block from the side information.

The decoder generates side information candidate blocks, which all correspond to half-pixel accurate displaced blocks in the reference frame, in a window positioned around the center of the block to decode. Each of the candidate blocks plays the role of side information for syndrome decoding. A first step deals with the coset channel coded bit planes and is performed for each candidate block; the other step deals with the entropy decoding of the least significant bit planes and is performed only for the selected block by hash matching. Each candidate block leads to a decoded block, from which a hash signature is generated. In order to select the best matching candidate block, each one is compared with the CRC hash received from the encoder. Candidate blocks are visited until decoding leads to hash matching.

While mainly the Stanford architecture was later adopted and improved by many research groups, over time some of the differences between the two early WZ video codecs disappeared.

2.4. Recent approaches in literature

In this section we present two state of the art approaches from literature. The first approach we consider is the DISCOVER DVC codec [15]. The DISCOVER codec is probably one of the most RD efficient DVC codecs currently available [71]. The second approach is based on rateless LDPC codes [50] and incorporates an encoder-based block classification [59].

2.4.1. DISCOVER Wyner-Ziv video codec

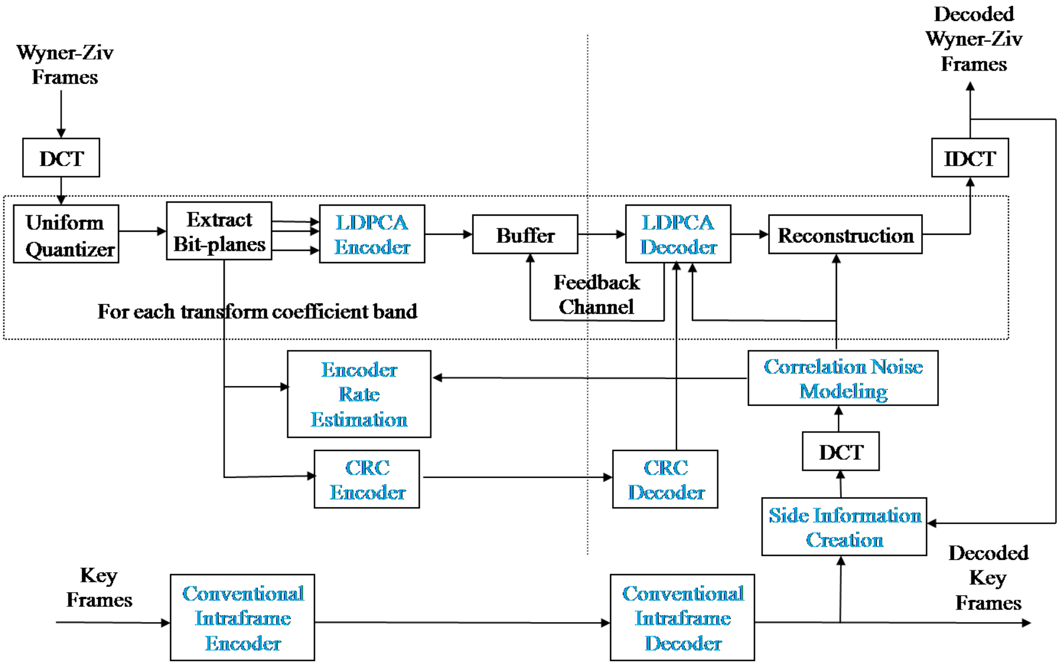


Figure 2.5.: Block diagram of DISCOVER codec [71].

The DISCOVER codec architecture is based on the early Stanford codec, introduced in Section 2.3.1. However, there have been many improvements [15]. The DISCOVER codec as shown in Figure 2.5 provides the following improvements [71]:

1. Optionally, the GOP size can be adapted to the temporal correlation in the sequence [17].

2. In addition to the use of Turbo Codes, also LDPC codes are considered.

3. An encoder rate control has been added. In order to limit the use of the feedback channel, i.e. requests made by the decoder, the encoder estimates an initial number of bits, sent for each bit plane, before any request is made [15]. The rate should be underestimated to prevent RD performance losses. The decoder will complement the rate by making one or more requests over the feedback channel.

4. The side information creation has been improved [17, 31]. The side information is generated based on block matching using a modified Mean Absolute Difference (MAD) to regularize the motion vector field. Then, a hierarchical coarse-to-fine bidirectional motion estimation is performed. Finally, spatial motion smoothing based on a weighted vector median filter is applied to the obtained motion field to remove outliers before motion compensation is finally performed [71].

5. The correlation noise modeling is no longer limited to an off-line training phase. The Laplacian parameter is estimated on-line and at different granularity levels, notably at band and coefficient level [30].

6. A CRC check has been added. To decide whether or not more bits are needed for the successful decoding of a certain bit plane, the decoder uses a simple request stopping criterion. It checks whether all LDPC code parity-check equations are fulfilled for the decoded (hard decision) codeword. If correct, the decoding of the next bit plane or band can start, otherwise another request for a chunk of parity bits has to be made.

The details about the VDC modeling employed in the latest DISCOVER codec can be found in [30, 71]. After applying the 4x4 DCT transform over the residual frame, between the two reference frames, each DCT coefficient is classified into one of two classes [71]. The first class indicates a reliable estimation. The second class corresponds to a block where the residual error is high, which means the side information generation process failed for that block. This information can help the channel decoder, since it is possible to give the decoder confidence information about the side information.

The DISCOVER codec is available for download at [1]. Furthermore, [1] also includes a comprehensive performance evaluation. As indicated in the overview above, the codec tackles problems like rate control as well as being tuned and optimized. For more details regarding the practical issues and the performance we refer the reader to the project website [1].

2.4.2. Rateless LDPC codec with skip and intra modes

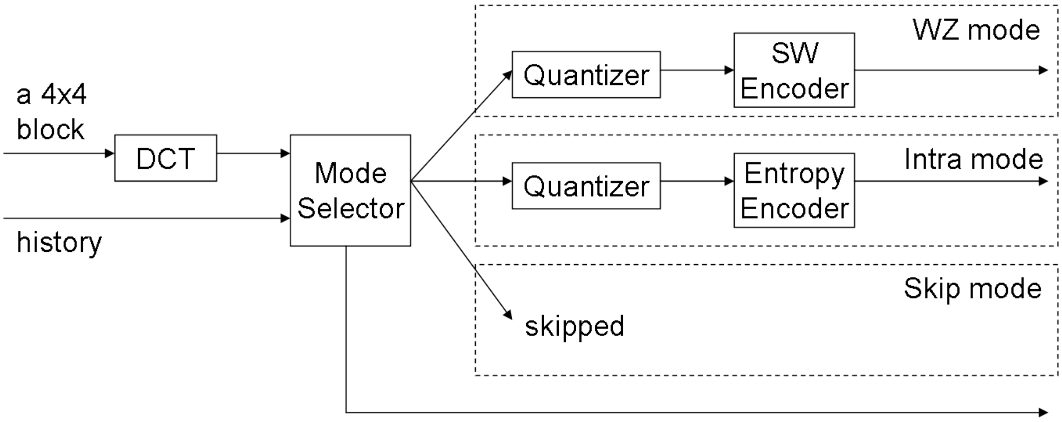


Figure 2.6.: WZ video compression with skip and intra mode selection [59].

DVC using rateless LDPC codes was first introduced in [50] and later extended in [59]. In addition to reducing the storage complexity in comparison to fixed-rate LDPC codes, rateless LPDC codes allow seamless integration with mode selection. The mode selection is depicted in Figure 2.6. Next to the three modes for WZ frames, the concept of key frames is still present. A sequence is coded using the GOP format *I-WZ-P-WZ-P...*, where I and P denote H.264 coded intra-predicted and single-list inter-predicted frames respectively, and WZ denotes a Wyner-Ziv coded frame [50].

As the number of WZ coded blocks becomes arbitrary, depending on how many blocks are skipped or intra coded, fixed-rate LDPC codes can not be applied to the cumulative WZ mode. Furthermore, the block length is too small to efficiently LDPC encode individual WZ blocks.

With the mode selection blocks can be classified according to their properties. Regarding their correlation with the decoder side information DCT blocks can be roughly classified into three distinct categories. The categories are: nearly identical, correlated and nearly independent [59]. The former two correspond to skip and WZ mode, the last one corresponds to the intra mode. As a result the WZ mode is limited to blocks that are correlated. Therefore, the WZ mode does not need to deal with the sub-optimal cases for the channel coding, namely near identity [50] or near independence [59].

As the encoder is complexity constrained, the complexity of mode selection is limited. The mode selection employs a history, which contains buffered DCT coefficients from the previous frame. With the history it is easy to determine which DCT blocks should be skipped [59]. For the iterative algorithm to find the best partition of a video frame into WZ and intra mode, we refer the reader to [59].

To our best knowledge, this codec [59] is the best performing DVC codec in terms of RD performance. The necessary components for such a good performance are the combination of ratelesss LDPC codes with skip and intra blocks. In addition, the quality of the side information is high as the interpolation is done from the neighboring key frames. Furthermore, the rate estimation proposed in [59] removes the need for a feedback channel.

However, all advantages are paid for by increasing the encoder complexity. Classifying skip, coded and intra blocks the encoder requires frame comparison. Thus both memory access and energy consumption increase significantly. In addition, half of the frames are coded as key frames. In this context it should be noted that every second frame is actually coded in the conventional predictive way, i.e. as H.264 inter-predicted frame. In summary, the distinction between a predictive conventional video codec and the WZ scheme with mode decision is not clear-cut anymore. It then becomes questionable what the benefits are of DVC schemes over an efficient implementation of for instance H.263.

2.5. Challenges in DVC

Recent years have seen a significant number of publications in the field of DVC. Many of the publications build on and improve the early WZ video codecs introduced in Section 2.3. Especially the Stanford architecture was later adopted and improved by many research groups around the world [71].

The focus of this thesis are the inherent performance limitations of DVC with low encoder complexity. For that purpose, we focus on the following three differences between conventional video coding and DVC as outlined in Table 2.2. In this thesis we focus on very low encoder complexity. Introducing for instance frame differencing requires frame buffering which significantly increases the complexity. Hence, we only consider intra coding, without any inter operability.

2.5.1. Channel coding

One enabling factor for DVC were important advances in channel coding technology, especially error-correcting codes with a capacity close to the Shannon limit. The two most frequently used codes in DVC are turbo codes [22] and LDPC codes [46]. Both

Table 2.2.: Functional differences conventional video coding and DVC

Component	Conventional video coding	DVC
coding	source coding	channel coding
ME/MC	access to reference frame	only access to decoded frames
quantization	variable-rate inter DCT coefficients	fixed-rate intra DCT coefficients

codes have capacity-achieving performance, but literature states LDPC codes can better approach the capacity of a variety of communication channels than turbo codes [92]. In addition, capacity can only be obtained for block lengths converging to infinity.

Next to the block length, an accurate channel model is crucial for the performance of channel codes. In the application field of error protection over a physical channel, statistical and physical modeling can be combined. For example in wireless communications the channel is often modeled by a random attenuation (known as fading) of the transmitted signal, followed by additive noise [106]. For the VDC, there is no physical channel. What needs to be modeled are prediction errors in the side information.

For the channel coding we are interested in which channel code to use and the VDC modeling. After using the popular turbo codes for a long time [70], the latest DISCOVER codec uses an LDPC Accumulate codec [71]. Since the latter is similar to the channel code we use, we refer the reader to **Chapter 3** for a more detailed evaluation and a comparison of turbo [10, 13, 12, 90, 38, 70] and LDPC codes [92, 71].

Since the actual statistics of the VDC (difference between the original video frame and the side information) are not known, a parametrized distribution has to be assumed. Most approaches in literature, including the Stanford, DISCOVER and rateless LDPC codes introduced previously, model the VDC as a Laplacian distribution. In **Chapter 3**, we review multiple alternatives and investigate how accurate the VDC needs to be modeled.

We establish that the VDC is non-stationary. Contrary to the Berkeley and rateless LDPC codes, which employ mode selection at the encoder, we focus on the decoder. In **Chapter 6** we investigate several decoder-based classification schemes. Recently, the DISCOVER codec also introduced classification at the decoder [30, 71].

2.5.2. Motion estimation and compensation

The motion estimation and compensation at the decoder, i.e. the side information generation, plays a crucial role for the RD performance. Both [58] and [88] addressed the problem of motion estimation in DVC by using the known methods from conventional video coding. Hence, the first method [58] uses H.264 motion vectors. The second paper [88] uses a similar method to get the motion vectors but then applies a spatial smoothing to the vector field.

We take a different approach and consider known methods not from conventional video coding, but from frame rate up-conversion [41, 65, 35, 36, 77, 39]. This work reports, that for the purpose of predicting the side information from neighboring frames, true motion is beneficial [69, 31, 37].

For the motion compensation step, the main focus is on interpolation for most systems in literature [14, 17, 37, 31, 87, 15, 113] which in terms of side information outperforms extrapolation for very small GOP sizes. Only if low delay is required, extrapolation is considered instead of interpolation [12, 11, 67]. In **Chapter 4** we argue, that taking the key frames into account extrapolation is in fact the better choice for DVC. For that purpose we will present our extrapolation scheme and compare it with interpolation.

To investigate the relation to conventional video coding, we also consider an extrapolation scheme with access to the reference frame X. This scheme is not practical in DVC and its results are provided both as an upper bound and as a comparison to motion compensation in conventional video coding.

2.5.3. Quantization

Quantization has a large impact on the RD performance. The inter coefficients in conventional video coding contain many zero coefficients. In general a large section of the tail end of the zig zag scanned coefficients will consist of zeros. By contrast, the intra frame DCT coefficients in DVC exhibit significantly less zero runs and larger DCT coefficients [81].

The quantization in DVC is not well represented in literature yet. Most schemes fix the quantization levels for certain rate distortion points [2, 15]. Also residual DCT coefficients are considered [66]. Other schemes, like the rateless LDPC scheme in Section 2.4.2 alleviate the problem of fixed quantization levels by using different modes for different block properties. The latter approaches require frame buffering at the encoder, increasing both encoder complexity and RD performance.

In **Chapter 5** we investigate the quantization of intra DCT coefficients. We argue, that the quantization should not be completely fixed. We consider several levels of adaptivity and their impact on the RD performance.

3. Virtual dependency channel: model and codes

In Chapter 2 we introduced channel coding for compression purposes as one of the main challenges in DVC. From literature, we highlighted turbo and LDPC codes. In this chapter, we focus on channel encoding the virtual dependency channel in DVC, i.e. how to model the VDC.

First, we focus on fundamental aspects, starting from the information theoretic bound. Then we report on the performance of the considered state of the art channel codes. Further, we consider models for the VDC. We evaluate their suitability to deal with prediction errors. Finally, we evaluate whether bit plane-based coding, favored in literature, can approach the performance of the more sophisticated symbol-based coding.

3.1. Background

Instead of quantizing and entropy encoding a motion compensated (transformed) frame difference, the Slepian-Wolf encoder generates compressed data at rate $R_X \geq H(X \mid Y)$ in a quite different fashion. The side-information (motion compensated prediction) Y available at the decoder is viewed as a by-channel-errors corrupted version of the video frame X being compressed at the encoder.

DVC literature mainly focuses on two efficient channel codes, namely turbo [10, 13, 12, 90, 38, 70] and LDPC codes [92, 71]. These state of the art channel codes are reported to be capacity achieving [22, 40]. They use a soft decoding procedure, like (near) optimum belief propagation [33]. The performance of the soft decoding is highly dependent on the transition probabilities $P(X|Y)$ of the VDC.

In the scope of this chapter we focus on practical design choices with regard to the channel codes. That encompasses choice of code, model and whether to code symbols or bits. We will revisit the VDC in Chapter 6. There we will focus on the problem of non-stationarity and how to take it into account. Initially, in this chapter we focus on a stationary VDC model.

3.1.1. Channel capacity

In his work work, Shannon [84] investigates the capacity of a channel to transmit information. Shannon proves that for the noiseless channel, the channel capacity is determined by the entropy. The fundamental theorem is as follows:

Let a source have entropy H (bits per symbol) and a channel have a capacity C (bits per second). Then it is possible to encode the output of the source in such a way as to transmit at the average rate $\frac{C}{H} - \epsilon$ symbols per second over the channel where ϵ is arbitrarily small. It is not possible to transmit at an average rate greater than $\frac{C}{H}$.

In general, if the channel is noisy it is not possible to reconstruct the original message or the transmitted signal with certainty by any operation on the received signal. Consequently, Shannon investigates ways of transmitting the information which are optimal in removing noise [84]. For that purpose he introduces a correction channel, which enables the receiver to correct errors:

If the correction channel has a capacity equal to $H(X|Y)$ it is possible to so encode the correction data as to send it over the channel and correct all but an arbitrarily small fraction of the errors. This is not possible if the channel capacity is less than $H(X|Y)$.

This theorem relates directly to the Slepian-Wolf theorem with $R_X \geq H(X \mid Y)$. However, both theorems just provide a lower bound for the needed rate. They do not provide a coding method with the desired properties, but show that such a code must exist in a certain group of codes. For channel coding there are practical state of the art codes, reported to be capacity achieving. While DVC employs the same state of the art codes, it is an open question whether they can achieve capacity for the VDC.

3.1.2. Capacity achieving channel codes

In information theory, turbo codes are a class of high-performance Forward Error Correction (FEC) codes developed in 1993, which were the first practical codes to closely approach the channel capacity [22]. Turbo codes are finding use in (deep space) satellite communications and other applications where designers seek to achieve reliable information transfer over bandwidth or latency constrained communication links in the presence of data-corrupting noise. Turbo codes are nowadays competing with LDPC codes, which provide similar performance [110].

LDPC codes originate from the 1960's and were proposed by Gallager [46]. Only recently they were first applied as channel codes, after they were reinvented by MacKay and Neal [61]. Consequently, implementation of LDPC codes has lagged that of turbo codes. Nevertheless, LDPC codes are finding increasing use in applications where reliable and highly efficient information transfer over bandwidth or return channel constrained links in the presence of data-corrupting noise is desired [108].

Turbo and LDPC codes use soft-decision decoding. Whereas a hard-decision decoder operates on data that take on a fixed set of possible values (typically 0 or 1 in a binary code), the inputs to a soft-decision decoder may take on a whole range of values in between. The extra information indicates the reliability of each input data point, and is used to form better estimates of the original data and to modify the input to a further decoding iteration. Therefore, a soft-decision decoder will typically perform better in the presence of corrupted data than its hard-decision counterpart [73].

Soft-decision codes are designed in such a way that they exploit the dependency information during channel decoding. To do so, an accurate channel model is necessary. Both turbo and LDPC codes perform well on noisy network communication channels. Such a channel can be modeled physically by trying to calculate the physical processes which modify the transmitted signal [106]. Whether turbo and LDPC codes perform well on the VDC depends on how accurate the virtual channel can be modeled.

3.1.3. Modeling choices for the VDC

In DVC we have to model the virtual dependency channel, which is an effect of motion compensated prediction. The statistical dependence between X and Y is then modeled as a virtual dependency channel analogous to Additive White Gaussian Noise (AWGN) channels or Binary Symmetric Channels (BSC) [16]. Both channel assumptions produce simple mathematical models, making them easier to analyze [103, 105].

In communications, the additive white Gaussian noise (AWGN) channel model is one in which the only impairment is a linear addition of wide band or white noise with a constant spectral density and a Gaussian distribution of amplitude. The model does not account for the phenomena of fading, frequency selectivity, interference, nonlinearity or dispersion. However, it produces simple and tractable mathematical models which are useful for gaining insight into the underlying behavior of a system before these other phenomena are considered [103].

The BSC is a binary channel; that is, it can transmit only one of two symbols (usually bit values 0 and 1). The transmission is not perfect, and occasionally the receiver gets the wrong bit. That such a bit is "flipped" occurs with a small probability (the "crossover probability"). This channel is often used by theorists because it is one of the simplest noisy channels to analyze. Many problems in communication theory can be reduced to a BSC. On the other hand, being able to transmit effectively over the BSC can give rise to solutions for more complicated channels [105].

In DVC, the symbols themselves or extracted bit planes can be encoded. The symbol-based approach is more general, while the bit plane-based approach can reduce the decoder complexity significantly [97]. The VDC modeling varies, depending on which implementation is chosen. For the Symbol-based Slepian-Wolf Decoder (S-SWD) we need a Symbol-based Dependency Model (S-DM), that describes the relation between the quantized symbols Q and the side information symbols Y with $P(Q|Y)$. The necessary information can be extracted directly from the dependency model between X and Y. The Bit plane-based SW Decoder (B-SWD) needs a Bit plane-based Dependency Model (B-DM). Such a model describes the relation between the bit planes of Q and side information Y and can be derived from the dependency between X and Y.

To implement dependency models in a practical LDPC coder, we need estimates of the probabilities $P(Q|Y)$, $P(Q^b|Y^b)$, $P(Q^b|Y)$ and $P(Q^b|Y, Q^{b+1}, ..., Q^{L-1})$. Here, b denotes the current bit plane and L the total number of bit planes. The probabilities have been derived in [97] and can be found in the Appendix A. $P(Q^b|Y^b)$, $P(Q^b|Y)$ and $P(Q^b|Y, Q^{b+1}, ..., Q^{L-1})$ each correspond to one B-DM we consider for B-SWD. Each dependency model, ranging from simple to sophisticated, leads to a different decoding strategy.

The statistics of $P(Q|Y)$ are only known after decoding. To model the conditional probabilities a Probability Density Function (PDF) is used [109]. A PDF is a function that describes the probability for a certain value for Q to occur given a certain side information Y. Since the underlying PDF can only be observed after decoding, we have to provide an estimate. The decoder only has access to Y to estimate the transition probabilities $P(Q|Y)$. Further, it has to account for motion compensated prediction errors. Consequently, inaccuracies in the modeling are inevitable. In this context, we will focus on the model choice for the PDF and the sensitivity of the estimate.

3.1.4. Differences turbo/LDPC coding

In theory, the two best performing state of the art channel codes are capacity achieving. Hence, their upper bound for compression is identical. The question then becomes how their performance differs in practice. The first relevant condition is the robustness against an inaccurate channel model. Not all regions in a frame can be modeled correctly. How turbo and LDPC codes compare in presence of inaccurate channel modeling will be investigated further after introducing the models we consider.

The second relevant condition is the block length on which to apply the channel coding. The longer the block length, the closer channel codes approach the Shannon bound. In

DVC, the block length is limited by the number of pixels in a frame, or for the PRISM approach in Chapter 2 even in a block. Consequently, we observe a practical advantage for LDPC codes, since, [40] reports that LDPC codes approach the performance of the best known turbo codes for significantly smaller block lengths. Nevertheless, DVC literature favors turbo [10, 13, 12, 90, 38, 70] codes over LDPC codes [92, 71].

3.2. Modeling the VDC

Since the actual statistics of the VDC are not known, a distribution has to be estimated. For such a distribution there are at least two important factors. The first is the statistical model and the second is the corresponding parameter(s). Since the parameter has to be estimated in a DVC system, it is not accurate.

We assume the noise in the VDC is zero-mean and i.i.d.. This assumption is justified because the interleaving in turbo coding [95] and the random construction of the parity-check matrix in LDPC [108] removes any kind of dependencies between pixels. It should be noted that LDPC codes are defined by a sparse parity-check matrix. The sparse matrix is often randomly generated, subject to sparsity constraints [108]. Following literature on modeling of natural images [56, 53, 79, 64, 115, 23], we consider the following PDF models $f_N(n)$ for the VDC.

Laplacian density

$$f_N(n) = \frac{1}{\sqrt{2\sigma_n^2}} exp\left(\frac{-\sqrt{2}|n|}{\sigma_n}\right), \tag{3.1}$$

Gaussian density

$$f_N(n) = \frac{1}{\sqrt{2\pi\sigma_n^2}} exp\left(\frac{-n^2}{2\sigma_n^2}\right), \tag{3.2}$$

Generalized Gaussian density

$$f_N(n) = \frac{\nu\alpha(\nu)}{2\sigma_x\Gamma(\frac{1}{\nu})} exp\left(-\left(\alpha(\nu)\frac{|x|}{\sigma_x}\right)^{\nu}\right), \tag{3.3}$$

Two-sided Gamma density

$$f_N(n) = \left(\frac{\sqrt{3}}{8\pi\sigma_n|n|}\right)^{\frac{1}{2}} exp\left(-\frac{\sqrt{3}|n|}{2\sigma_n}\right). \tag{3.4}$$

Here σ_n denotes the standard deviation of the noise in the VDC, $\Gamma(\,\cdot\,)$ the Gamma function and ν the shape parameter of the Generalized Gaussian density. For the Generalized Gaussian, we set the second parameter, the shape parameter ν to the manually chosen value $\nu = 0.25$, which we experimentally found to best fit the data. It should be noted, that a lower ν yields a peaked distribution. The Generalized Gaussian contains the Gaussian distribution for $\nu = 2$ and the Laplacian distribution for $\nu = 1$ as special cases [115].

Using a Generalized Gaussian gives more freedom in changing the shape of the distribution. It is then possible to find a better match with the real distribution. However, to achieve extra flexibility in a practical system, two parameters have to be predicted at the decoder. Consequently, there is an additional source of errors and the parameter estimation becomes more difficult.

3.3. Evaluation of best code and VDC model combination

Instead of focusing on the accuracy of the PDF match alone, we consider the end-to-end performance. In the following, we concentrate on finding a distribution that gives overall good performance. By *overall good performance* we mean a channel model that allows for high compression ratios and minimal sensitivity to deviations in the estimated parameters. To provide generality, we focus on symbol-based modeling and coding in this section. Initially, we apply only the Laplacian density, favored in literature, to both turbo and LDPC codes. Thereafter, we chose the best code and focus on the best PDF choice.

3.3.1. Practical turbo and LDPC code to evaluate

We use a **symbol-based implementation** of both turbo and LDPC codes. The symbols of the original frame X are first quantized. The Slepian-Wolf coding is then done on the quantized symbols Q. The codes are implemented as follows:

Turbo Slepian Wolf Encoder [63] The Slepian-Wolf encoder is implemented as two identical 16-state convolutional constituent codes with rate 4/5 and with parity polynomials (23,35,31,37,27). The complete encoder uses two interleavers, i.e., both convolutional coders are preceded by an interleaver. Interleaving is done on symbol level. Only the non-systematic bits of the two convolutional coders are transmitted to the decoder, since the systematic part is estimated by the decoder-based on the side information Y. To obtain a specific bit rate R_X, the output of the convolutional coders is (randomly) punctured.

Turbo Slepian Wolf Decoder [63] At the decoder, a prediction of the current video frame is made available, based on the temporal information in the past. The decoder consists of two SISO (Soft-Input Soft-Output) maximum likelihood decoders for the symbols/pixels Y. In our setup the decoders are serially concatenated and both are preceded by the corresponding interleaver. Extrinsic information is passed between the constituent SISO decoders and the number of bit errors is decreased after every iteration.

The extrinsic information is *a posteriori* information and represents the reliability of a decoded bit. Extrinsic information is used for decoding in following iteration [54]. The constituent SISO decoders use a Maximum *a posteriori* (MAP) algorithm [21] to provide the conditional probabilities $P(Q|Y)$. The MAP algorithm assumes that the underlying characteristics of the VDC, modeled as Laplacian, are given.

LDPC Slepian Wolf Encoder [99] At the encoder side the syndrome Z is determined by $Z = \mathbf{H}Q$, where $\mathbf{H}$ is a regular parity check matrix over GF(q) with a uniform number of non-zero values in each row J and a uniform number of non-zero values in each column K. The non-zero entries in H are generated randomly and placed according to the design rules in [60], such that the short cycles in the bipartite graph representing H are kept to a minimum. The rate R is defined as

$$R = \frac{\#syndromes}{codewordlength} = \frac{J}{K} \tag{3.5}$$

where the symbol node degree (J) in the employed setup is 3 and the check node degree $K \geq J$.

LDPC Slepian Wolf Decoder [99] The decoder uses a computationally efficient soft decoding procedure, known as (near) optimum belief propagation algorithm [33]. The decoder computes the vector of decoded (quantized) input $\hat{Q}$ that is the most likely, under the restriction that $\mathbf{H}\hat{Q} = Z$ over GF(q), with the probability of $\hat{Q}$ given by the conditional probability $P(\hat{Q}|Y)$ of the VDC. The latter PDF models the correlation between source video frames and decoder side-information, i.e. it models the virtual dependency channel.

Both, SISO decoding performance of turbo codes and belief propagation of LDPC codes, depend heavily on VDC characteristics, and providing incorrect model information may drastically degrade the performance. In both cases the performance can be measured as bit rate where we still have near perfect reconstruction. That is the bit rate for which $\hat{Q}$ is equal to Q or almost equal with only a very small number of errors. The number of errors can be measured as Bit Error Rate (BER).

3.3.2. Turbo vs LDPC coding evaluation

To evaluate the coding performance of turbo and LDPC codes, we conduct a controlled experiment. We use a synthetic video sequence, shown in Figure 3.1. The sequence contains a picture-in-picture which moves to the left by 5 pixels each frame, while the background moves in opposite direction by 5 pixels.

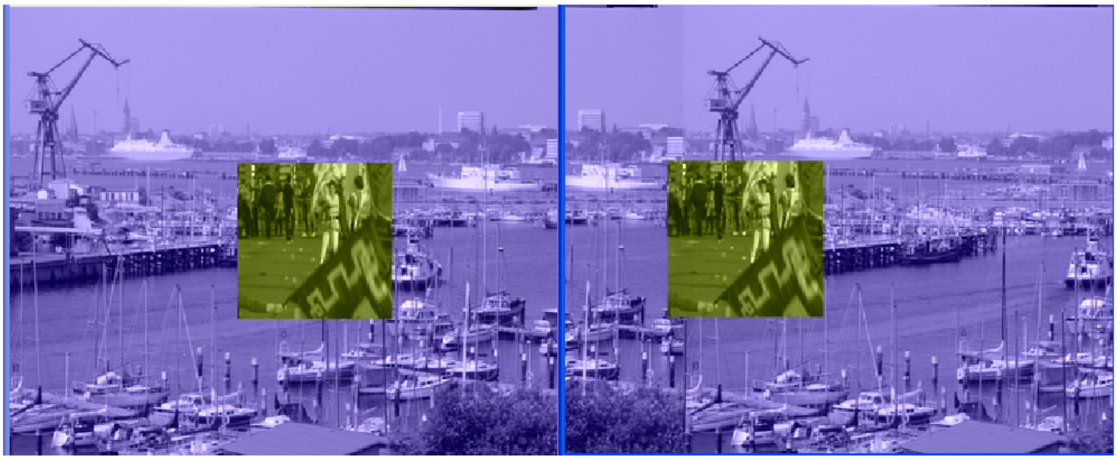


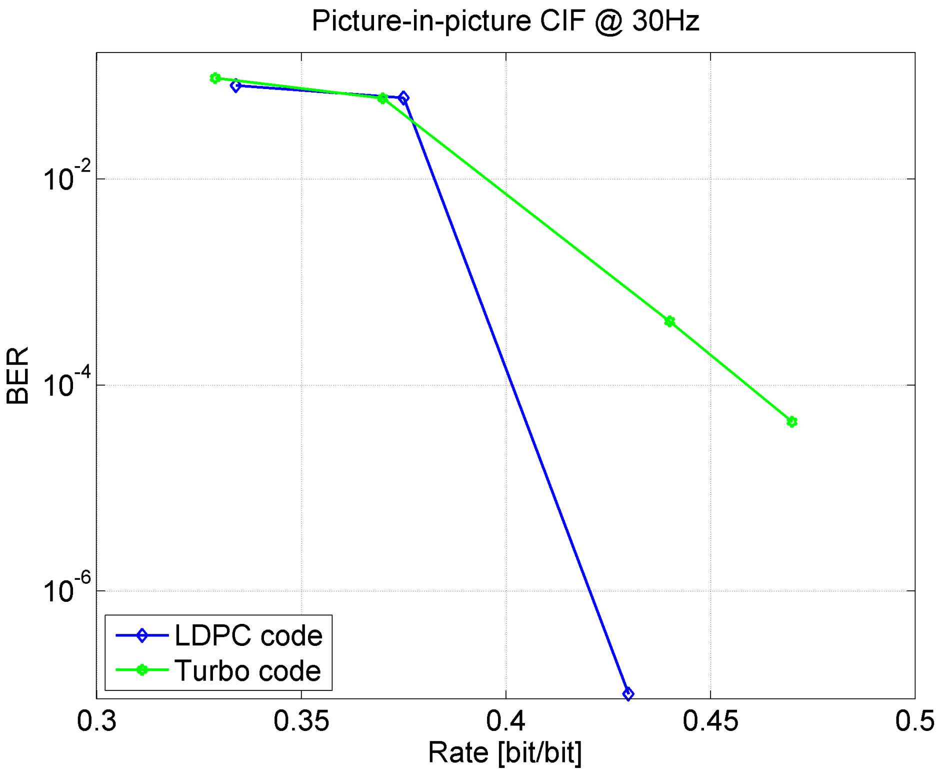
Figure 3.1.: First and last (18th) frame of the Picture-in-picture sequence. Foreground and background move opposed to each other by 5 pixels per frame.

The image content is detailed real image data on CIF (Common Intermediate Format) resolution (352x288 pixels) with a frame rate of 30 frames per second. We chose the Picture-in-picture sequences because it combines relevant (image) data with simple and traceable motion. The latter enables us to locate prediction errors.

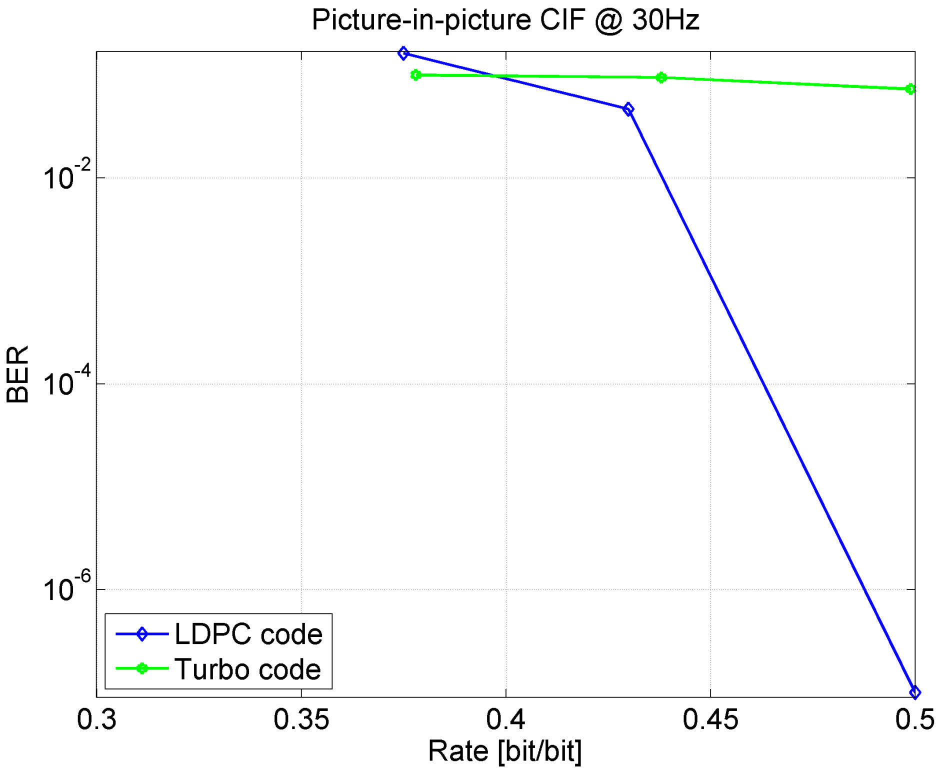
The Probability Mass Functions (PMF) $P(Q|Y)$ are initially measured from the real quantized input and the estimated side information. It should be noted, that this oracle PMF is not available in DVC. Consequently we also consider a VDC, that is not measured from the real data, but modeled as a Laplacian distribution with a fixed variance. To account for differences in the side information quality, we combine two variances. First, we manually chose ($\sigma^2 = 1$) to account for well predicted areas and second ($\sigma^2 = 510$) to account for poorly predicted areas.

Figure 3.2 shows the performance of the system for the turbo and LDPC codes, in terms of the number of remaining errors versus bit rate. If the PMFs are measured from the real data in Figure 3.2 (a) we observe that the BER converges faster to zero. If the PMFs are assumed to follow the synthetic Laplacian model in Figure 3.2 (b) the performance of both codes degrades.

With a synthetic Laplacian model, the LDPC code degrades moderately and needs 0.07 bit/bit more to converge to (close to) zero BER in Figure 3.2 (b). By contrast, the turbo



(a) measured PMF



(b) modeled PMF

Figure 3.2.: Results for coding subframe with (a) measured PMFs and (b) a synthetic Laplacian channel model.

code degrades significantly, resulting in a large rate increase for convergence towards zero BER. This observation implies that LDPC codes are less sensitive to inaccurately chosen channel models. In addition to the increased robustness, literature reports a better performance of LDPC codes for shorter block lengths [40]. Based on these two advantages, we focus on LDPC codes in the remainder of this thesis.

3.3.3. Evaluation of considered models

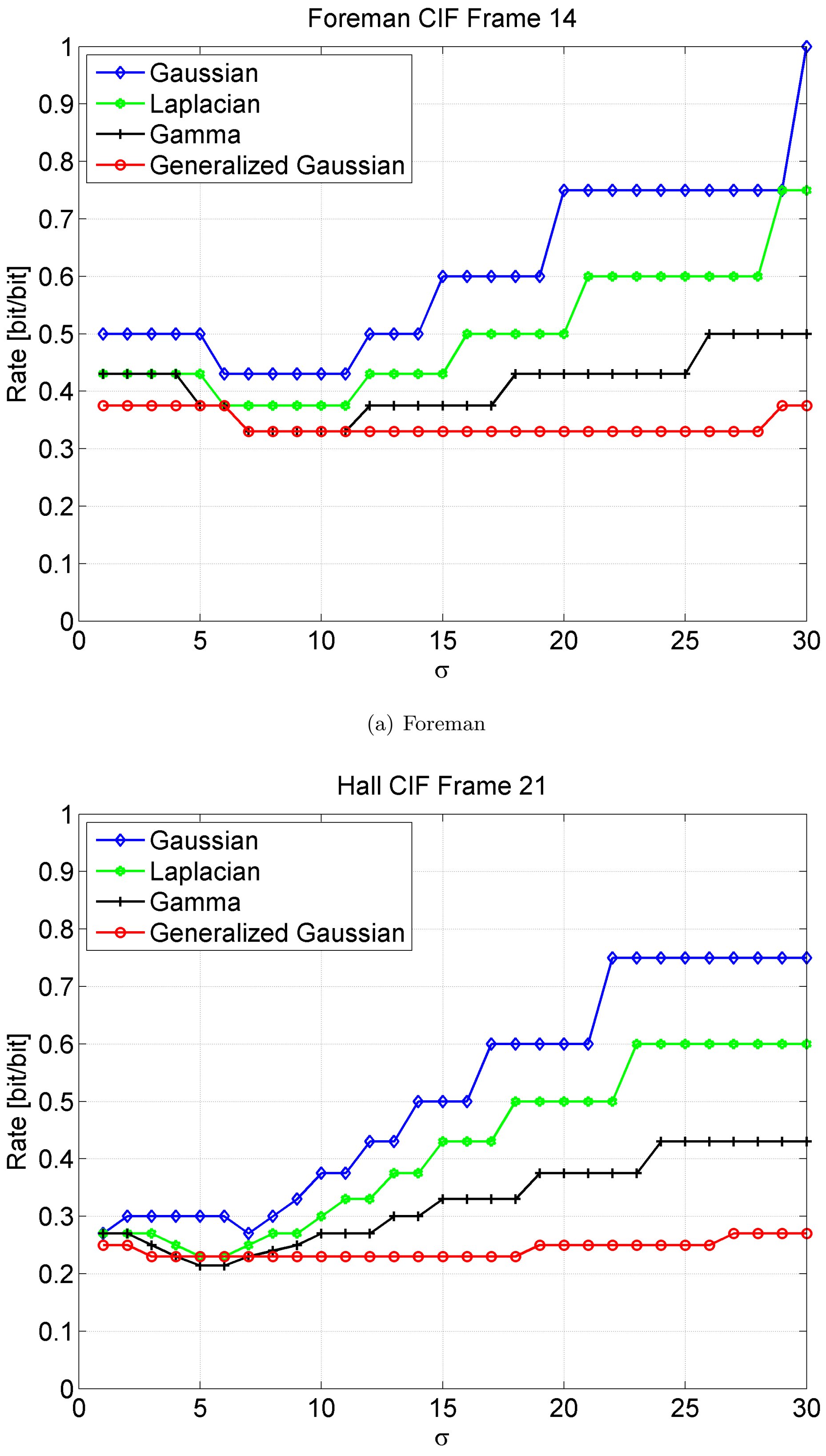
In Section 3.2 we introduced the four models we consider. The common parameter in the Laplacian, Gaussian, Generalized Gaussian and two-sided Gamma density is the standard deviation σ. To investigate the accuracy of the models for natural video, we use real video sequences with the symbol-based LDPC code presented in Section 3.3.1. The sequences are the high motion Foreman and the low motion Hall-monitor sequences in CIF resolution with a frame rate of 30 frames per second.

The LDPC coder setup is explained in Section 3.3.1. The symbol node degree J is set to 3 and the check node degree $K \geq J$. With this implementation only a discrete set of bit rates can be achieved $(1, 3/4, 3/5, 3/6, ...)$. Figure 3.3 shows the lowest rate for which decoding is successful, depending on the standard deviation σ employed by the PDF models.

First, we are interested the global minimum for each PDF, which indicates the best compression that can be achieved with that model. Second, we are interested in the range of σ for which the lowest rate can be achieved, which indicates the robustness of that model against inaccurate parameter choices. Increasing the rate results in a larger range of σ for which decoding is possible.

Figure 3.3 shows the results for one frame from each sequence. The overall best performing distributions are then the two-sided Gamma and Generalized Gaussian distributions. These models provide the lowest global minimum and at the same time exhibit the best robustness.

While the Generalized Gaussian distribution shows better performance, it requires the estimation of two parameters. To make analysis easier, we focus on simple models with a single parameter and consequently an easier parameter estimation. For that reason, we use the single parameter two-sided Gamma distribution to evaluate the robustness of the parameter estimation.



(a) Foreman

(b) Hall-monitor

Figure 3.3.: Bit rate versus parameter value σ for decoding of (a) the 14th frame of the Foreman sequence and (b) the 21st frame of the Hall-monitor sequence.

3.3.4. Robustness of the parameter estimation

As the previous section illustrated, for a particular bit rate a tolerance range of the PDF parameter value exists, for which decoding is possible. The decoder has to make sure that the estimated PDF parameter value falls inside the admissible range. This situation is visualized in Figure 3.4. The minimum σ_{min} and the maximum σ_{max} value of the PDF parameter for which perfect reconstruction is possible is shown together with the real (yet unknown) σ_k. The values $\hat{\sigma}_k^{(i)}$ indicate the estimated parameter value for frame k.

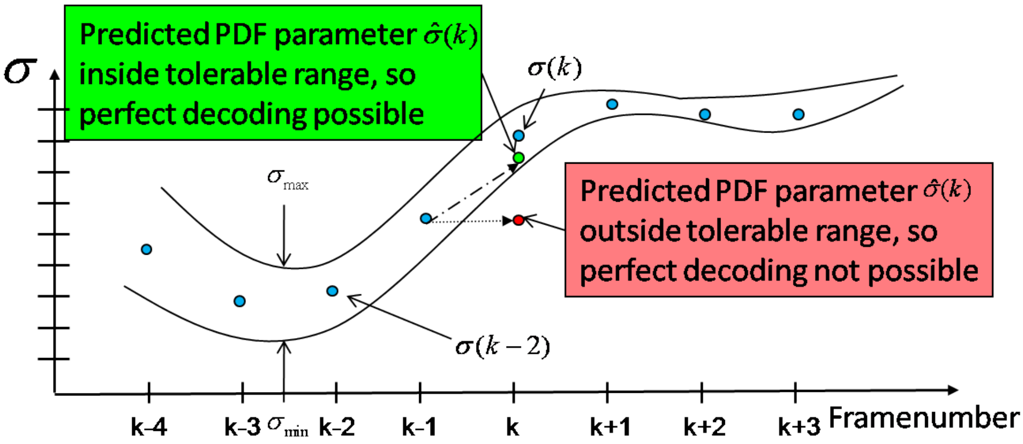

Predicted PDF parameter $\hat{\sigma}(k)$ inside tolerable range, so perfect decoding possible
σ
$\sigma(k)$
$\sigma_{\max}$
Predicted PDF parameter $\hat{\sigma}(k)$ outside tolerable range, so perfect decoding not possible
$\sigma(k-2)$
k-4
k-3
$\sigma_{\min}$
k-2
k-1
k
k+1
k+2
k+3
Framenumber

Figure 3.4.: Plot of σ parameter per frame. The tube indicates the range in which the parameter value should be chosen.

If the estimated value $\hat{\sigma}_k^{(1)}$ falls inside the range $(\sigma_{min}, \sigma_{max})$, perfect reconstruction is possible. However, if the estimated value is $\hat{\sigma}_k^{(2)}$, perfect decoding is not possible. There are two possible reason for $\hat{\sigma}_k^{(2)}$ to be outside the range $(\sigma_{min}, \sigma_{max})$. First, the compression factor is too large, yielding an empty range $(\sigma_{min}, \sigma_{max})$. This situation can only be remedied at the encoder by applying less compression. Second, the range $(\sigma_{min}, \sigma_{max})$ is not empty but the soft channel decoder has not used an admissible $\hat{\sigma}_k^{(i)}$. This situation can be remedied at the decoder, either by searching different values for the PDF parameter or by using better procedures for estimating admissible $\hat{\sigma}_k^{(i)}$ values.

From Section 3.3.3 we know that the bit rate applied by the encoder determines the size of the range $(\sigma_{min}, \sigma_{max})$. More compression makes it harder for the decoder to select an admissible parameter value for the dependency channel. However, the smoother the temporal behavior of the range $(\sigma_{min}, \sigma_{max})$, the easier a admissible value of $\hat{\sigma}_k^{(i)}$ can be estimated from $\hat{\sigma}_{k-1}^{(1)}$,$\hat{\sigma}_{k-2}^{(1)}$,... .

3.3.5. Evaluation of the parameter robustness

We use real video sequences with the symbol-based LDPC code presented in Section 3.3.1. The sequences are the high motion Foreman and the low motion Hall-monitor sequence in CIF resolution. Here, we present the admissible parameter range for which decoding is possible, given a fixed rate R (3/8 for Foreman and 3/15 for Hall-monitor).

For each video frame the parameter range $(\sigma_{min}, \sigma_{max})$ is determined for different bit rates. The results for the Foreman sequence and for the Hall-monitor sequence are shown in Figure 3.5. The connected lines indicate the admissible parameter range for shown rate. If successive points are not connected, the intermediate video frame could not be decoded correctly, i.e. the bit rate was too low for that particular frame. In addition we show the best fit σ, i.e. the σ which provides the best fitting Gamma PDF measured from the data.

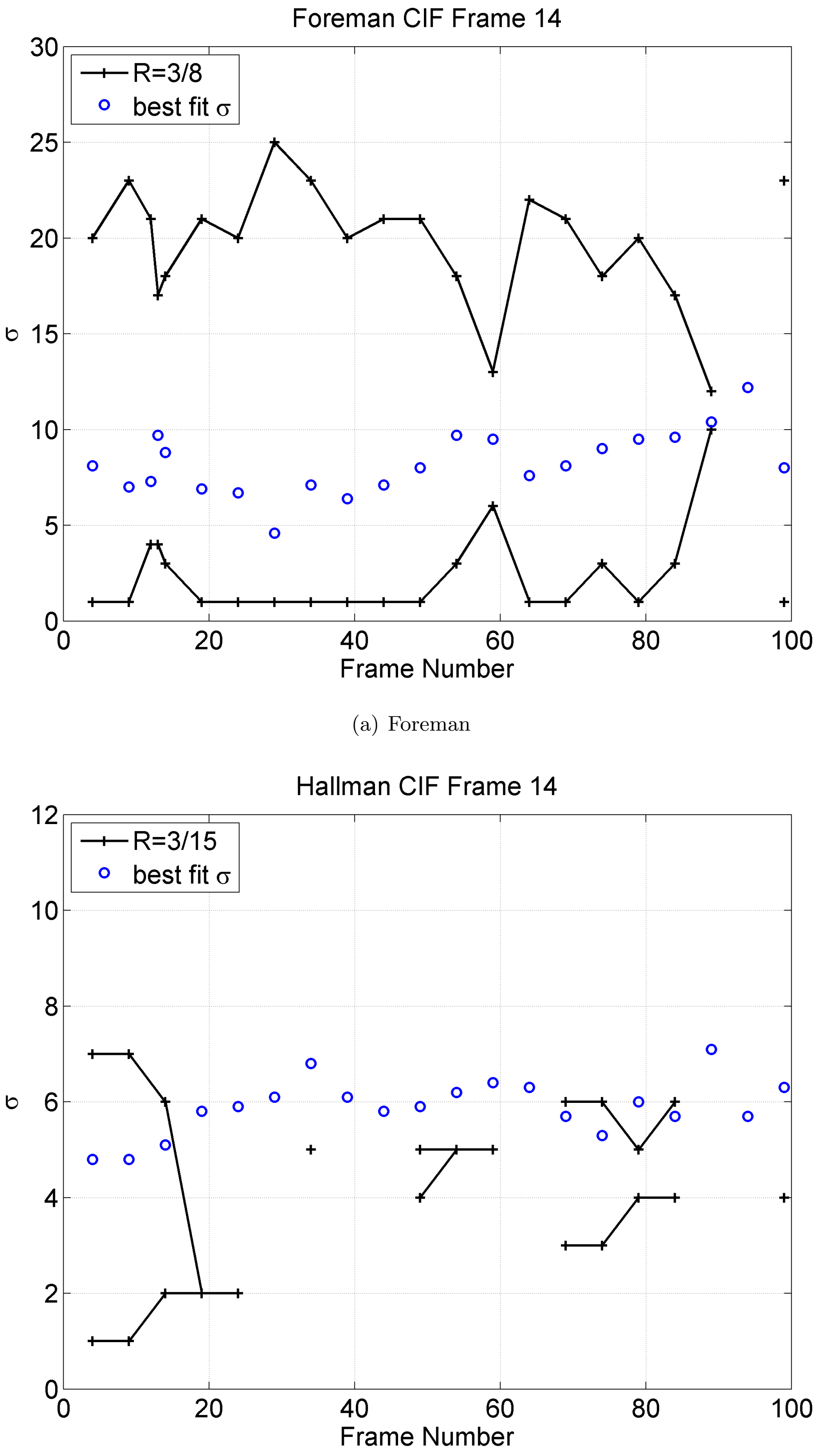
For many frames in Figure 3.5 it holds that the larger the best fit σ, the smaller the admissible range. But the midpoints of the admissible rate do not fluctuate that much. This suggests, that estimating the parameter value at the decoder side is not an unrealistic task. However, for the best compression performance in practice we have to use the lowest bit rate and the lower the bit rate, the higher the accuracy required of σ.

Especially Figure 3.5 (b) shows that already very small variations in the best fit σ can make the difference between successful and failed decoding for a low bit rate. There are instances when a higher best fit σ can be decoded while decoding fails for a lower one. In addition, under-estimating σ is better than over-estimating it. In some cases it even has to be under-estimated for successful decoding, i.e. the admissible range is lower than the best fit σ. These characteristics indicate that one global parameter (and consequently one global distribution) is not sufficient to model the VDC reliably.

3.4. Bit plane-based LDPC coding

The most complex part in current distributed video coding systems is the LDPC decoding [71]. Up till now we focused on a symbol-based implementation of LDPC for generality. As shown in Figure 3.6, the symbols of the original frame X are first quantized by a 2^L-level quantizer and SW coding is done on the symbols of Q and Y.

For the bit plane-based approach shown in Figure 3.6, the Slepian Wolf coding is done on the L bit planes of Q separately. In terms of complexity, a symbol-based LDPC coder is roughly L times (L denotes the number of bit planes) more complex than a bit plane-based one [97]. The bit plane-based approach is well represented in literature, as it was already used in the initial Stanford architecture [10, 12].



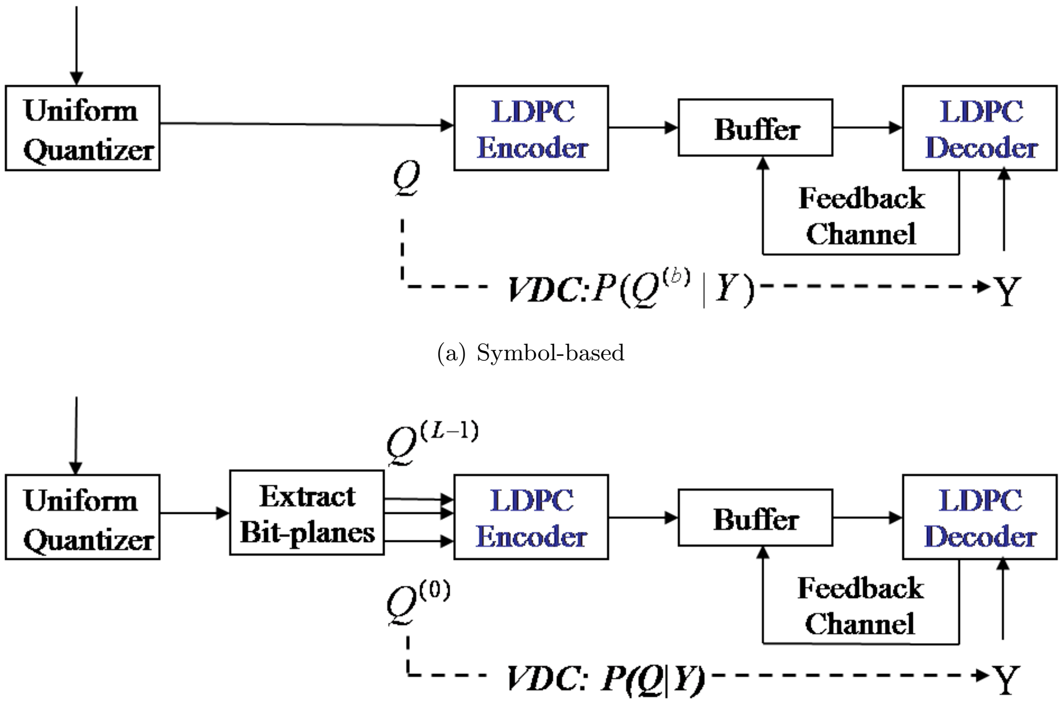
(a) Foreman

(b) Hall-monitor

Figure 3.5.: Admissible parameter range and best fitting parameter for Gamma PDF for (a) Foreman sequence $R = 3/8$ and (b) Hall-monitor sequence $R = 3/15$

3.4.1. Bit plane-based dependency models

The first question that needs to be answered is how to choose the dependency models for S-SWD and B-SWD in order to compare both approaches. In many studies simplified models are used to describe the bit plane-based dependency [12, 38]. It is either seen as a BSC dependency [12] or directly calculated from the earlier S-DM [38]. Both approaches oversimplify the actual dependency.



(a) Symbol-based

(b) Bit plane-based

Figure 3.6.: Conceptual block diagram of (a) symbol-based and (b) bit plane-based DVC.

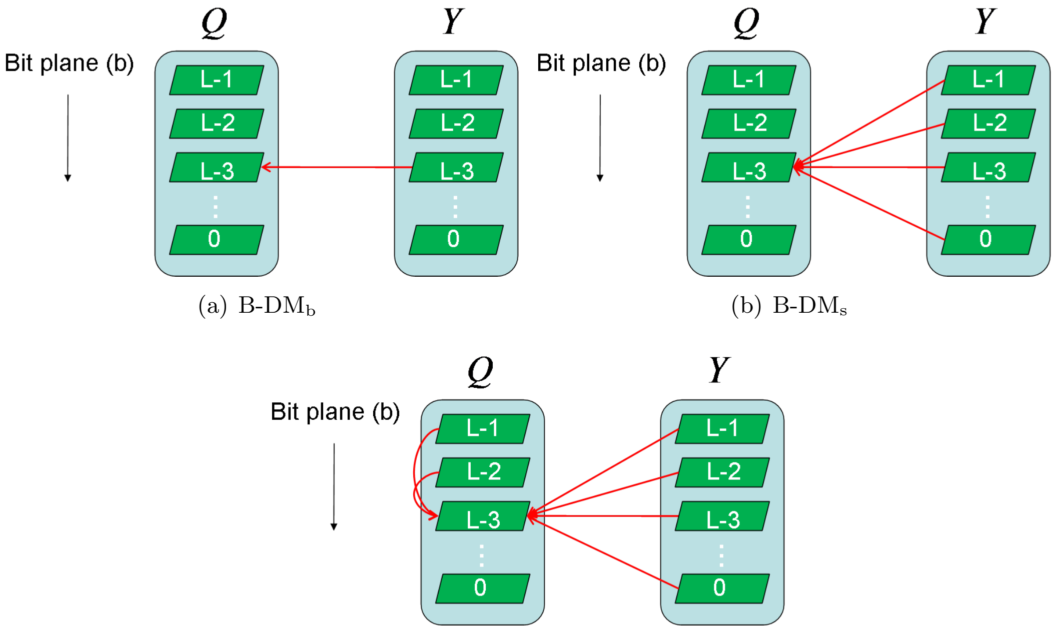
The minimal achievable rate to have lossless Slepian Wolf coding is expressed in terms of the conditional entropy between Q and Y. With knowledge of $P(Q|Y)$ and $P(Y)$ calculating the conditional entropy is straightforward in the case of S-SWD and given by $H(Q|Y)$. In the case of B-SWD the rate is dependent on the type of information that is taken into account in the conditional entropy measure. We investigate three different B-SWD models:

1. The dependency model takes the corresponding bit plane of Q and Y into account.

For the $\mathbf{B\text{-}DM_b}$ dependency model we require $P(Q^b|Y^b)$.

2. The dependency model takes the bit plane of Q and all bit planes, i.e. the symbol, Y into account. For the $\mathbf{B\text{-}DM_s}$ dependency model we require $P(Q^b|Y)$.

3. The dependency model takes the bit plane of Q and the symbol Y together with already decoded bit planes of Q into account. For the $\mathbf{B\text{-}DM_{s+b}}$ dependency model we require $P(Q^b|Y, Q^{b+1}, ..., Q^{L-1})$.



(a) B-DM$_\mathrm{b}$

(b) B-DM$_\mathrm{s}$

(c) B-DM$_\mathrm{s+b}$

Figure 3.7.: Model dependencies between bit planes for (a) B-DM$_\mathrm{b}$, (b) B-DM$_\mathrm{S}$ and (c) B-DM$_\mathrm{s+b}$.

The three dependencies are visualized in Figure 3.7. The order of the list from one to three indicates the complexity of the dependency incorporated in the model. By using all available information at the decoder side, that is B-DM$_\mathrm{s+b}$, the minimal achievable rates for S-SWD and B-SWD are *identical* since:

$$\begin{aligned} H(Q|Y) &= H(Q^{(0)},\ldots,Q^{(L-1)}|Y) \\ &= H(Q^{(0)},\ldots,Q^{(L-2)}|Y) + H(Q^{L-1}|Y) \\ &= \sum_{b=0}^{L-2} H(Q^{(b)}|Y, Q^{(b+1)},\ldots,Q^{(L-1)}) \\ &\quad +H(Q^{(L-1)}|Y), \end{aligned} \tag{3.6}$$

where $Q^{(0)}...Q^{(L-1)}$ are the L bit planes of the quantized symbol Q. After decoding the first bit plane $Q^{(L-1)}$ based on side information Y, every bit plane $Q^{(b)}$ with $b = (L-1, L-2, ..., 0)$ takes not only Y into account, but also the already decoded bit planes $Q^{(L-1)}...Q^{(1)}$.

The models B-DM$_{\mathrm{b}}$ and B-DM$_{\mathrm{s}}$ result in an increase in entropy since:

$$\begin{aligned} H(Q|Y) &= \sum_{b=0}^{L-2} H(Q^{(b)}|Y, Q^{(b+1)},\ldots,Q^{(L-1)} \\ &\quad +H(Q^{(L-1)}|Y) \\ &\leq \sum_{b=0}^{L-1} H(Q^{(b)}|Y) \\ &\leq \sum_{b=0}^{L-1} H(Q^{(b)}|Y^{(b)}). \end{aligned} \tag{3.7}$$

where the bit plane Q^b only depends either on the side information Y or the corresponding side information bit plane Y^b.

Consequently, the B-DM$_{\mathrm{b}}$ and B-DM$_{\mathrm{s}}$ models will result in a performance loss when compared to the B-DM$_{\mathrm{s+b}}$ and S-DM in Eq. (3.6). If we rank the expected performance of all models, we expect S-DM and B-DM$_{\mathrm{s+b}}$ to perform the best, and B-DM$_{\mathrm{b}}$ to have the worst performance.

3.4.2. Evaluation of bit plane decoding

We incorporate our dependency models into the bit plane-based codec from [91], which allows source rates from 2/66 to 66/66 [8]. We use real video data to account for practical DVC coding. We use the first 100 frames of the Foreman sequence in Quarter Common Intermediate Format (QCIF) with 176x144 pixels and a frame rate of 30 frames per second. The side information Y is an extrapolated prediction of X, on which we elaborate in the next chapter.

Due to the implementation from [91] the dependency between X and Y is estimated by a Laplacian PDF with zero mean. The variance of the PDF is estimated from frame Q and predicted side information Y and consequently an oracle estimation. For all possible Y values per frame we calculate $P(Q = q|Y = y)$ for each Q and Y value and then calculate the entropy $H(Q|Y)$:

$$H(Q|Y) \quad = \quad -\sum_{q}\sum_{y} P(Q = q|Y = y)P(Y = y)logP(Q = q|Y = y) \qquad (3.8)$$

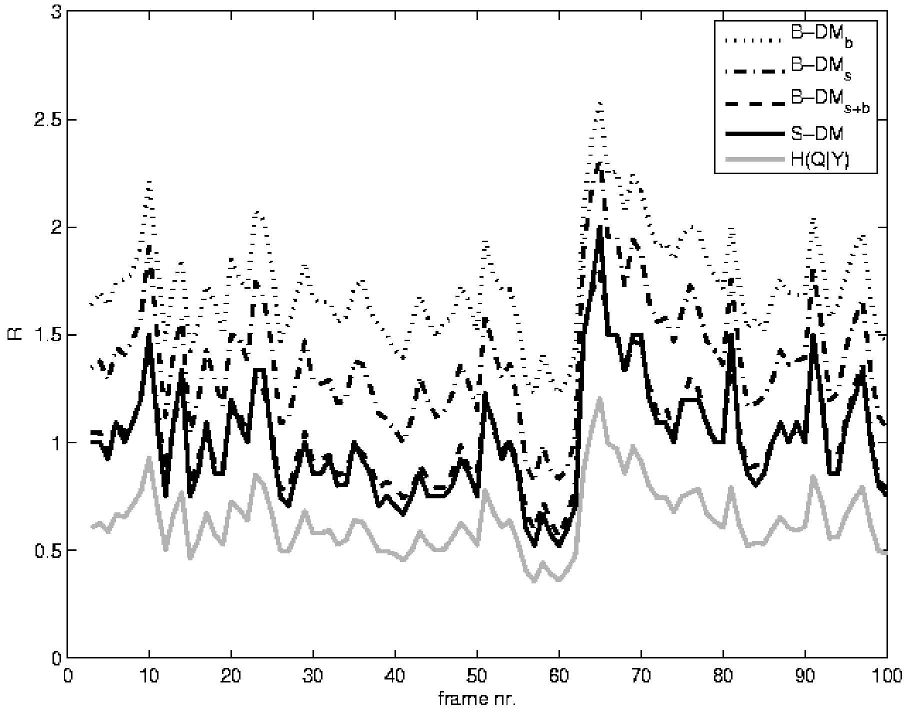


Figure 3.8.: Coded bit rate results using different dependency models for the symbol and bit plane-based distributed video coders for the Foreman sequence.

Figure 3.8 shows the minimal bit rates needed to decode. First, **S-DM** and **B-DM$_{\mathbf{s+b}}$** perform equally and both outperform the **B-DM$_{\mathbf{b}}$** by 1 bit/symbol and **B-DM$_{\mathbf{s}}$** by 0.5 bit/symbol. The differences between **S-DM** and **B-DM$_{\mathbf{s+b}}$** are caused by the limited number of coding rates for the symbol and bit plane-based DVC coder. Hence, **S-DM** and $\mathbf{B\text{-}DM_{s+b}}$ perform equally.

We further observe the following in Figure 3.8. There is a gap of up to 1 bit/symbol between the conditional entropy $H(Q|Y)$ and **S-DM** and $\mathbf{B\text{-}DM_{s+b}}$. The gap indicates, that despite oracle estimation of the variance, one global PDF is not sufficient to model the VDC reliably.

3.5. Discussion

In this chapter we investigated three aspects of channel coding in DVC. Our findings can be summarized as follows:

LDPC codes are superior to turbo codes
We find in Section 3.3.2 that LDPC codes are less sensitive than turbo codes to inaccurate VDC model parameters. In particular, LDPC codes outperform turbo codes if the model for the channel is assumed to follow a chosen PDF distribution, in this case Laplacian. We put forward that LDPC codes are to be preferred over Turbo codes because of the superior performance for shorter block length, as reported in literature, and our results on the reduced sensitivity for inaccurate channel models.

Channel models and their parameters
We show the importance of accurate channel modeling. First, an analysis shows that more complex models, like two-sided Gamma or Generalized Gaussian, outperform simpler models like Gaussian or Laplacian for symbol- and pixel-based DVC. Second, the best RD performance can only achieved within a certain parameter range for these models. The range depends on both, the used model and the amount of compression. For the best compression, the margin for error in estimating the parameter is very small.

Bit plane-based coding superior to symbol-based coding
We show that a bit plane-based LDPC coder is preferred for the purpose of distributed video coding when using an appropriate dependency channel model. Both the symbol-based and the best bit plane-based approach can perform equally.

While the performance of both approaches, a symbol-based and the proposed bit plane-based one, is approximately similar, the symbol-based LDPC decoding is roughly L times (L denotes the number of bit planes) more complex. Based on these findings we use bit plane-based LDPC coding for the remainder of this thesis.

4. Motion estimation at the decoder

In Chapter 2 we introduced motion estimation at the decoder as one of the main challenges in DVC. We highlighted the two dominant prediction schemes in literature, extrapolation and interpolation. In this chapter, we focus on a comparison of the two schemes.

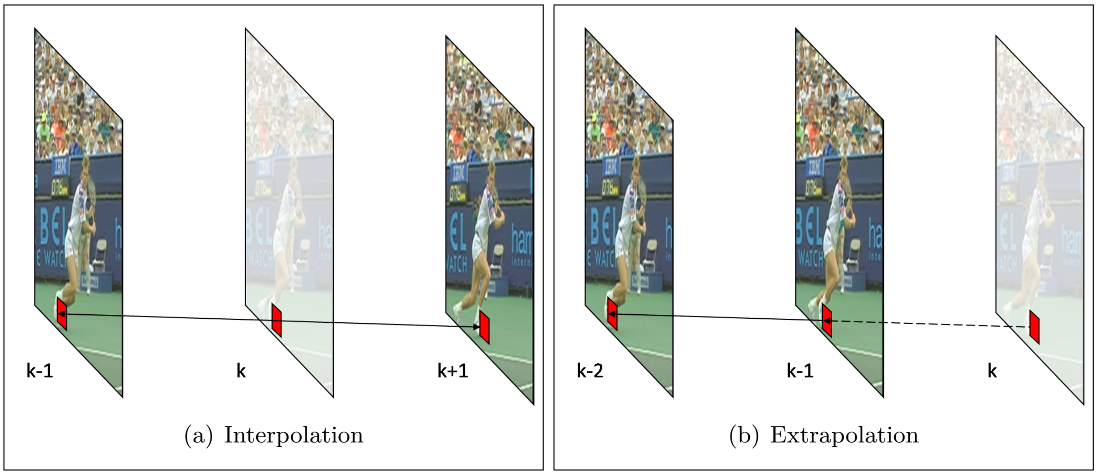
First, we focus on fundamental aspects, starting from an entropy-based consideration. Then we present how extrapolation and interpolation compare in terms of information theory. Further, we present the actual schemes we consider to generate side information. As these schemes include motion estimation, we discuss our motion estimators and their application. Finally, we evaluate the options we presented. We do this both from a side information quality perspective and from the system RD performance perspective.

4.1. Background

As opposed to conventional coding it is not possible in a DVC system to use the reference frame for motion estimation. Thus, we need at least one key frame, which can be decoded without side information. It is only possible to use either such an intra coded key frame or an already decoded Wyner-Ziv frame as input for the ME/MC. The methods we consider in this chapter are depicted in Figure 4.1. That is, we can either interpolate between available frames, or extrapolate from past frames. The former can be compared to B-frames and the latter to P-Frames in conventional video coding.

Clearly, both interpolation and extrapolation have advantages and disadvantages. Current literature mainly focuses on interpolation [14, 17, 37, 31, 87, 15, 113]. The main advantage of interpolation is access to information from past and future frames. However, this information is only useful if the temporal distance is small enough, i.e. if the GOP-size is small. Such a small GOP-size has the drawback of a large number of intra coded key frames. The key frames are either expensive in terms of bit rate or increase the encoder complexity significantly. For instance with a GOP-size of 2, the lowest encoder complexity is at least 50% of the key frame encoding complexity.

In literature, extrapolation is considered mainly for low latency [12, 11, 67]. In the comparison of interpolation with a GOP-size of 2 and extrapolation, interpolation was clearly superior in terms of prediction PSNR quality in [58, 88]. However, this advantage



(a) Interpolation

(b) Extrapolation

Figure 4.1.: Conceptual difference between predicting frame k by interpolation and extrapolation.

only holds for very small GOP-sizes, already at a GOP-size of 4, [17] reports a significant quality decrease for interpolation. Since already decoded WZ-frames can be used, the temporal distance in extrapolation is independent from the GOP-size. Therefore, the main advantage of extrapolation is given by either possible bit rate savings by sending less intra coded key frames, or for rateless LDCP coding in Section 2.4.2 lower complexity by sending less inter coded key frames.

4.1.1. Entropy comparison

First, we consider the trade-off between prediction quality and key frame cost from an information theoretic point of view. Since the reconstruction is perfect for lossless compression, essentially only the rate is required as performance measure. The floor to this compression is defined by the entropy. As long as all information in the source is preserved, the entropy is the fundamental limit [82].

Desired compression rates in practice make the use of lossy compression inevitable. Next to the rate, we need an additional quality measure. For that purpose the distortion in the reconstructed data, i.e. the difference between the original frame X and the reconstructed frame $\hat{X}$, is used. The two extreme cases of the trade-off between minimizing the rate and keeping the distortion small is to transmit no information or to keep all information [82]. Of the two extreme cases only the latter is useful and analyzing the lossless case gives important insights to the expected behavior of lossy coding.

As an example consider 5 frames. For the lossless transmission of these frames, we have to send the joint entropy $H(X_1, X_2, X_3, X_4, X_5)$. According to the chain rule, it is possible to express the joint entropy with either of the two following equations:

$$\begin{aligned} H(X_1, X_2, X_3, X_4, X_5) = {} & H(X_4|X_1, X_2, X_3, X_5) + H(X_2|X_1, X_3, X_5) \\ & + H(X_1, X_3, X_5) \end{aligned} \tag{4.1}$$

$$\begin{aligned} H(X_1, X_2, X_3, X_4, X_5) = {} & H(X_5|X_1, X_2, X_3, X_4) + H(X_4|X_1, X_2, X_3) \\ & + H(X_3|X_1, X_2) + H(X_2|X_1) + H(X_1) \end{aligned} \tag{4.2}$$

Eq. (4.1) reflects the interpolation case with a GOP-size of 2 (*I-WZ-I-WZ-I*), i.e. key and WZ frames alternate, and Eq. (4.2) the extrapolation case with one initial key frame (*I-WZ-WZ-WZ-WZ*). The two entropies are equal. However, in practice key frames are encoded independently. Consequently dependencies between key frames are not exploited at all. If we take the independent frame-by-frame encoding of the key frames into account, we observe a larger loss for the interpolation case since $H(X_1, X_3, X_5) \leq H(X_1) + H(X_3) + H(X_5)$ than for the extrapolation case with $H(X_1)$.

In practice, not all available frames are used for the motion estimation. Since temporally close frames have by far the highest influence, the motion estimation schemes in DVC use mostly 2 frames. For interpolation we have the adjacent past and future frame and for extrapolation the two closest past frames. Thus simplifying Eqs. (4.1) and (4.2) for a single WZ encoded frame yields:

$$H(X_2|X_1, X_3) = H(X_1, X_2, X_3) - H(X_1) - H(X_3|X_1) \tag{4.3}$$

$$H(X_3|X_2, X_1) = H(X_1, X_2, X_3) - H(X_1) - H(X_2|X_1) \tag{4.4}$$

Here, we observe that both terms only differ by the last element, $H(X_3|X_1)$ for interpolation in Eq. 4.3 and $H(X_2|X_1)$ for extrapolation in Eq. (4.4). Since the temporal correlation between adjacent frames is higher, the interpolated frame needs $H(X_3|X_1) - H(X_2|X_1)$ less bit rate. This counteracts the higher cost for independent key frame coding. Thus, we observe an advantage for interpolation in terms of prediction quality and an advantage for extrapolation in terms of key frame cost.

To quantify the trade-off between prediction quality and key frame cost, we consider an example with 16 frames. For interpolation with a GOP-size of 2 we have 8 key frames and 8 interpolated WZ frames. A similar extrapolation case can have as little as 1 key frame and 15 extrapolated WZ frames. We use H.264 to simulate quantitative numbers for the three frame type entropies.

Intra coded key frames To quantify $H(X_k)$ we consider H.264 coded I-frames

Interpolated WZ frames To quantify $H(X_k|X_{k-1}, X_{k+1})$ we consider H.264 coded B-frames

Extrapolated WZ frames To quantify $H(X_k|X_{k-1}, X_{k-2})$ we consider H.264 coded P-frames with 2 past references

To quantify the entropies with lossy coding, all frame types are set to the highest possible H.264 quality (QP=0). At this quality, the PSNR of the reconstruction is above 70 dB, which is nearly lossless. We use the Foreman sequence in CIF resolution with 352x288 pixels and a frame rate of 30 frames per second.

For the three frame types we then use the average rate over all available frames. Consequently, the rate for the intra coded key frames is the average rate of 300 H.264 I-frames, the rate for the interpolated WZ frames is the average rate of 150 H.264 B-frames and the rate for the extrapolated WZ frames is the average rate of 299 H.264 P-frames.

For our example with 16 frames the interpolation case then yields:

$$\begin{aligned} R_I &= 8 \cdot H(X_k) + 8 \cdot H(X_k|X_{k-1}, X_{k+1}) \\ &= 8 \cdot (\text{ H.264 I-frame avg. rate}) + 8 \cdot (\text{ H.264 B-frame avg. rate}) \\ &= 8 \cdot 82.3kB + 8 \cdot 53.1kB = 1083.2kB. \end{aligned} \tag{4.5}$$

The extrapolation case produces:

$$\begin{aligned} R_E &= 1 \cdot H(X_k) + 15 \cdot H(X_k|X_{k-1}, X_{k-2}) \\ &= 1 \cdot (\text{ H.264 I-frame avg. rate}) + 15 \cdot (\text{ H.264 P-frame avg. rate}) \\ &= 1 \cdot 82.3kB + 15 \cdot 55.7kB = 917.8kB. \end{aligned} \tag{4.6}$$

This example, albeit only an approximation, clearly shows an advantage of extrapolation in the lossless case. The high key frame cost outweighs the relatively small gain from using B-frames over P-frames. In the entropy consideration we find extrapolation to have an advantage over interpolation in terms of RD performance. While the entropy only provides a lower bound, we expect the advantage of extrapolation to hold in practice.

4.1.2. Motion estimation

The block-based motion estimators used in state of the art video coding standard H.264 [102] are designed to reduce the residue between predicted and original frame in combination with keeping vector cost minimal. As such these methods are called minimum residue block matching. Their purpose is to reduce the remaining residue $X - Y$

as much as possible. For the purpose of predicting Y from neighboring frames, as required in DVC, it is not clear whether such a minimum residue block matching is optimal.

The minimum residue block matching is block-based and as such implies two assumptions. First, the motion is assumed to be translational. Second, objects are assumed to be larger than blocks. If the first assumption does not hold, i.e. for instance global zooming or rotation are present, a block-based approach will suffer compared to for instance a global motion estimation [27]. But such a global motion estimation is not able to model independent motion. Hence, we focus on the well established block-based motion estimation, also used in state of the art codecs like H.264 [102].

If the second assumption does not hold, i.e. the block size is not suited to the content, the compression performance suffers. To counter such degradations, it is possible to work with dynamic block sizes. As such it is possible to combine the strengths of large and small blocks. Large blocks are beneficial in homogeneous areas for higher robustness against noise. Small blocks are beneficial in dynamic, detailed areas. So for dynamic block sizes, the estimation process starts on larger blocks. The block size is only reduced when the spatial accuracy must be high [51].

Looking at other block-based motion estimators from literature, there is another field which does not require access to the reference frame. This field is frame rate up-conversion [41, 65, 35, 36, 77, 39]. This work reports, that for the purpose of predicting Y from neighboring frames, "true" motion is beneficial [69, 31, 37]. True motion entails a consistent and smooth motion field without outliers. So, next to the two block-based assumptions, a new assumption is made. The assumption that objects have inertia then implies that the movement of objects varies gradually from frame to frame.

The candidate we propose for motion estimation in DVC is such a true motion estimator, the well known 3DRS (3-D Recursive Search) algorithm [41]. The 3DRS algorithm is an iterative process, i.e. the motion vectors from the previous frame are used as an initialization for the current frame. The 3DRS algorithm works as follows. It constructs a small set of candidate motion vectors. Next to spatial and temporal candidates, also update candidates are added to the candidate set. An update candidate is computed by taking a spatial candidate and adding a small random vector update to it. The method yields a smooth and consistent vector field. To also tackle dynamic block sizes, we consider CARS (Content-Adaptive Recursive Search) as an extension of 3DRS [76]. CARS provides content adaptivity by varying the block size.

Related to the question of how to estimate the motion is the problem of evaluating prediction quality. The prediction in DVC needs to give the LDPC decoder the best objective estimate of X, to be able to recover the correct pixel values. For that reason we consider the PSNR a suitable quality measure in the context of DVC. The PSNR is a well known objective quality measure, which uses the MSE to compute the difference between frames. In DVC these frames are the reference frame X and the side information Y. The main drawback in the DVC context is that the PSNR is computed globally, whereas local information is beneficial for the decoding.

In the following we move on to the actual side information generation schemes we consider. For these schemes we discuss the algorithms involved. The final selection of schemes is then evaluated in terms of prediction quality, and finally from the RD perspective.

4.2. Generating side information

In Section 4.1 we briefly introduced extrapolation and interpolation. We outlined the importance of the GOP-size, which for interpolation has a large impact on both the prediction quality of the side information and the required number of key frames. For the extrapolation the GOP-size is only important for error recovery and random access.

In Section 4.1 we report that the prediction quality of extrapolation is inferior to interpolation with a small GOP-size. With a small GOP-size, interpolation has access to accurate information from both, past and future frames. Section 4.1.1 showed that in terms of entropy, the interpolated WZ frames outperform their extrapolated counterparts. In extrapolation, we only have past frames available. Consequently, events like occlusion are difficult to handle.

For occlusion, i.e. a foreground object moving on top of another object or the background, there are two specific problems. First, the foreground needs to be correctly detected. Second, uncovering background areas may reveal previously hidden objects. To better deal with the two problems we propose an approach, using three instead of only two frames during the motion estimation [24]. In the following sections, this scheme will be denoted as **MX** for Motion(-compensated) Extrapolation. Motion(-compensated) Interpolation will be denoted as **MI**.

To investigate the relation to conventional video coding, we also consider an extrapolation scheme with access to the reference frame X. It should be noted, that this scheme is not practical in DVC and its results are provided both as an upper bound and as a comparison to motion compensation in conventional predictive coding. To emphasize the non-practicality, it will be denoted as **MO** for Motion Oracle.

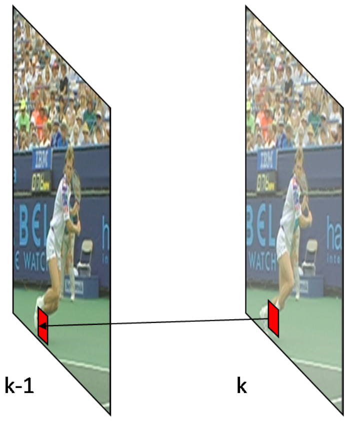


Figure 4.2.: Conceptual prediction of frame k by MO.

All three schemes use the same motion estimation, the CARS algorithm. The first main difference is where the motion estimation is applied, i.e. between which frames the motion is estimated. For the extrapolation, the motion is estimated between previous frames as shown in Figure 4.1 (a). For the interpolation, the motion is estimated between the closest previous and following frame as shown in Figure 4.1 (b). The oracle estimates the motion between the previous and the current (reference) frame as shown in Figure 4.2.

The second main difference is the consistency between motion estimation and compensation. For the extrapolation, the motion is estimated for the previous frame, but needs to be applied to the current one. For the interpolation, motion is applied to both frames it was estimated from. For the oracle, the motion is estimated for the current frame and can be applied directly.

Due to the smaller temporal distance, i.e. estimating the motion between directly adjacent frames, the motion estimation for extrapolation and oracle are superior in terms of accuracy. But only the latter can apply that information directly. During the compensation, the interpolation is the only scheme with access to the following frame. Consequently, it is best suited to deal with the occlusion problem. We consider the PSNR of the non-quantized side information Y, based on the non-quantized reference frame X. Based on the scheme properties we expect the following PSNR performance.

Sequence with little motion The motion estimation is easy and the temporal distance only secondary. Here, we expect interpolation to benefit from its compensation advantage to outperform extrapolation and possibly also the oracle. Further, the oracle should only marginally outperform extrapolation.

Sequence with large motion The motion estimation is difficult and changes significantly over frames. Here, we expect the oracle to greatly outperform both other schemes. Extrapolation faces the problem, that the estimated motion for the previous frame is not a good match for the current frame. Interpolation has to estimate the motion over at least two frame, but can compensate for some of the resulting errors by motion compensated frame averaging. Consequently, we expect interpolation to outperform extrapolation.

4.3. Proposed extrapolation algorithm and derivatives

In this Section we elaborate on the algorithms we use for the three side information schemes. While the motion estimation algorithm is similar for all three schemes, the compensation differs significantly. Again, the main focus is on the proposed three-frames extrapolation scheme, which will be introduced in detail.

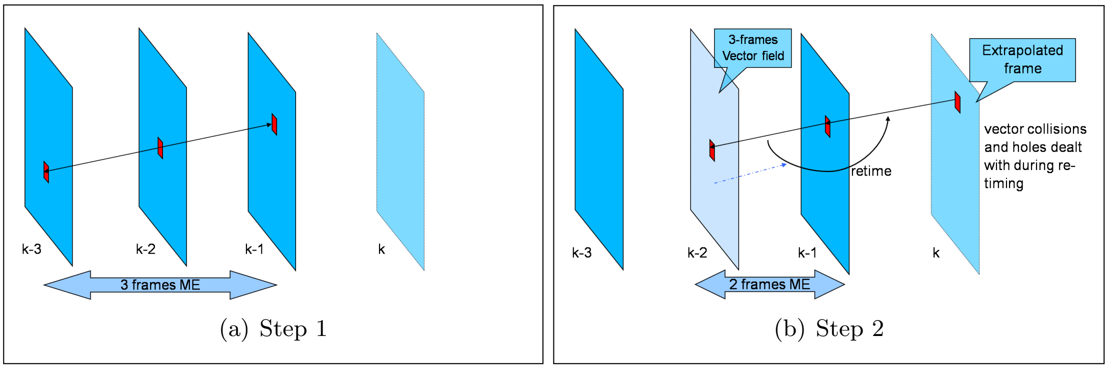
4.3.1. Three-frames extrapolation

In [24] we investigated some measures to improve the prediction quality of the three-frames extrapolation. For clarity, we present the final, best performing algorithm.

The primary purpose of using 3 frames is to help handle occlusion. After motion estimation, the two occlusion problems affect the vectors as follows. First, in front of a moving object motion vectors will collide, as background and foreground vectors point to the same area. Second, uncovering areas, which may reveal previously hidden objects, can not be known beforehand in case of extrapolation. These areas, called holes, are unreferenced by any motion vectors.

On of the advantages of the three-frames motion estimation is a better initial estimate for the subsequent two-frames motion estimation. Figure 4.3 illustrates the two ME steps. In step 1 we estimate the motion between the three previous frames. The resulting motion vectors are used as an improved initialization for the two-frames ME in step 2.

The other advantage of the three-frames motion estimation is an improved motion compensated extrapolation of the motion vector field, further referred to as vector retiming. The estimated motion vectors of the two-frames ME are only valid between the two previous frames. To extrapolate the current frame, the vectors should be valid between the previous and the current frame. Hence, the vectors have to be extrapolated along the motion trajectory, as indicated in Figure 4.3



(a) Step 1

(b) Step 2

Figure 4.3.: Extrapolating with 3 frames at the decoder.

Vector collisions have to be handled during the retiming. Very simple approaches, i.e. the last colliding vector is treated as foreground, suffer from significant degradation if the last colliding vector belongs to the background. The three-frames vector field provides a more sophisticated solution. It is based on vector consistency. We assume, that the foreground vector is the most consistent vector over time, i.e. the one that changes least over the three previous frames. To find this vector, we perform a retiming step for the three-frames vector field.

During the retiming of the three-frames vector field from $n-2$ to $n-1$, vector collisions occur. The two-frames vector field is also valid for $n-1$. We then compare the colliding vectors from the three-frames vector field with co-located vector from the two-frames vector field. The best match is labeled as foreground vector. If a subsequent vector is a better match than previous one(s), the region it points to is labeled as foreground.

As this approach is based on the quality of the vector field, it becomes more reliable the better the vectors are. One of the problems in this context is the quality of the vector field at object boundaries and edges. To get a smoother vector field without loosing information around the edges, we use cross bilateral filtering [89].

The second occlusion problem, uncovering areas, also profits from a higher quality of the vector field. In [24] we opted for a temporal hole filling procedure, where we assign co-located motion vectors from the previous frame to unreferenced areas. The temporal hole filling gives a closer representation of the uncovered background, than spatial hole filling [24].

The execution of the proposed scheme is shown in Figure 4.4 and performs as follows:

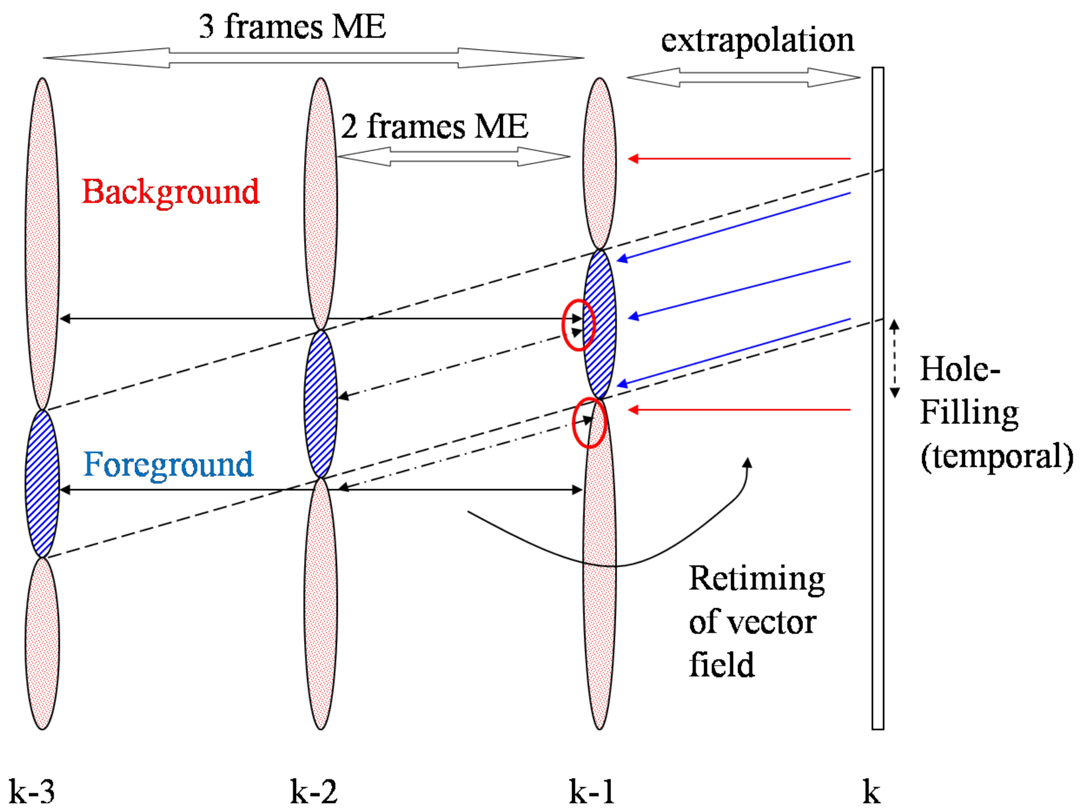


Figure 4.4.: Conceptual execution of proposed three-frames extrapolation scheme.

1. Generate three-frames motion vector field for position $k-2$. This is done by estimating the motion between frames $k-2$ and $k-3$, and between frames $k-2$ and $k-1$.

2. Generate two frames motion vector field for position n-1. This is done by estimating the motion between frames $k-2$ and $k-1$, using the three-frames vector field as initialization. Further, apply cross bilateral filtering to the vector field.

3. Find areas that are more than once addressed. Check consistency of the vectors by shifting the field generated in **1.** to n-1. Compare shifted vectors and available ones in **2.**. The vector with the lowest difference defines the foreground.

4. Retiming of the vector field, taking **3.** into account. Fill unreferenced areas with temporally previous vectors (temporal hole-filling), i.e. copy the vector at the coinciding position from vector field **2.**. Most likely, this vector follows the foreground motion and hence references the closest uncovered background.

5. Extrapolate current, motion compensated frame Y at position k.

4.3.2. Interpolation and oracle schemes

The remaining two compensation schemes are interpolation and oracle. They use the same framework as the extrapolation and their implementation is straightforward. The interpolation scheme we consider is not optimized for performance like for instance in the DISCOVER codec, but shares a general structure.

The derived Interpolation scheme performs as follows:

1. Generate two frames motion vector field for position k. This is done by estimating the motion between frames $k-1$ and $k+1$, using the vector field from the previous frame as initialization. Further, apply cross bilateral filtering to the vector field.

2. Use non-weighted motion compensated averaging between $k-1$ and $k+1$, i.e. average along the motion trajectories.

The interpolation scheme can be extended to larger GOP-sizes. The only change for using a GOP-size of 4 instead of 2 is to slightly modify step 1. For the larger GOP-size, the motion is estimated between $k-2$ and $k+2$.

The derived Oracle scheme performs as follows:

1. Generate three-frames motion vector field for position $k-1$. This is done by estimating the motion between frames $k-1$ and $k-2$, and between frames $k-1$ and k.

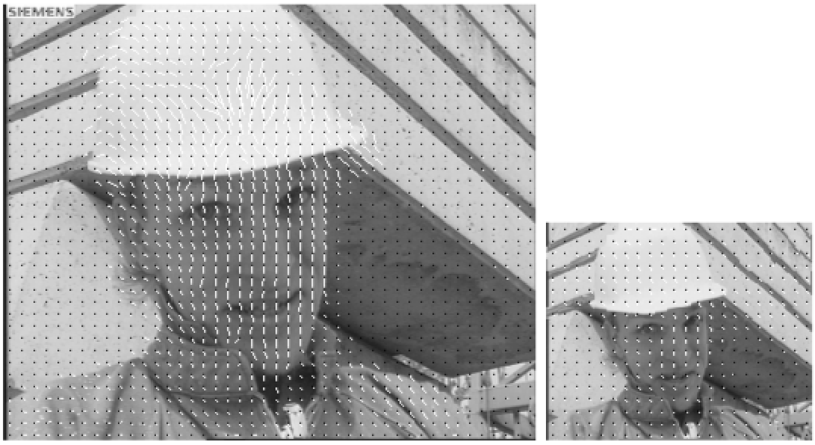
2. Generate two frames motion vector field for position k. This is done by estimating the motion between frames $k-1$ and k, using the three-frames vector field as initialization. Further, apply cross bilateral filtering to the vector field.

3. Extrapolate current, motion compensated frame Y at position k.

4.4. Evaluation of prediction quality

In this section we compare the side information quality of the three prediction schemes. For that purpose, we measure the PSNR between the side information Y and the corresponding reference frame X. To investigate the validity of our proposed prediction schemes, we compare their performance to approaches in literature. Non-quantized results in literature are only reported for QCIF resolution. Consequently, we chose this resolution for comparison purposes.

4.4.1. Comparison with literature



(a) CIF

(b) QCIF

Figure 4.5.: Frame with motion vectors from foreman for (a) CIF and (b) QCIF

However, QCIF is an atypical resolution for many applications. Since interpolation is less error prone with averaging between past and future frames, this very low resolution gives an extra penalty to extrapolation. Figure 4.5 shows the difference in scale between QCIF (176x144) and the higher CIF (352x288) resolution for both, the frame size and the motion vector size. As we focus on the PSNR of the side information, the results we provide are generated from original (non-quantized) key frames. Consequently, for instance varying settings for the key frames have no influence on the results.

From literature we consider the following approaches:

[58] We use the quarter-pixel resolution extrapolation results (MX and MO) and the bidirectional multiple references ones (MI-2).

[88] We use the results for MO, MI-2 and MX. While the results for MO and MI-2 are similar to [58], the results for MX differ.

[87] We use the results for MI-2. The main difference to [88] is that [87] uses a spatially smooth vector field as described in [31], while [88] relies on a minimum residue approach.

[17] We use the results for MI-4 to show the impact of an increase in GOP-size.

Results of the comparison are summarized in Table 4.1.

Table 4.1.: Average PSNRs [dB] of different side information schemes

sequence	proposed	[88, 58]	proposed	[58]	[88]	[88, 58]	[87]	[17]	based on proposed	
-	**MO**	MO	**MX**	MX	MX	MI-2	MI-2	MI-4	**MI-2**	**MI-4**
Carphone	**36.6**	33.80	**30.9**	29.45	29.03	31.72	-	-	**34.9**	**30.1**
Stefan	**27.6**	24.84	**26.3**	23.73	22.88	23.54	-	21.53	**27.6**	**23.2**
Foreman	**34.6**	33.21	**32.3**	31.57	30.66	32.56	32.2	28.85	**34.9**	**29.2**
Coastguard	**34.9**	32.61	**34.1**	31.32	30.82	32.17	34.2	30.59	**37.5**	**31.9**
Mother	**42.9**	-	**41.9**	-	-	-	38.1	-	**44.4**	**39.6**
News	**37.3**	-	**33.5**	-	-	-	33.0	32.8	**36.4**	**32.7**
Hall-monitor	**37.7**	-	**37.4**	-	-	-	36.7	-	**40.0**	**36.2**
Silent	**35.8**	38.11	**34.0**	34.26	33.62	36.09	-	-	**36.4**	**31.7**

MX On average our extrapolation scheme yields the best extrapolation performance. The exception of the Silent sequence will be analyzed in detail for MO. On average our extrapolation scheme comes very close to the performance of the MI-2 schemes. The decreased quality of the extrapolated Carphone sequence can be explained by the high amount of jerky global motion. It is very hard to adapt to this motion, with only past frames available and no camera stabilization.

MI-2 Compared to the status in literature, the proposed interpolation scheme shows a significantly better PSNR. While the motion estimation is similar to MX, the compensation provides a higher robustness towards errors. This difference is enhanced by the use of QCIF resolution, which is too small for the employed motion estimation. In some cases MI-2 even outperforms MO.

MI-4 If we look at a GOP size of 4 instead of 2 the quality of the interpolated side information decreases significantly. In all cases it is noticeably below the extrapolation results

MO First, the higher the difference between our proposed MX and the corresponding MO, the lower the temporal consistency of the motion in the sequence. In this case the motion from the previous frame is not a good representative for the current one. Here, interpolation approaches have the biggest advantage. An example for this is the Carphone sequence. Second, for extrapolation purposes a smooth true motion vector field, as provided by 3DRS, yields a better performance compared to a minimum residue one [58] with one exception. The Silent sequence contains a lot of sudden and fast arm/hand movement. This fast movement is hard to model with 3DRS, since they violate the assumption of objects moving gradually. For this case an approach without a smoothness-constraint is better.

Apart from the Silent sequence, which violates the smoothness assumption, schemes based on true motion estimation [87] (and proposed) outperform schemes based on minimum residue motion estimation [58, 88].

4.4.2. Choices for further consideration in this thesis

For interpolation we only consider MI-2 in the following, since increasing the GOP-size impairs the quality of Y significantly. Table 4.2 shows the decrease for a GOP-size of 4. The quality is already below that of extrapolation, which needs significantly fewer key frames. As little as only a single key frame (I-WZ-WZ-WZ-...) can be sufficient. Consequently, we only consider a GOP-size of 2 for interpolation in the following.

We consider a subset of the sequences in Table 4.1 and chose the following four test sequences with varying properties:

Table 4.2.: Avg. luminance-PSNR of Y [25] from non-quantized QCIF (30Hz) - extracted from Table 4.1

sequence	MI-2 [dB]	MX [dB]	MI-4 [dB]	MO [dB]
Stefan	27.6	26.3	23.2	27.6
Foreman	34.9	32.3	29.2	34.6
Coastguard	37.5	34.1	31.9	34.9
Hall-monitor	40.0	37.4	36.2	37.7

Table 4.3.: Avg. luminance-PSNR of Y from non-quantized CIF (30Hz) - MI-4 no longer considered

sequence	MI-2 [dB]	MX [dB]	MO [dB]
Stefan	25.5	24.5	28.6
Foreman	32.8	30.2	34.6
Coastguard	33.8	31.3	33.0
Hall-monitor	36.9	34.7	36.0

Stefan Large motion and highly detailed textures present.

Foreman Significant motion and mostly little detailed textures (changes towards end of the sequence) present.

Coastguard Noticeable motion and highly detailed textures present. The water motion in the sequence is inherently unpredictable.

Hall-Monitor Little motion present and little detail in textures.

In addition, the following experiments are carried out with CIF resolution sequences. As indicated before in Figure 4.5, both the resolution and the amount of motion increase significantly. The motion is no longer restricted to comparably few pixels as in QCIF, where many zero motion vectors are chosen. For CIF resolution the importance of the motion estimation increases. Since high resolution also implies more image detail, the prediction quality decreases for all schemes. For reference, the non-quantized prediction quality of the three schemes on CIF resolution is given in Table 4.3. The ratio between the schemes is similar to the QCIF resolution results.

4.5. Evaluation of RD performance

Allowing for a low complexity encoder, we choose H.263 intra coding for the key frames. It performs efficient intra coding, and spatial correlation in key frames is exploited very well. To enable at least limited exploitation of spatial correlation we use a straightforward DCT scheme for the WZ frames. We will elaborate on the DCT scheme in Chapter 5. We use the high motion Foreman and the low motion Hall-monitor sequences in CIF resolution with a frame rate of 30 frames per second.

Figure 4.6 shows the RD performance of **MX**, **MI** and **MO**. Since the oracle has access to the reference frame it clearly outperforms both other schemes. For the high motion Foreman sequence we observe a PSNR difference of up to 1.5 dB between the motion oracle **MO** and extrapolation **MX**. For the low motion Hall-monitor sequence, the PSNR between **MX** and **MO** differs at most by 0.2 dB.

As we have shown in Section 4.4, in terms of side information PSNR the interpolation scheme **MI** always outperforms extrapolation **MX** (and in some cases even the motion oracle **MO**). In the RD performance of the complete system however, the gain is offset by the higher key frame cost. For the Foreman sequence, **MX** outperforms **MI** by 0.5 dB for low rates and shows similar RD performance for high rates. For the Hall-monitor sequence, providing more temporal correlation to exploit, **MX** outperforms **MI** by up to 1.5 dB for low rates and 0.7 dB for high rates.

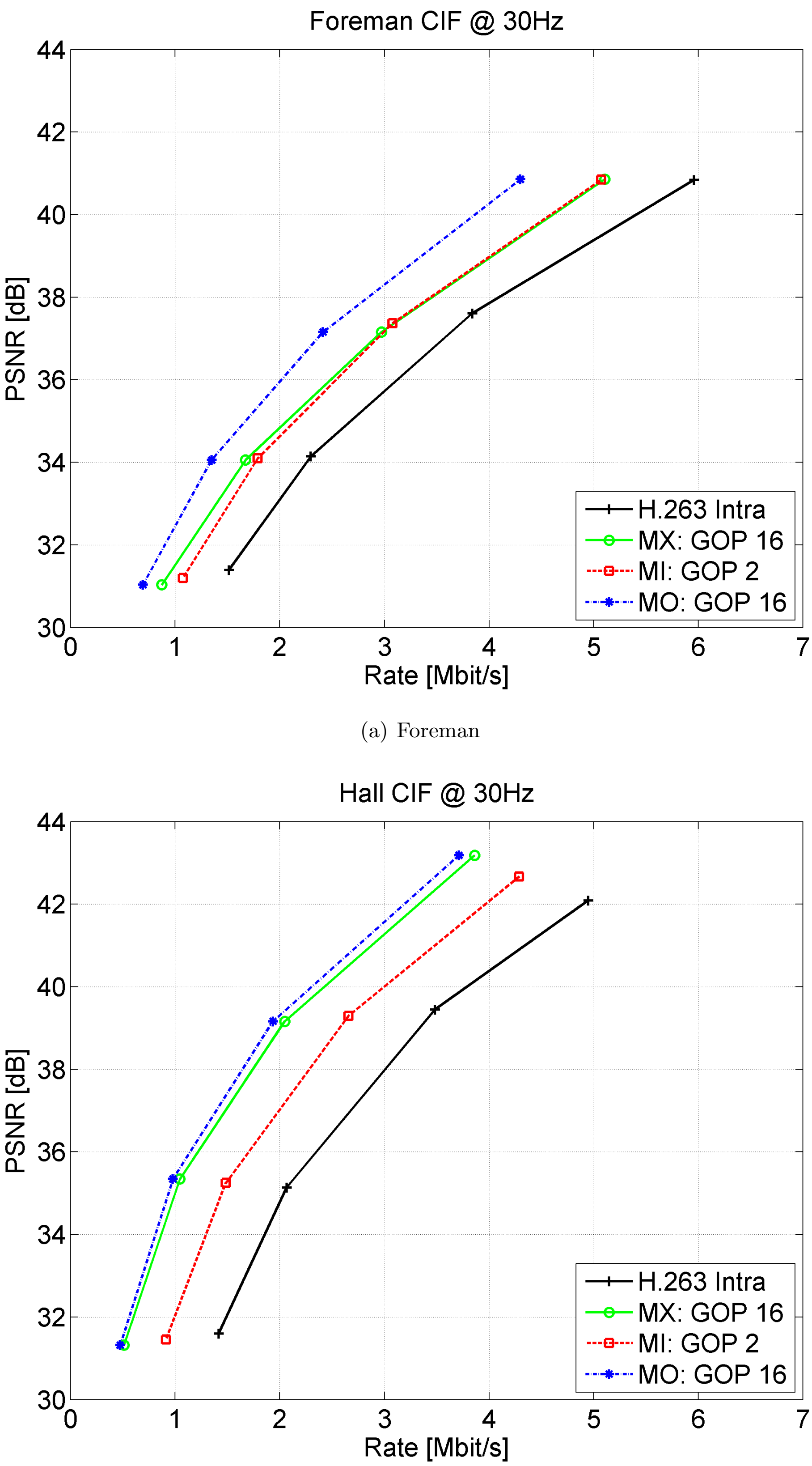
The more temporal correlation and spatial correlation can be exploited in the WZ frames, the better extrapolation becomes in comparison to interpolation. With the high temporal correlation of the Hall-monitor sequence, extrapolation clearly outperforms interpolation. Any further improvements in WZ will benefit extrapolation more, since intra coded key frames become worse in comparison.

4.6. Discussion

In this chapter we have investigated the motion estimation and compensation aspect of DVC. Our findings can be summarized as follows:

True motion estimation superior to minimum residue motion estimation in DVC
We find in Section 4.4 that with the exception of the Silent sequence, schemes estimating the true motion consistently outperform minimum residue schemes. Especially the proposed scheme with CARS and underlying 3DRS, shows an excellent performance and outperforms for instance the next best extrapolation scheme by up to 2.8 dB.

Proposed extrapolation scheme superior to interpolation
We show that extrapolation is preferred for the purpose of DVC. The initial side information quality experiments show that the PSNR of interpolated side information with a GOP-size of 2 consistently outperforms the PSNR of extrapolated side information.



(a) Foreman

(b) Hall-monitor

Figure 4.6.: Rate distortion curves of MX, MI and MO for (a) Foreman and (b) Hall-monitor.

That statement holds for interpolation schemes from literature and the interpolation scheme based on our proposed extrapolation.

We make clear that the higher side information PSNR deteriorates quickly with an increase in the GOP-size. A small GOP-size induces the need for a large number of expensive key frames. Taking the key frames into account, interpolation falls behind the system RD performance of extrapolation. To measure the RD performance we considered an entropy-based information theoretic approach and RD results from a transform domain DVC system, which will be presented in Chapter 5.

Based on these findings we use the CARS-based extrapolation scheme for the rest of this thesis. In Chapter 5 we introduce a transform domain system, of which we anticipated some results for our argumentation in Section 4.5.

5. Quantization in the transform domain

State of the art video coders exploit temporal correlation in video sequences by motion compensated prediction, and spatial correlation by adaptively quantizing DCT coefficients of intra frames or motion compensated frame differences. With the exception of the RD performance evaluation in Section 4.5 the DVC approach we discussed so far ignored spatial correlations. In this chapter we follow the work of [12], [74] and [15] and include the DCT transform in DVC, as shown in Figure 5.1. The resulting DCT coefficients need to be adaptively quantized - as in state of the art coders - to exploit the DCT transform. Quantization of DCT coefficients in a DVC setting is, however, a nontrivial challenge because of the restrictions imposed by the LDPC coding.

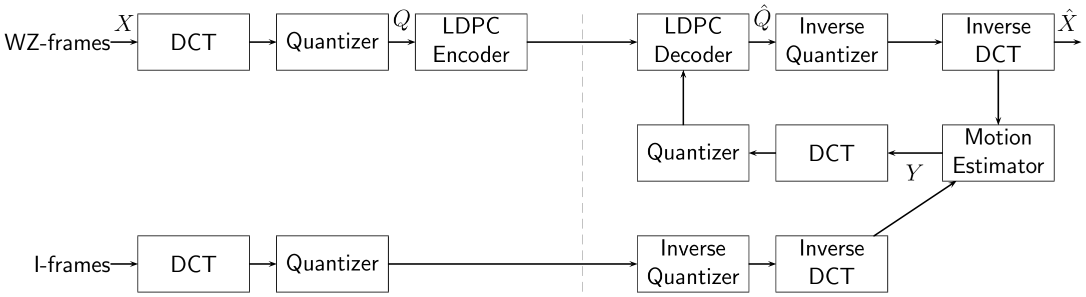


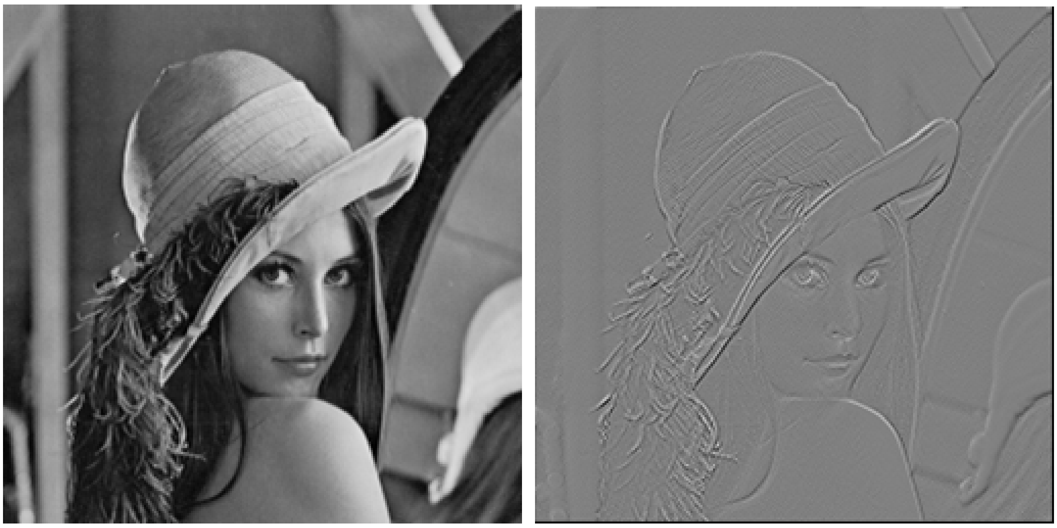
Figure 5.1.: Proposed DVC transform domain scheme [26].

First, we elaborate on these restrictions and their impact on the quantization. Second, we briefly outline quantization in state of the art predictive video coders. We then focus on the limitations present in DVC and discuss possible options to tackle the quantization. We further propose a simple side information update, exploiting the iterative decoding of DCT coefficients. Finally, we evaluate the presented quantization choices and the proposed side information update.

5.1. Background

A spatially decorrelating transform removes correlation by packing energy in as few transform coefficients as possible. We focus on the DCT, which is used in many image/video compression standards [83]. It should be noted that the DCT itself is a lossless process, i.e. the original data can be reconstructed.

The lossy step is then to quantize the coefficients. Various techniques exist for the quantization of coefficients. The best quantization depends on data properties. Here, we observe the first big difference between conventional video coders and a low complexity DVC coder. While the former works on inter frame information, the latter only has access to intra information.


(a) Lena

(b) Lena DPCM

Figure 5.2.: Different properties of (a) full Image (intra information) and (b) prediction difference (simple inter information) for Lena.

Figure 5.2 depicts an example of the difference between intra and inter data for the Lena image. The intra data contains the full scope of the original image data. In contrast, the inter data is highly peaked around zero. Consequently, in the latter many more coefficients are quantized to zero. Especially such zero coefficients are handled very efficiently in state of the art predictive video coders.

The second main difference between conventional video coders and a low complexity DVC coder is the encoding of the quantized values. Conventional coders employ efficient 2D Variable Length Codes (VLC). The coarseness of the quantization can be selected adaptively, based solely on the rate control aspect. In DVC we have to rely on fixed-length coding. Here, the LDPC encoder limits the adaptivity of the quantization.

In DVC, a fixed number of bit planes per band is addressed since for LDPC coding the number of bits in a coefficient band needs to be known at the decoder. Each quantized coefficient inside a band can be represented by a fixed number of bits, independent of the quantized value [100]. Alternatively, it is possible to use adaptive quantization and send the number of bits for a coefficient as overhead information to the decoder before decoding the DCT coefficients.

In the scope of this chapter, we focus on the different possibilities to perform the quantization in DVC, ranging from fully adaptive but not realizable to simple with significant performance degradations.

5.1.1. State of the art inter frame coders

If the amount of information conveyed by each DCT coefficient is different, it makes sense to assign a varying numbers of bits to DCT coefficients, i.e. an adaptive bit allocation. There are two approaches to allocating bits. One approach relies on the average properties of the transform coefficients, while the other approach assigns bits as needed by individual transform coefficients. In both approaches, coefficients with higher variance are assigned more bits than coefficients with smaller variance [81].

In addition, quantized DCT coefficients are dependent. Variable length coding in conventional predictive video coding is designed to exploit the dependency. Per block, pairs of non-zero values and zero runs are taken and more common (length of zero run, value) pairs are assigned small codewords. Less frequent pairs are assigned longer codewords [100].

If we scan an 8x8 block of quantized transform coefficients in a zig zag fashion [34], we will find that in general a large section of the tail end of the scan will consist of zeros, especially for quantized inter data. In combination with mid tread quantizers and higher step sizes for the higher-order coefficients generally chosen to be quite large, many of the inter frame DCT coefficients will be quantized to zero. Therefore, there is a high probability that after a few coefficients along the zig zag scan, all coefficients will be zero. In this situation, Chen and Prat [34] established the transmission of a special *End-Of-Block* (EOB) symbol [81].

For instance in H.263, after quantization and zig zag scanning, 2D-Huffman coding is applied to send combinations of length of zero-runs and non-zero DCT amplitude. Consequently, in the bit stream we find the following: *(length of zero run, value), (length of zero run, value), ... (length of zero run, value), EOB.* This representation provides a very efficient compression of the long zero runs in inter frame coders.

5.1.2. DVC coders

In DVC coding the dependencies between DCT coefficients inside a block are not exploited. Grouping coefficients into bands and coding bands separately eliminates the dependency structure between the DCT coefficients. In addition, the intra frame DCT coefficients exhibit significantly fewer zero runs and a higher number of bits per DCT coefficient. When the DCT coefficients are grouped together in bands, the number of bits per DCT coefficient varies. Moreover, the number of bits varies temporally.

Fixing the number of bits can have adverse effects. If the number of bits is insufficient the quantized value it will be clipped. The quality of the reconstructed data will suffer accordingly. If more bits than required are assigned, the LDPC decoder will have to deal with the noise introduced by the additional bit planes. The LDPC decoder then has to match zero bit planes in the original quantized frame with possibly non-zero bit planes in the side information.

It should be noted that the VDC channel model properties of bit planes of DCT coefficients are different from pixel-based symbols, for which we identified the two-sided Gamma distribution as best pixel-based distribution in Chapter 3. For DCT coefficients, the zero-mean Laplacian distribution is the most widely accepted model in the literature for the temporal correlation channel [42]. The results in [79] show that the DCT coefficients tend to be more Laplacian than Gamma distributed. In traditional video coding, the Laplacian distribution is typically used to model the distribution of the motion compensated residual DCT coefficients [20].

More accurate models can be found in literature, such as the generalized Gaussian distribution; however, the Laplacian distribution constitutes a good trade-off between model accuracy and complexity and, therefore, it is often chosen, e.g., [56]. For the same reason, the Laplacian distribution is widely used to model the transform domain VDC in the DVC literature, e.g., [47], [12], and [18], and, therefore, following the argumentation in [30], our transform domain codec uses a Laplacian distribution.

Most approaches in literature fix the number of bits per DCT coefficient at the encoder [15]. An alternative to fixing the number of bits at the encoder would be to adapt the number of bits per coefficient to the content and send this information as overhead next to the DCT coefficients. Consequently, there would be neither clipping nor noise only coefficients. At the same time, this method incurs an overhead for the number of bits per coefficient. In the following we will investigate some practical methods ranging from simple fixed quantization to VLC-like adaptive quantization.

5.2. Considered quantization schemes for DVC

In this thesis we use a block size of 8x8, opposed to the often used 4x4 in literature [15]. As we focus on CIF resolution, less energy is left in the higher coefficients of an 8x8 transform, so that overall fewer information needs to be coded. Consequently, in [100] the observation is made that an 8x8 transform is better suited for CIF resolution than a 4x4 DCT transform.

For the quantization of the DCT coefficients we consider four schemes. The schemes exhibit an increasing level of adaptivity at the cost of transmitting additional overhead, i.e. information about the number of bits per DCT coefficient. We are then interested in the trade-off between the cost for sending the quantized DCT coefficient and overhead information. For comparison purposes we also include an oracle quantization, which exhibits full adaptivity without any overhead. The schemes we consider are:

set globally In this scheme, the quantization is completely fixed. For each chosen quality level, the number of bits per coefficient is set globally, regardless of sequence properties. If a DCT coefficient requires more bits than set, it will be clipped. It should be noted that this scheme was used for the RD performance experiment in Section 4.5.

fixed frame This scheme fixes the bits per coefficients for each video frame. The highest bit value for each of the 64 coefficients across all blocks in the video frame is calculated and sent as small overhead to the decoder. Consequently, unlike the **set globally** scheme, the quantization is able to adapt to changes between frames and sequences. Changes within a frame are not taken into account.

adaptive This scheme allows an arbitrary number of bits per coefficient. As such, the scheme is able to adapt to changes between and within frames. A consequence of the adaptive quantization is that zero bit planes are deterministically fixed at the decoder, i.e. do not need to be decoded and incur no rate. Thus, the scheme relies on the decoder already knowing how many bits belong to each coefficient. It should be noted that this scheme corresponds to the fully adaptive quantization in conventional predictive coders. Consequently, its purpose here is as a reference only as it can not be implemented in DVC.

adaptive+overhead This scheme is similar to **adaptive**. But to make the **adaptive** scheme practical, overhead information needs to provide the number of bits per coefficient separately. For that purpose we use VLC coding for the overhead information. For each coefficient band we take the residue DCT coefficients between every two spatially neighboring DCT blocks. The residue DCT coefficients are then arranged in (length of zero run, value) pairs. It should be noted, that we employ a very simple scheme and store the length of zero runs and the values in separate arrays. For each of the arrays we calculate the best Huffman dictionary. We then apply the corresponding dictionary to each of the arrays. The overhead rate is then the sum of the two array costs.

The actual quantization implementation we use varies slightly over the schemes. The **adaptive** and **set globally** schemes are based on MPEG-2 intra quantization. The DCT coefficients are quantized using the MPEG-2 intra quantization matrix with various scaling factors (2,1,0.5 and 0.25). For the last scheme (**set globally**), introduced in [26], clipping of coefficients is possible. We use experimentally chosen allocation tables. The allocation for the lowest quality setting Q1 and the highest one Q4 can be seen in Table 5.1 and Table 5.2.

Table 5.1.: Bits spent for each coefficient at the lowest quality setting

7	7	5	4	4	3	0	0
7	5	4	3	0	0	0	0
5	4	3	3	0	0	0	0
4	3	3	0	0	0	0	0
4	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 5.2.: Bits spent for each coefficient at the highest quality setting

10	10	8	7	7	6	5	5
10	8	7	6	5	4	3	0
8	7	6	6	5	4	3	0
7	6	6	5	5	4	3	0
7	5	5	5	5	4	3	0
6	4	4	4	4	3	3	0
5	3	3	3	3	3	0	0
5	0	0	0	0	0	0	0

The quantization for the **fixed frame** scheme is slightly different as it is derived from H.263. It yields almost similar results to MPEG-2 quantization. The H.263 intra quantization does not rely on a quantization matrix. The first coefficient band is uniformly quantized with a step size of 8. Each of the other bands uses quantizers with equally spaced reconstruction levels with a central dead-zone around zero. Their step size is 2 times the Quantization Parameter (QP), where QP is the quality factor [100].

To avoid over- or underestimation of the number of bits, the maximum quantization index is estimated per coefficient band. The maximum number of bits per coefficient band is calculated from the maximum quantization index and transmitted to the decoder [100]. There are 64 coefficients bands due to the block size of 8x8. Consequently, the overhead is limited to 64 values for each frame.

5.3. Motion estimation on quantized data

The investigation of the prediction quality in Chapter 4 was based on non-quantized video content. Here, the motion estimation has to deal with a noisy version of X since both, WZ- and I-frames are quantized. The quantization errors in $\hat{X}$ also affect the quality of Y.

5.3.1. Clipping as effect of fixed number of bits

One of the considered quantization schemes also includes clipping of quantized DCT coefficients (**set globally**). As the clipping is done during quantization, the decoded quality of reconstructed frame $\hat{X}$ suffers. Content is irrevocably lost. For the lowest quality setting, we observe the difference between a non-clipped and a clipped reconstruction in Figure 5.3. The most notable difference is the 'Siemens' logo, which is comparably well preserved without clipping. The same holds true for the sharp background edges.

The coarser the quantization, the less edges are preserved. In addition quantization introduces noise. The quantization noise may induce additional motion artifacts between consecutive frames as the motion estimation is prone to quantization errors. We focus on both the impact of quantization on the ME (for **fixed frame** and **adaptive**) and the additional impact of possible clipping (for **set globally**).

Table 5.3 shows the respective PSNRs for **set globally** with clipping and **adaptive** without clipping for each quality level. Both schemes are quantized using the MPEG-2 intra quantization matrix with the four quality levels Q1-Q4, introduced in Section 5.1.2. The **fixed frame** scheme is not listed as it employs H.263 quantization with different quality levels. With H.263 quantization the three RD points we consider are the H.263 quantization parameters $QP = 16$, $QP = 8$ and $QP = 4$. The respective reconstructed qualities are $31.5dB$, $35.2dB$ and $39.5dB$. However, the prediction quality of **fixed frame** is similar to **adaptive**.

(a) Low Quality clipping

(b) Low Quality no clipping

Figure 5.3.: Reconstructed quality of frame 100 of the Foreman sequence for (a) clipping and (b) no clipping. Main difference visible in Siemens logo.

The impact of the quantization noise can clearly be seen at the lower qualities in Table 5.3. The motion oracle loses more than 3dB in prediction quality at the lowest quality. The degradations in the motion oracle depend solely on the quality of the motion estimation. For extrapolation, a large part of the loss is incurred by the motion not being estimated for the current frame and consequently the quantization errors have less impact on the prediction quality.

Table 5.3.: Avg. luminance-PSNR of Y in [dB] (against $\hat{X}$) at respective quantization levels (Q1-Q4) for Foreman CIF at 30Hz.

scheme	Q1	Q2	Q3	Q4	non-quantized
Reconstruction quality **set globally**	31.0	34.1	37.1	40.9	∞
Reconstruction quality **adaptive**	32.0	35.3	39.1	43.5	∞
MX **adaptive**	28.9	29.5	30.0	30.2	30.2
MX **set globally**	28.1	29.0	29.6	30.0	30.2
MO **adaptive**	31.9	32.8	33.6	34.2	34.3
MO **set globally**	31.0	32.1	33.1	34.0	34.3

The prediction qualities in Table 5.3 are based on the difference between $\hat{X}$ and Y, i.e. the errors relevant for the LDPC decoder. $\hat{X}$ for **set globally** is practically a low pass filtered version of $\hat{X}$ for **adaptive**, as especially high frequency DCT coefficients are clipped in Table 5.1 and Table 5.2. Especially for highly textured background or sharp edges such low pass behavior can increase the PSNR of the side information.

The quality of the reconstruction itself however suffers, as shown in Figure 5.3. The PSNR difference between MX with **set globally** and MX with **adaptive** is caused by sharp edges, i.e. high frequency components. As the LDPC decoder corrects the errors between quantized prediction $\hat{Y}$ and quantized reference frame $\hat{X}$, we expect the PSNR difference between clipping in MX **set globally** and no clipping in MX **adaptive** not to translate to the decoding performance.

5.3.2. Side information update after partial decoding

Following the work in [93], we consider how to make use of partial reference frame information at the decoder. Since the DCT coefficients are decoded in a zig zag fashion, more and more reference information becomes available the further the decoding proceeds. It is then possible to use such a partially decoded frame to generate a better side information by estimating the motion based on the low frequency coefficients, i.e. a low pass filtered version of the reference frame.

An important design choice in this context is the amount of coefficients we decode before re-estimating the motion and generating new side information. Decoding more coefficients before updating the side information increases the quality of the side information update. Yet at the same time, there are less coefficients that can benefit from the improved side information quality. For this reason, we only consider a single update. In addition, to keep the horizontal and vertical frequency components balanced, we consider only complete zig zag lines for the side information update.

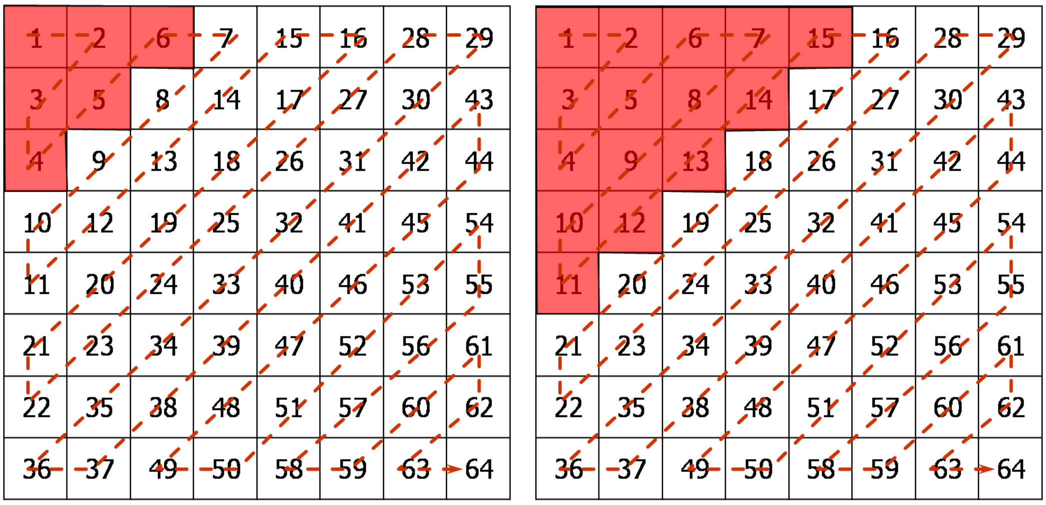
In Table 5.3 we observed how quantization itself affects the prediction quality. The coarser the quantization, the less reliable the motion oracle became. Consequently, motion estimation fails when applying it to the reconstruction of only very few low frequency coefficients. For the **fixed frame** H.263 quantization, Table 5.4 shows how the number of decoded DCT coefficients, i.e. zig zag lines, affects the quality of the side information update **MXupdate**.

Further Table 5.4 investigates the trade-off between decoding fewer or more DCT coefficients before updating the side information. The number of decoded zig zag lines affects both the PSNR of **MXupdate** and the RD performance of the system. While additional zig zag lines increase quality of the side information update, at a certain point the RD performance starts decreasing.

For each QP we highlight the number of zig zag lines we use. It should be noted that

Table 5.4.: Trade-off between side information quality and number of decoded coefficients for **MXupdate** for Foreman CIF resolution at 30Hz

QP	**# zig zag lines**	PSNR MX	**PSNR MXupdate**	$\hat{X}$ [dB]	**Rate [Mbit/s]**
16	2	28.71	29.32	31.45	0.95
16	**3**	**28.71**	**30.67**	**31.45**	**0.92**
16	4	28.71	31.23	31.45	0.92
16	5	28.71	31.46	31.45	0.93
8	2	29.48	29.63	35.24	1.93
8	3	29.48	31.52	35.24	1.83
8	**4**	**29.48**	**32.32**	**35.24**	**1.81**
8	5	29.48	32.63	35.24	1.83
4	3	30.0	32.17	39.48	3.43
4	4	30.0	33.15	39.48	3.38
4	**5**	**30.0**	**33.56**	**39.48**	**3.4**
4	6	30.0	33.74	39.48	3.44



(a) Low Quality

(b) High Quality

Figure 5.4.: Coefficients used for side information update for (a) QP=16 and (b) QP=4.

while for QP=16 and QP=8 our choice is optimal, for QP=4 we use an additional zig zag line from the optimum. The optimum varies over sequences, depending on for instance the amount of detailed texture. Here, we prefer to accept a minimal rate loss (compared to the optimal number of zig zag lines) in some sequences rather than risking a significant rate loss because of an insufficient quality of the side information update.

For the lowest quality we update the side information after the first three zig zag lines. Hence, 6 coefficients are available from the reference frame. For each higher quality we add one more zig zag line before we update the side information. An example for low and high quality is shown in Figure 5.4. In this configuration we try to preclude the possibility of the side information update decreasing the side information quality.

5.4. Evaluation of the quantization schemes

5.4.1. Quantization without side information update

For the quantization experiments we use the CIF resolution Foreman sequence. The RD curves include all frames for a frame rate of 30 frames per second. The side information is generated using the extrapolation scheme proposed in Chapter 4 (*MX*).

The first observation in Figure 5.5 is the performance of the fully **adaptive** scheme. Since only non-zero bit planes incur transmission rate, this oracle scheme outperforms all fixed schemes by up to 9 dB. However, the **adaptive** scheme relies on the decoder knowing how many bits belong to each coefficient. In practice, the encoder would have to transmit this information as overhead. Overhead information incurs an additional rate cost. The additional rate decreases the performance of **adaptive** by 6 to 9 dB. Hence, the VLC coded overhead information (**adaptive+overhead**), increases the rate by almost 200% in Figure 5.5.

Moving to the less adaptivity, the second observation in Figure 5.5 is the performance difference between limited adaptivity with small overhead (**fixed frame**) and no adaptivity without any overhead (**set globally**). In the former, the number of bits per coefficient is set to predefined values for each quality. The latter fixes the number of bits per coefficient for each frame and sends a relatively small overhead of 64 values per frame, each requiring at most 8 bits. At higher qualities the overhead of 15 kbit/s is negligible.

There are two reasons for the difference between **fixed frame** and **set globally**. First, the reconstruction quality increases since there is no clipping of higher frequency coefficients. Second, the adaptivity on frame level yields fewer coefficients that get too many bits assigned, which incurs a rate penalty for decoding noise. As expected, the PSNR difference in side information between clipping with **set globally** and no

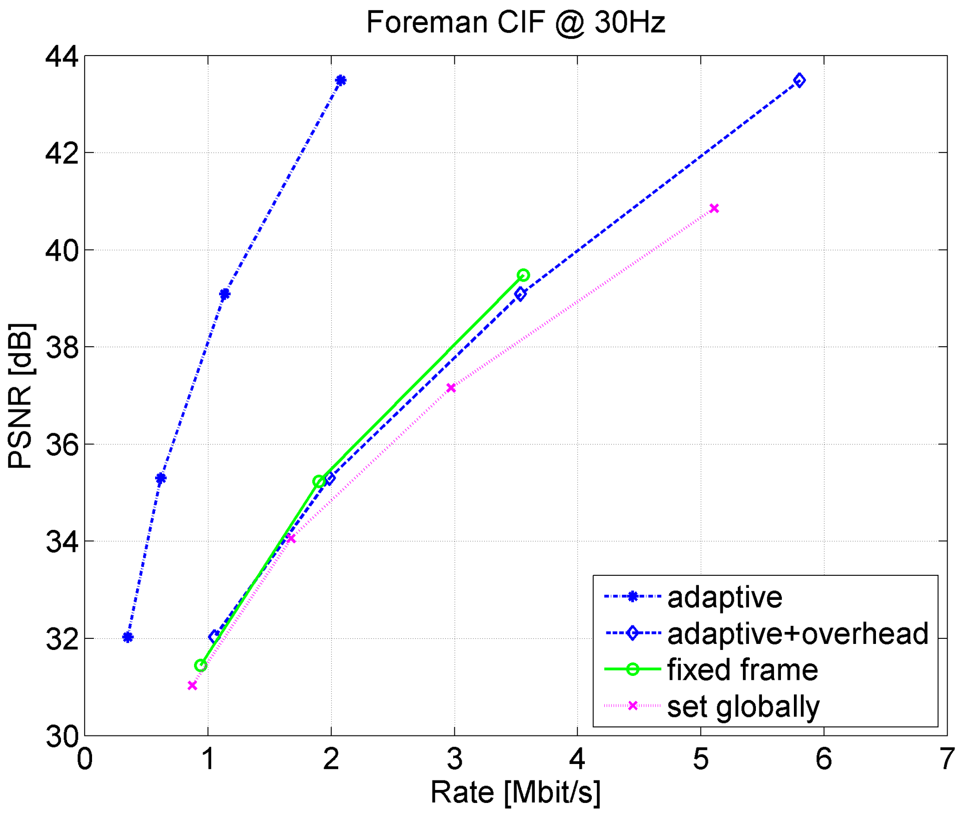


Figure 5.5.: Comparison of proposed DCT quantization schemes for the Foreman sequence.

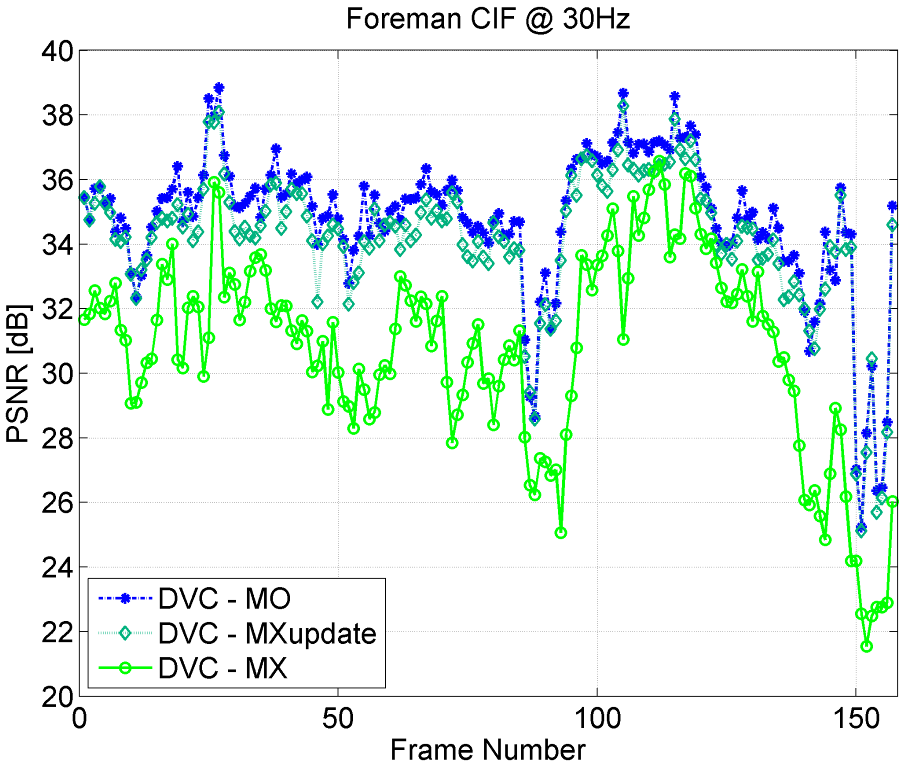
clipping with **fixed frame** does not translate to the system RD performance.

We conclude that among the alternatives presented in Section 5.2 limited adaptivity with small overhead **fixed frame** is the best performing practically applicable quantization scheme.

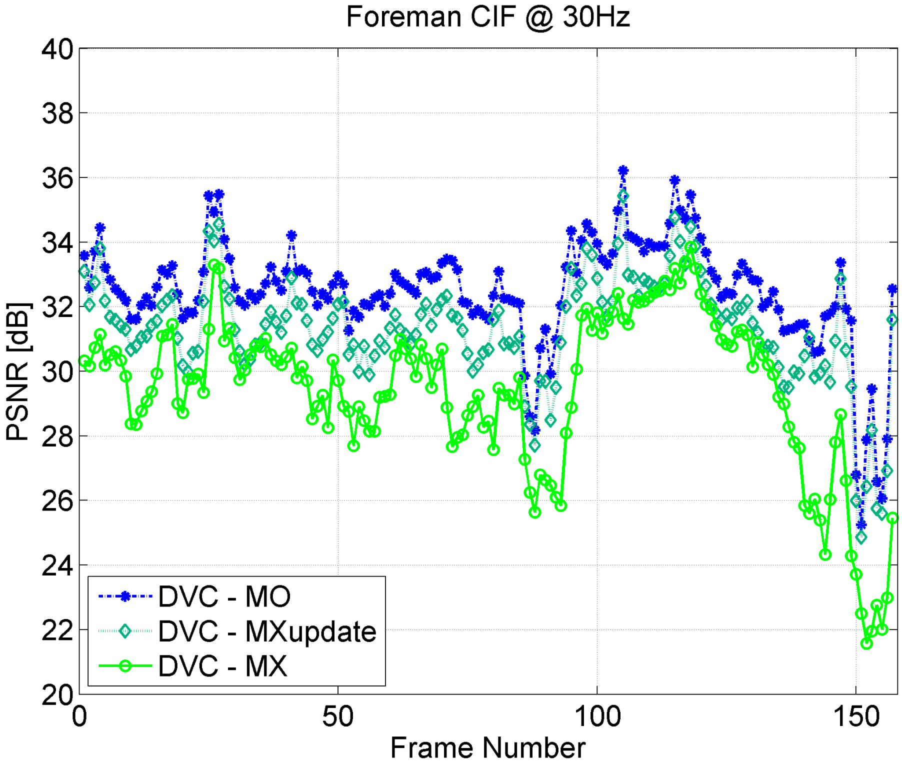
5.4.2. Evaluation of side information update

For the investigation in this section we use the best performing quantization scheme from Section 5.4: **fixed frame**.

We use the Foreman sequence up to Frame 160, to include the largest hand motion present in the sequence. Such large motion, as well as a change in motion direction, are difficult to predict for extrapolation. We compare the prediction quality of the normal extrapolation **MX**, the side information update **MXupdate** and the motion oracle



(a) $QP = 4$



(b) $QP = 16$

Figure 5.6.: Prediction quality comparison for MX, MXupdate and MO at (a) $QP = 4$ and (b) $QP = 16$.

MO in Figure 5.6.

Especially for hard to predict frames, we see a large increase in PSNR performance from extrapolation to the motion oracle. The side information update is quite close to the performance of the motion oracle **MO** for high qualities. With an increase in quality, the side information for the update converges to the motion oracle one. But even at the lowest quality, using only very few decoded coefficients, the side information update **MXupdate** yields a side information quality up to 6 dB higher than for extrapolation **MX**.

Although the side information update quality is far superior, the results in Figure 5.7 show a limited impact on the RD performance. The side information update **MXupdate** can only be applied after partial decoding, but most of the bits are spent on the low frequency DCT coefficients. Hence, their influence on the RD performance is significant. In comparison, the higher frequency coefficients left to benefit from a side information update do not contribute much to the RD performance.

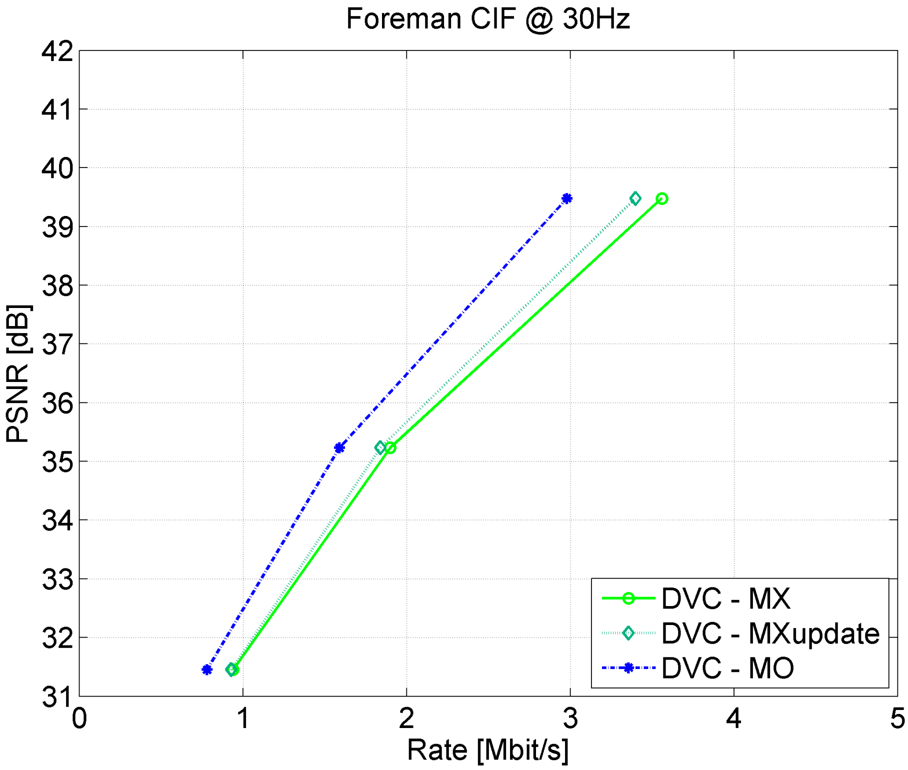
We conclude that while a side information update can improve the RD performance, especially for difficult frames with a high bit rate, in total the update only covers a small part of the gap between extrapolation and motion oracle.

5.5. Discussion

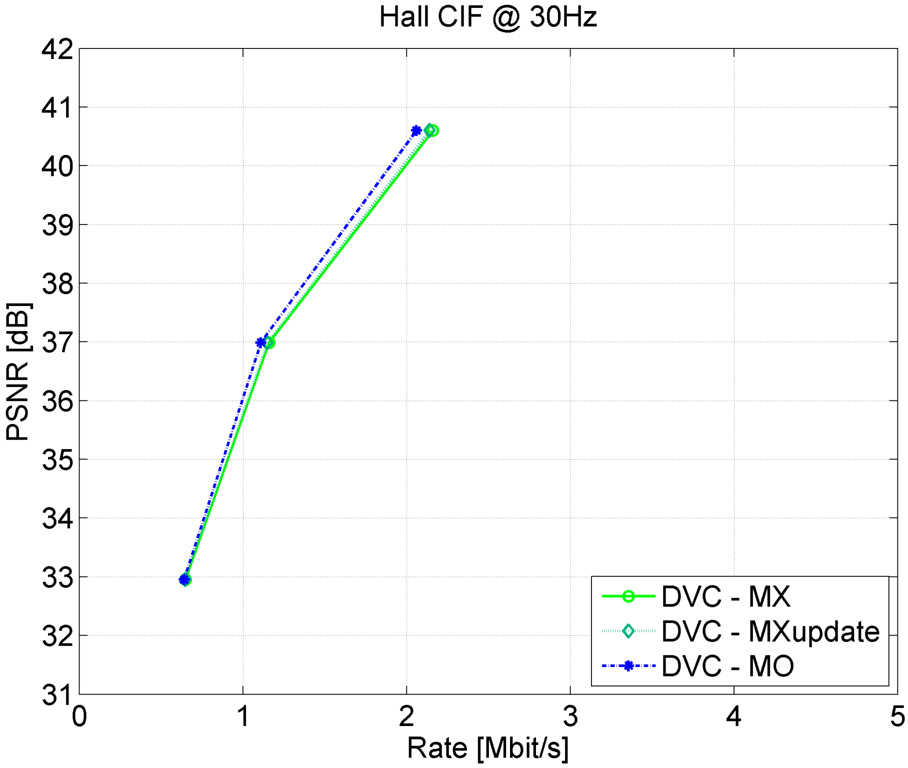
In this chapter we investigated our transform domain implementation of DVC. The main focus in this context was on the quantization. Our findings can be summarized as follows:

Quantization needs a trade-off between adaptivity and overhead
We show that the bit rate for each coefficient should be defined as adaptive as possible. The adaptivity is limited by the requirements of the LDPC decoder. For successful decoding, the decoder needs to know the number of bits per coefficient. The decoder needs to know this information in advance. So there is a compromise between the overhead for this information and the RD gain during decoding. We find the best compromise in the **fixed frame** scheme, which is adaptive up to the frame level, with only a small overhead.

ME suffers from quantization noise
The best performing quantization scheme does not suffer from reconstruction artifacts from clipping. The reconstructed quality is PSNR wise similar to that of H.263 or MPEG-2 intra coding. The motion estimation is applied to such reconstructed frames and the associated quantization noise. The extrapolation suffers less from the quantization noise than the motion oracle which is closer to minimum residue ME. The



(a) Foreman



(b) Hall-monitor

Figure 5.7.: Performance gain given by MXupdate.

errors in the extrapolation are larger to begin with, i.e. there is not as much to lose by adding quantization noise.

Propose a simple yet efficient side information update
We propose a simple scheme, which uses partially decoded data to update the side information. It is important to decode a sufficient number of coefficients first. Otherwise, the updated prediction might be of lower quality than the original one. If too many coefficients are decoded, the updated prediction will have a quality similar to the motion oracle. However, most of the coefficients have already been decoded and the impact of the improved prediction on the RD performance is negligible. With our proposed compromise we achieve a noticeable RD performance increase for higher qualities.

Based on these findings we use the **fixed frame** quantization scheme for the rest of this thesis. With regard to the side information update scheme **MXupdate**, the gain is limited to hard to predict frames/sequences. As such we only refer to it for comparison purposes. It should be noted however, that the scheme is easy to implement and does not increase the encoder complexity.

6. Non-stationary VDC modeling

State of the art video coders take the varying statistical image properties over blocks into account. As a consequence, each block type is coded differently, taking the non-stationarity of prediction errors into account. Blocks are classified as either intra, inter or skip blocks. The DVC approach we discussed so far ignored the non-stationarity of the VDC. Modeling the non-stationary VDC accurately is a significant challenge in DVC, as neither the encoder nor the decoder have access to the reference frame X and side information Y simultaneously.

First, we elaborate on how conventional coders deal with the issue of non-stationarity. Second, for low complexity DVC, we focus on decoder-based solutions. In this context, the main question is how to classify the VDC. After giving a short summary of possible improvements with manual classification, we consider such an oracle classification for transform domain DVC. We then focus on mask-based approaches for region classification. We discuss how to acquire helpful information from the motion estimation and the Motion Learning (ML) scheme introduced by Varodayan *et al.* in [93].

6.1. Background

When operating an H.264 encoder, a key step is selecting the most suitable prediction mode. A wrong selection means the encoder can not achieve maximum compression. Representative H.264 encoders select the mode that minimizes the rate-distortion cost function based on Lagrange's method of undetermined multipliers [19].

When using the intra mode, each block is predicted from spatially neighboring samples only. In addition to the intra macroblock coding types, various predictive or motion-compensated coding types can be specified as inter macroblock types. An inter macroblock can also be coded in the so-called skip mode. The skip mode is very efficiently coded since no quantized prediction error signal is transmitted [102]. Only the macroblock header and possibly the motion vectors need to be coded.

Which mode is used depends on the RD cost, calculated at the encoder itself. In low complexity DVC, we do not have access to the side information Y at the encoder. As outlined in previous chapters, the DVC encoder only performs intra coding. Without access to inter information the DVC encoder can provide no information about the VDC.

Consequently, we have to estimate the non-stationarity at the decoder. In Chapter 3 we focused on the best global channel model, ignoring the non-stationarity. Since state of the art channel codes require an accurate channel model for high coding efficiency, in the following we consider two (or more) distinct channel models for different regions.

6.1.1. Virtual dependency channel revisited

In DVC, the side information Y at the decoder is computed as the prediction of the video frame X. The prediction errors in Y are modeled as noise N:

$$X = Y + N. \tag{6.1}$$

In the earliest work on DVC and in the Source Encoding with side-information under Ambiguous State of Nature (SEASON) framework [52], the deviation of the side information Y from the video frame X is modeled as an additive stationary white noise signal N. Prakash *et al.*[52] state that the residual frame N will truly appear as white noise if the motion estimation is reliable. More sophisticated motion estimation algorithms can be used. Nevertheless, the above model for the dependency channel between X and Y is fundamentally flawed because of events like occlusion.

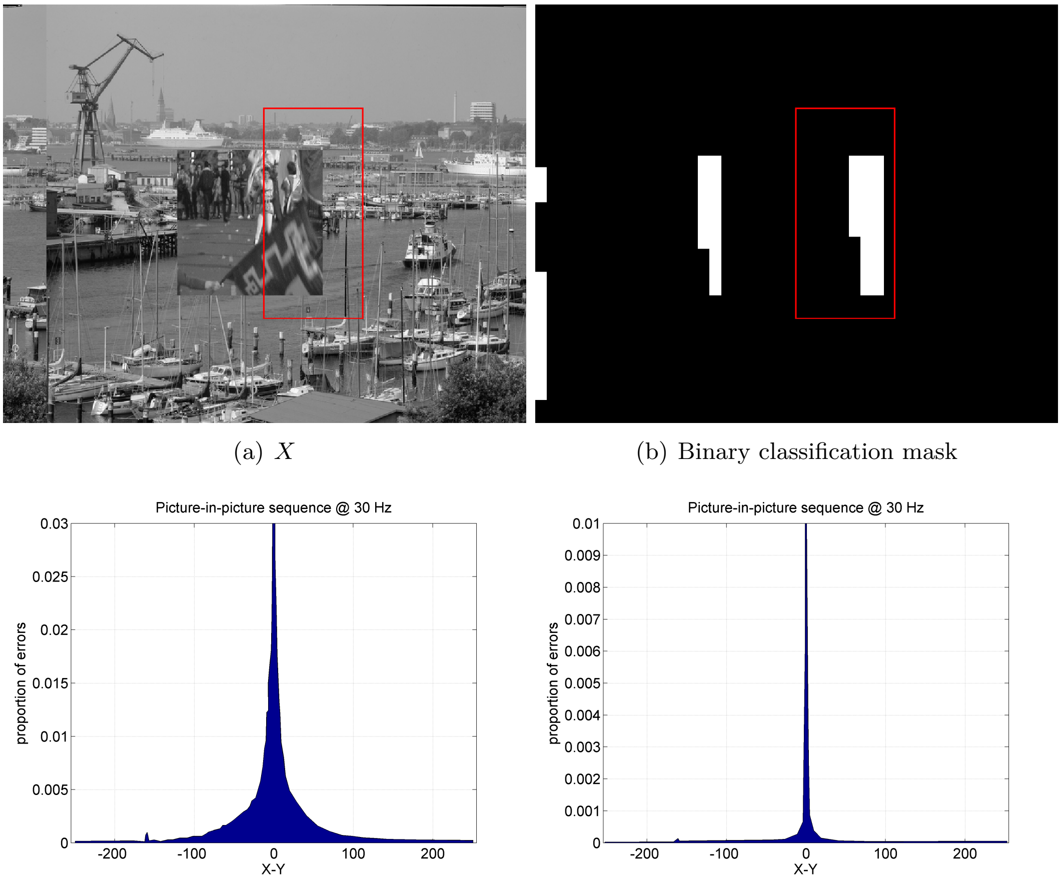
Occlusion occurs when video contains moving objects. These objects occlude other objects (or the background) and at the same time uncover previously concealed regions. In these regions, motion can not be estimated properly and consequently the motion compensated prediction will fail. This failure creates a noise contribution N that has statistical properties substantially different from the regions in which motion estimation and compensation can be carried out reliably.

Occlusion noise always occurs at the edge of moving objects or the edge of a video frame in case of camera panning. In addition, occlusion noise is hard to characterize. The dependencies $P(X|Y)$ in the unreliable regions are difficult to estimate since the revealed area is not known. Here, we have to rely on hole filling, for example by background extrapolation. Since the noise in the unreliable and reliable areas are both modeled by N, we have to conclude the noise process in Eq. (6.1) is inherently non-stationary.

A more accurate model for describing the VDC between X and Y is based on the observation that the noise N is a spatial mixture of two or more different noise processes. To distinguish between a reliable and an unreliable class we propose to model the two classes with different Laplacian PDF models. For instance we model the reliable class with a small variance and the unreliable class with a large variance.

6.1.2. Experimental validation with manual segmentation in pixel domain

To analyze the potential of using two classes to model the VDC, we consider the coding performance. We do not yet take into account the problem of region classification. Hence, we manually assign a binary mask to separate the reliable from the unreliable regions.



(a) X

(b) Binary classification mask

(c) Histogram unreliable regions

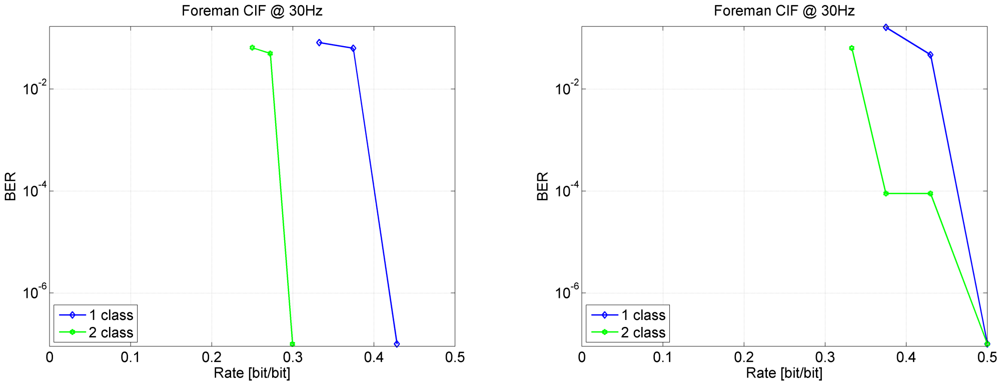
(d) Histogram reliable regions

Figure 6.1.: (a) Original frame X,(b) categorization into reliable (black) and unreliable (white) regions, (c) histogram (zoomed) of unreliable regions and (d) histogram (zoomed) of reliable regions.

We carry out an experiment similar to the one in Chapter 3.3.1, which only considered a stationary VDC model. Figure 6.1 shows an example frame of the synthetic Picture-in-picture sequence with its corresponding categorization into a reliable and an unreliable class. Further, we show the histograms of the two classes. The difference between the histograms illustrates, that one channel model is not able to model both regions accurately. We then look at two different assumptions for the channel model:

2 class model. The purpose of this model is to identify the performance gain when taking non-stationarity into account. This model models both regions separately. The channel is assumed to be non-stationary. A different PMF is used for the reliable and unreliable regions.

1 class model. The purpose of this model is to provide the reference with one class for the VDC. The channel is assumed to be stationary and the PMFs of the reliable and unreliable regions are combined into a single channel. The combined PMF is the average of the two PMFs, weighted by the number of reliable/unreliable pixels.



(a) Picture-in-picture data

(b) Picture-in-picture synthetic

Figure 6.2.: Results encoding/decoding subframe, with and without informing the decoder about occluded regions with (a) $P(Q|Y)$ measured from the data and (b) $P(Q|Y)$ from a synthetic channel model ($\sigma^2 = 1$ and $\sigma^2 = 510$).

Figure 6.2 shows the performance for the two channel models in terms of their BER for a number of bit rates. First, the PMF is measured from the data, constituting an oracle estimation. Following the argumentation in Section 3.3.2, the channel is modeled based on two Laplacian distributions ($\sigma^2 = 1$) and ($\sigma^2 = 510$).

It should be noted that the manual segmentation into reliable and unreliable pixels is not free of misclassification. In combination with fixing the reliable pixels to ($\sigma^2 = 1$)

and the unreliable pixels to ($\sigma^2 = 255$) the decoding might converge to an incorrect result as is the case in Figure 6.2 (b).

The results in Figure 6.2 show that the performance of decoding improves if the mixed noise model is assumed. First, with accurate channel parameters, the bit rate can be reduced by 30% in Figure 6.2 (a). Second, with fixed channel parameters ($\sigma^2 = 1$ and $\sigma^2 = 510$), the bit rate can still be reduced by 20% in Figure 6.2 (b). With these results we put forward that future practical distributed video coders should incorporate more than one class for the VDC.

6.2. Classification oracle in transform domain

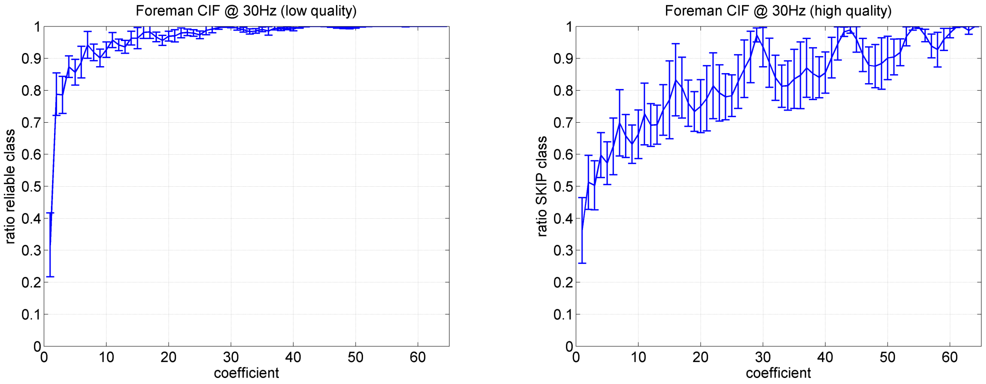
In the previous section we observe performance gains when applying classification to model the non-stationary VDC. The performance gains are observed for a pixel-based DVC system. The classification is based on manual segmentation. To investigate this issue further in the transform domain, we consider a controlled experimental setup with an oracle classification. Here, we do not segment the classes manually, but employ a threshold for the difference between reconstructed reference frame $\hat{X}$ and side information Y. Such oracle classification is closer to the ground truth than our coarse manual segmentation.

6.2.1. Model for transform domain classification

For each band, we have a classification into reliable and unreliable coefficients. We base the classification, henceforth referred to as oracle classification, on the difference between the to be decoded reference frame $\hat{X}$ and side information Y. It should be noted that in practice the decoder does not have access to $\hat{X}$ and therefore any practical classification will perform worse than the oracle classification.

The oracle classification has an important tuning parameter, namely the threshold between the classes. As will be shown in the experiments, the most RD efficient threshold is at a difference equal to zero between $\hat{X}$ and Y. The classification in reliable and unreliable coefficients then complies with the distinction between skip and inter mode in conventional video coding. To put a high certainty on the reliability of the reliable class, while being able to deal with errors from misclassification, we use a Laplacian distribution with $\sigma = 0.1$ to model the reliable class in the experiments.

For higher frequency coefficients, which are heavily quantized, the errors are usually very close to zero. At such high frequencies, the quantization is coarse for all qualities. Consequently, quantizing $Y + N$ for very high quantization parameters becomes similar



(a) $QP = 16$

(b) $QP = 4$

Figure 6.3.: Relative amount and standard deviation of perfectly predicted coefficients, assigned to the reliable class, for the lowest and the highest quality of the Foreman sequence at CIF resolution.

to quantizing only Y. For lower frequency such high quantization parameters are only present at lower reconstruction qualities.

The higher the reconstructed quality of $\hat{X}$, the finer the quantization. Figure 6.3 shows that ratio of the reliable class for two quantization levels. The high frequency coefficients are well predicted at both qualities. However, at the lowest quality ($\hat{X} = 31.5dB$) also a large fraction of the low frequency DCT coefficients is predicted correctly. At the highest quality ($\hat{X} = 39.5dB$) the number of correct low frequency coefficients is much smaller. Furthermore, the ratio deviates far more over different frames.

6.2.2. Model to introduce misclassification

To analyze the robustness of the classification we have to consider the effect of classification errors. For the oracle classification the reliable class only contains reliable DCT coefficients and the unreliable class only unreliable DCT coefficients. We introduce classification errors by manually misclassifying a certain number of DCT coefficients. Here, the fraction of misclassified coefficients refers to the percentage of reliable coefficients that are moved from the reliable to the unreliable class. A similar fraction is moved from the unreliable to the reliable class.

In practice misclassification does not occur randomly but is more likely to occur for coefficients, that are closer to the threshold and consequently the wrong class. For the coefficients in the unreliable class, we misclassify the coefficients with the smallest error $\hat{X}$-Y first. The reliable coefficients, all with the same distance from the zero threshold, are misclassified randomly. We label the misclassification scheme **misclass**.

6.3. Evaluation of oracle classification and sensitivity towards misclassification

In this section we investigate the oracles influence on the RD performance in a transform domain DVC scheme [28]. We use the CIF resolution Foreman and Hall-monitor sequences. The RD curves include all (300) frames for a frame rate of 30 frames per second. Only the first three frames are intra coded. The remaining frames are all extrapolated WZ frames [26].

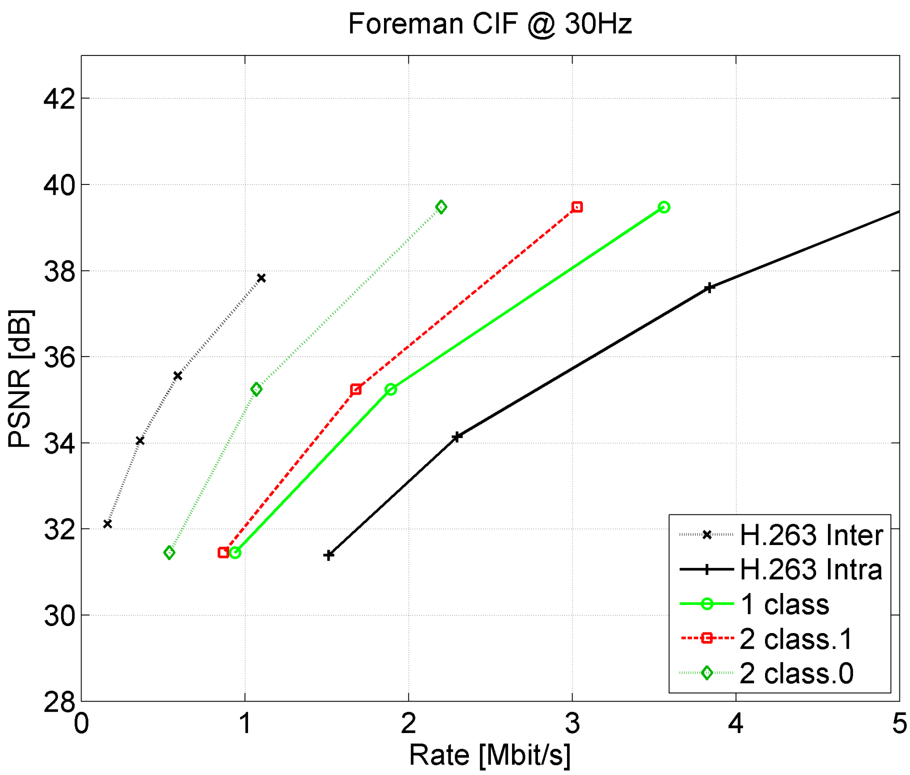
6.3.1. Oracle classification without errors

Figure 6.4 shows the RD performance of three classification approaches. For comparison we included reference results with H.263 intra and inter and a stationary DVC one class model with **1 class** [28]. For the proposed two class model we show **2 class.1** and **2 class.0**, which are similar, except for the threshold indicated by the extension. A threshold of zero effectively implements a decoder-based skip mode.

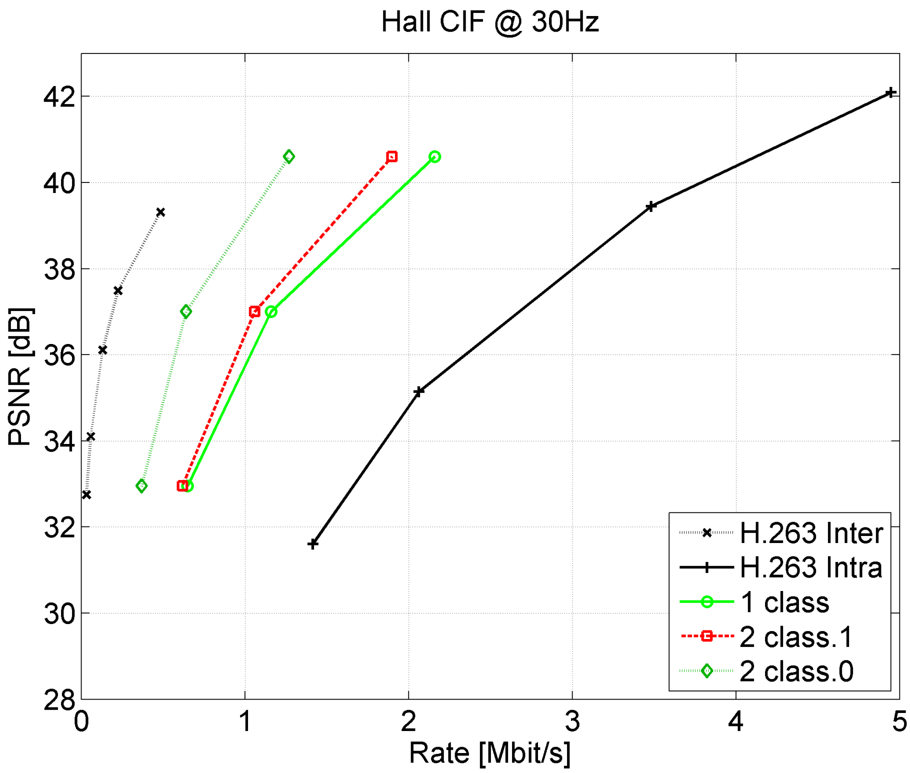
The most important observation for both sequences in Figure 6.4 is the large difference between a threshold of 1 or 0. While the former **2 class.1** only gives a moderate improvement over **1 class**, the latter **2 class.0** decreases the bit rate by about 40%. Since with a threshold of zero, all coefficients in the reliable class are correctly predicted, in theory no bits at all are needed for these coefficients.

In our setup however, the model is peaked around zero with $\sigma = 0.1$. With such a high certainty, the decoder requires only very few bits. By contrast, the unreliable DCT coefficients are decoded using a less peaked distribution compared to the one class case. Such a broader distribution is beneficial for the RD performance as the decoder relies more on the error correcting syndromes than on the unreliable side information for these coefficients.

The oracle classification is outperformed by H.263 inter. The performance difference between DVC and H.263 inter is partially caused by the higher side information quality for H.263. But for the low motion Hall-monitor sequence the side information quality in DVC is almost as good as the quality of a motion oracle in H.263. Here, the performance difference between DVC and H.263 is mainly due to the high efficiency of the skip mode



(a) Foreman



(b) Hall-monitor

Figure 6.4.: RD performance of oracle classification for (a) Foreman and (b) Hall-monitor sequence.

in H.263.

Rather than just skipping coefficients, H.263 can completely skip many macro blocks in the Hall-monitor sequence. In conclusion, our decoder-based oracle classification can only compete up to a certain point with the encoder-based skip mode of H.263. While H.263 can efficiently implement a skip mode for coefficients, blocks and complete frames, our oracle classification is limited to efficiently encode reliable coefficients. The coding efficiency is further limited by the performance loss of LDPC codes for very low conditional entropies [50].

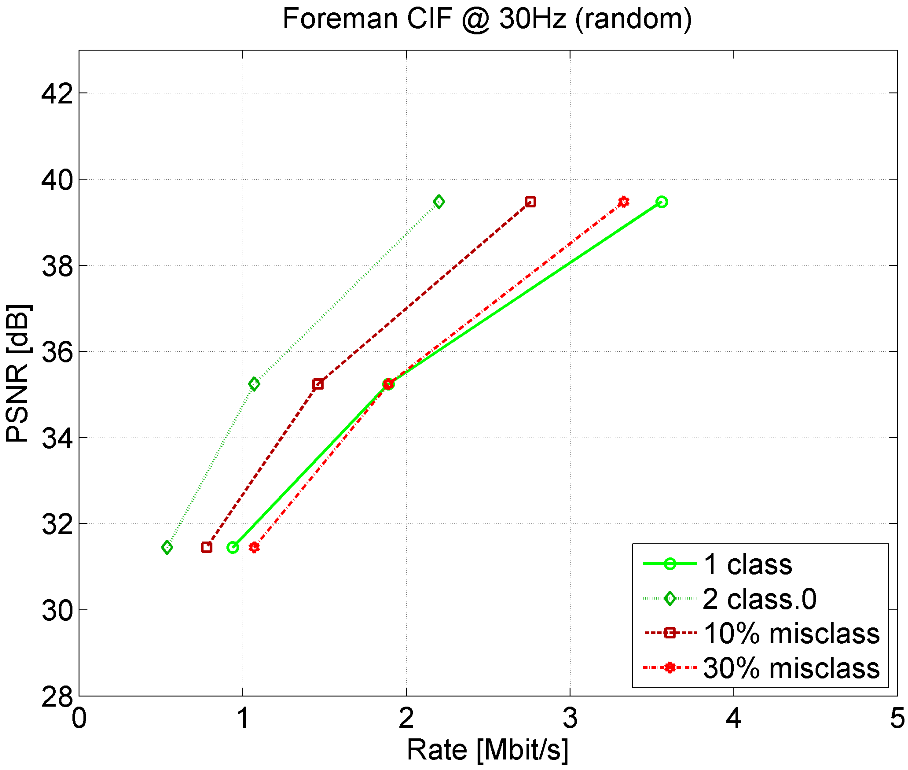
6.3.2. Oracle classification with misclassification

The RD performance gain of 3dB by oracle classification highlights the importance of accurate VDC modeling. In the following, we investigate the sensitivity to classification errors. For this purpose we analyze misclassification and its impact on the RD performance. It must be pointed out that the variance of the Laplacian model for each class is calculated after introducing misclassifications. As a result decoding is always possible.

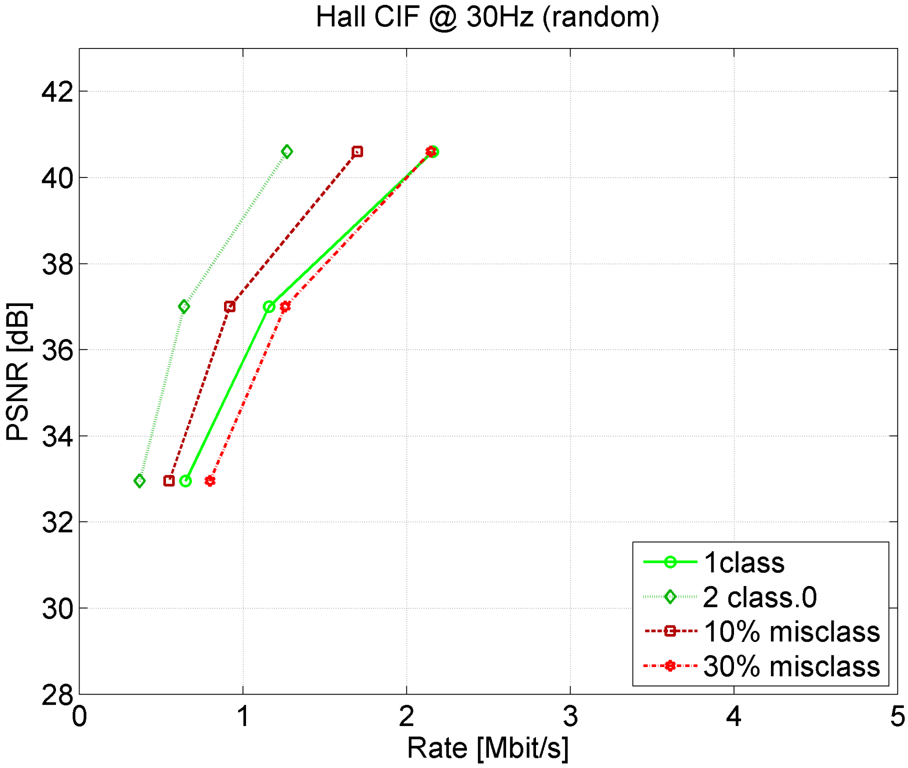
Figure 6.5 then shows the RD performance for **misclass**. We chose to limit the fraction of misclassified coefficients to 10% and 30%. We expect the former to be very difficult to achieve in practice and hence consider the latter to be more relevant for a practical system. We then observe a high sensitivity to misclassification.

The RD performance in Figure 6.5 degrades rapidly with an increase in the fraction of misclassified coefficients. Even for a very small misclassification of 10% the benefit from using two classes over one class is cut in half. At 30% a one class model outperforms the two classes approach by up to 1.5 dB at low qualities. At higher qualities, the one class model performs either equally for the Hall sequence or 0.5 dB worse for the Foreman sequence. Here, the reliable and unreliable class are more balanced.

In conclusion, since only the zero threshold shows a significant performance gain, to avoid misclassification it is necessary to identify perfectly predicted areas. It is questionable whether even sophisticated classification schemes can reduce misclassification below 10%. We already consider 30% misclassification a challenge without access to the reference frame X at the decoder.



(a) Foreman



(b) Hall-monitor

Figure 6.5.: RD performance of oracle classification with misclassification for (a) Foreman and (b) Hall-monitor sequence.

6.4. Classification based on motion estimation and motion learning

To incorporate multiple classes at the decoder, an automatic classification into reliable and unreliable regions is needed. In this section we investigate such automatic classification methods to distinguish between reliable and unreliable regions. For that purpose we consider classification based on motion information, which is available before and during LDPC decoding. The motion information is used to generate binary masks, indicating unreliable regions.

6.4.1. Classification based on motion estimation

Classification based on block-based motion estimation

First, we consider information from the block based CARS motion estimation, we use to generate the side information. With the extrapolation scheme presented in Chapter 4, we can use the position of vector collisions and holes to identify unreliable regions. However, to accurately classify reliable and unreliable regions is difficult due to halo effects.

Halo effects occur around a foreground object in a video sequence when the object is moving relative to a background. The background could be stationary and the object moving or vice versa. For example, the face of a person walking in a scene could seem to be surrounded by a halo, as if a portion of the background was moving along with the face [7]. The origin of the halo effect is erroneous motion estimation in occlusion areas.

While the errors in the side information are located on the edges themselves, the vector collisions and holes form a halo around the edges. Furthermore, homogeneous areas like for instance the white helmet in the Foreman sequence can induce vector collisions, which are then classified as unreliable. The extrapolated prediction for these regions however does not suffer from these vector collisions. Hence, the vector collisions lead to misclassification.

For a closer estimate of the edges, we consider a global motion-based approach to identify foreground edges. The approach works as follows.

Global motion estimation

We consider a three parameter global motion estimation. We restrict the number of parameters to two translational and a single scaling parameter. As such the global motion estimation is suited for estimating translational motion and zooming. The three parameters are estimated using the gradient-based search method of Hager and Belhumeur [49].

We find values for the translation and scaling such, that the difference between the predicted video frame and the original frame is minimal. This scheme was introduced in [27]. To prevent independent motion from influencing the global motion parameters, a dynamic threshold to exclude the pixels with the highest error is used. After initial convergence to the global parameters, further iterations only include the 80% least error pixels.

Classification based on global motion estimation

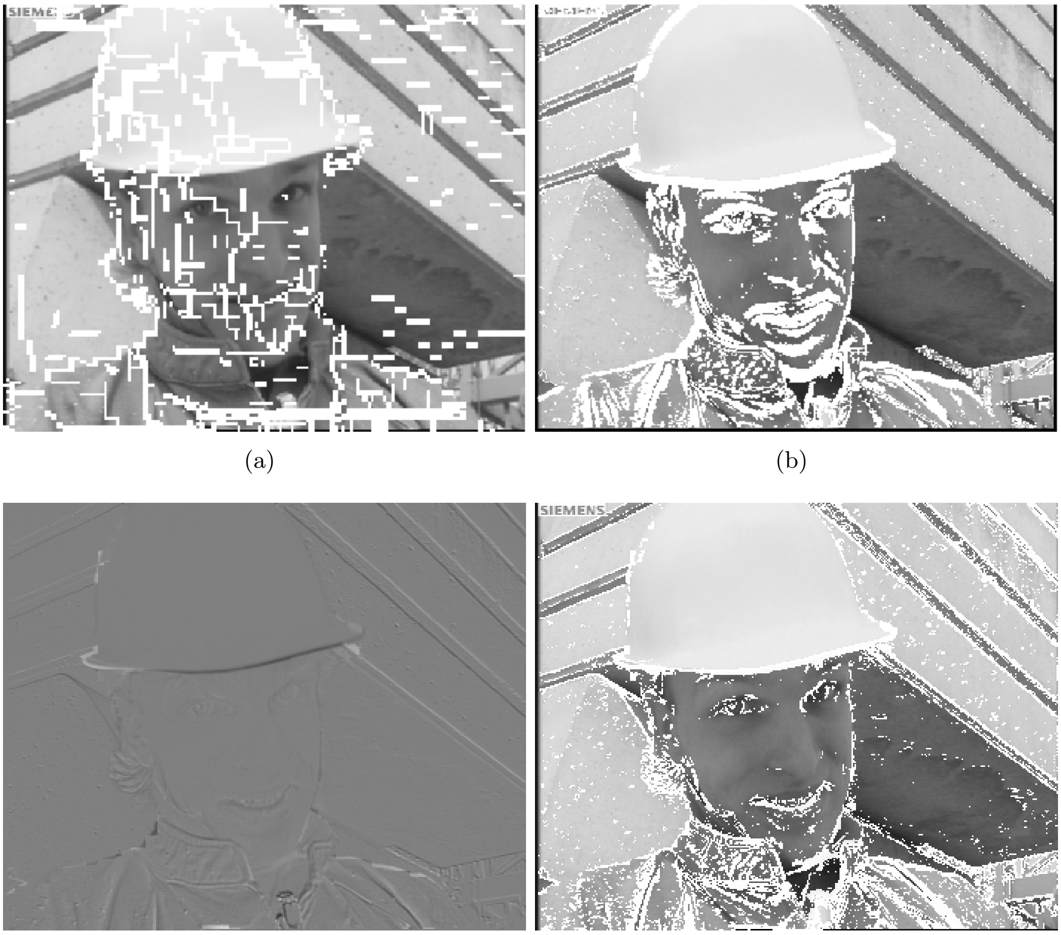
We consider the remaining 20% of pixels with the highest error to classify unreliable regions. Since we use discrete histogram binning during the global motion estimation until at least 80% of the pixels are included, the amount of pixels classified as unreliable can vary from 0 to 20%. The unreliable pixels belong to distinct edges, especially edges of independently moving foreground objects.

Block-based versus global motion estimation

Figure 6.6 shows an example classification for a single frame in the Foreman sequence. The unreliable class is indicated by a mask of white pixels overlaying the frame. Figure 6.6 (a) shows the block-based and Figure 6.6 (b) the global motion-based classification. Figure 6.6 (c) illustrates the differences between X and Y. To convert the differences into a binary mask which can be applied to our classification problem, we introduce an oracle mask in 6.6 (d). For the threshold between reliable and unreliable pixels we manually found $|X - Y| > 6$ to provide a good distinction between the two classes.

We observe, regions classified as unreliable for the block-based approach are on a halo around the object boundaries. By contrast, the unreliable pixels in the global motion approach are concentrated on the edges themselves and especially the foreground edges, which do not follow the global motion. As such the binary mask generated by means of global motion estimation is better suited to classification.

Nevertheless, the quality of the classification based on motion estimation suffers from large and unpredictable motion. Since the global motion estimation is performed between the two previous frames, it is not necessarily accurate for the current frame. Hence, Section 6.4.2 will introduce a motion learning scheme, where the classification takes the motion in the reference frame itself into account.



(a) (b) (c) (d)

Figure 6.6.: Classification masks for (a) global, (b) block-based, (c) the differences $X - Y$ and (d) the oracle mask for $|X - Y| > 6$.

6.4.2. Classification based on motion learning

The binary masks generated for the purpose of classification for both motion estimation schemes do not have access to the reference frame X. Consequently, they are not necessarily accurate for the reference frame. In this section we consider a motion learning-based approach with access to partial reference frame information.

In [93], Varodayan *et al.* propose to replace the motion estimation at the decoder side with an unsupervised motion learning scheme, blending motion estimation and channel decoding into a single procedure. However, the improvement of motion learning over motion estimation turned out to be very small. Using the same concept, we propose to employ the motion learning in addition to the motion estimation, rather than replacing it. The motion learning implementation from [93] is available at [9].

We investigate the applicability of motion learning to our classification problem. With motion learning is possible to use partial reference information from the already received channel coding syndromes. We propose to use the partial reference information to locate prediction errors in Y. If the motion learning is able to find motion between the side information and the (partial) reference frame, there are still errors in the side information. The corresponding regions should then be classified as unreliable.

Motion learning

The motion learning is an iterative Expectation-Maximization (EM) algorithm. During the E-step, the decoder updates the motion field distribution. Once the distribution of the motion field has been updated, in the M-step, the decoder proceeds to maximize the likelihood of Y and the syndrome S. The algorithm iterates between the E-step and the M-step, learning the motion vectors and improving the quality of the soft estimate of the WZ frame.

The unsupervised motion learning, introduced in [93], uses the syndrome bit stream of a WZ frame to learn the motion vectors. Figure 6.7 shows the flow diagram for the decoding process. The algorithm iteratively updates the probability distribution of the motion. At first the motion distribution is initialized with a default setting. Together with the previous decoded frame and its VDC information, the algorithm is able to generate soft side information.

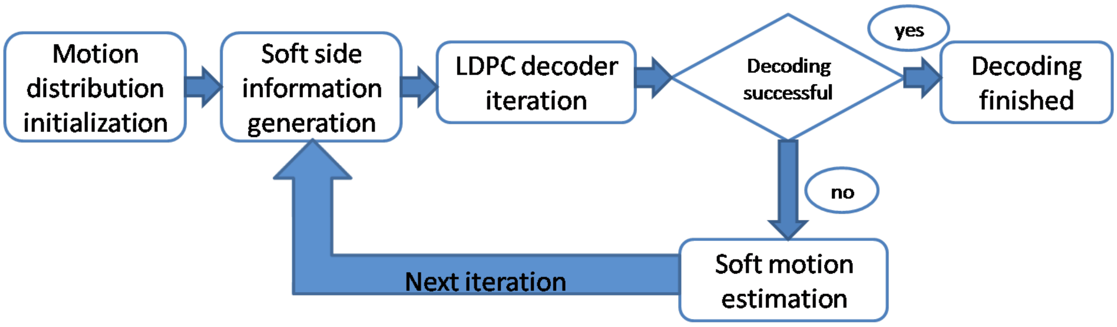


Figure 6.7.: Conceptual flow diagram for unsupervised motion learning [93].

The LDPC decoder then generates an improved soft estimate of the WZ frame based on the soft side information and already received error correcting syndromes. Since the new soft estimate is a better estimation of the WZ frame, the algorithm can update the probability distribution of the motion. For that purpose it performs a soft motion search between the previous decoded frame and the current soft estimate. The algorithm iteratively repeats itself until decoding is successful.

Classification based on motion learning

To classify unreliable regions we analyze whether motion learning is able to find motion between the side information and the (partial) reference frame by means of the motion field distribution. An example of how the motion field distribution behaves are given in Figure 6.8 (a). The initialization of the probabilities highly favors zero motion.

If the side information is reliable for a motion block, the final motion distribution will be peaked at zero motion, e.g. Figure 6.8 (b). If the side information is unreliable, the distribution will show peaks at different positions, e.g. Figure 6.8 (c). However a similarly peaked distribution might occur in homogeneous areas. This problem is similar to vector collisions in block-based ME for homogeneous areas.

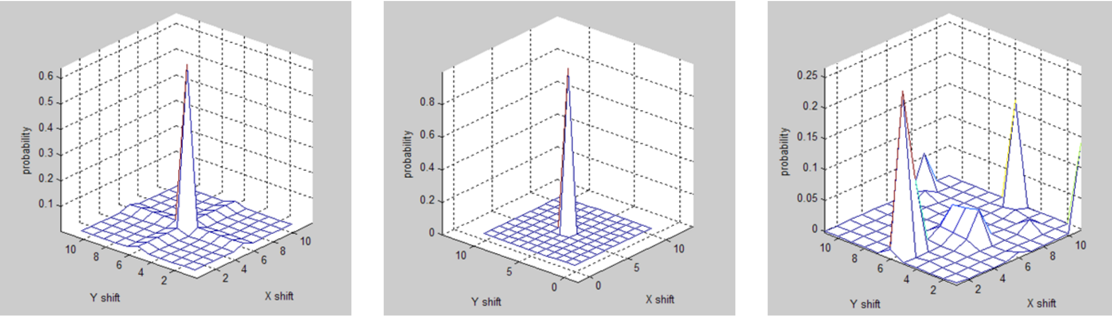


Figure 6.8.: Example of first motion field distribution update (From left to right: PMF initialization, well predicted block, block with errors.

The binary mask indicating unreliable regions is generated as follows:

1. Use previously decoded frames $\hat{X}^{k-1}$ and $\hat{X}^{k-2}$ for motion compensated extrapolation.

2. Use resulting side information as input for the motion learning.

3. Check the probability distribution of the motion field.

4. Perform a predefined number of motion learning iterations. We experimentally found 10 iterations to provide sufficient convergence towards the final motion distribution.

5. Generate motion learning mask based on final motion distribution. For that purpose we use a threshold on the motion vector probability. If a non-zero motion vector has a higher probability than zero motion, the block is classified as unreliable. We use an additional Sum of Absolute Differences (SAD) threshold to exclude homogeneous areas, which can yield false motion probabilities.

6. Commence regular LDPC decoding, employing the binary mask of reliable and unreliable blocks.

It should be noted that the motion learning between the side information Y and the reference frame can improve the RD performance. To take the performance increase into account, we also investigate the performance gain of LDPC decoding with motion learning over regular LDPC decoding.

Three classes mask

To get a finer distinction between prediction errors, we propose to combine the binary mask of unreliable regions found with motion learning, with the binary mask found with global motion estimation. The benefits of the two binary masks are combined in the three classes mask. Each class can be modeled separately, allowing more accurate channel modeling.

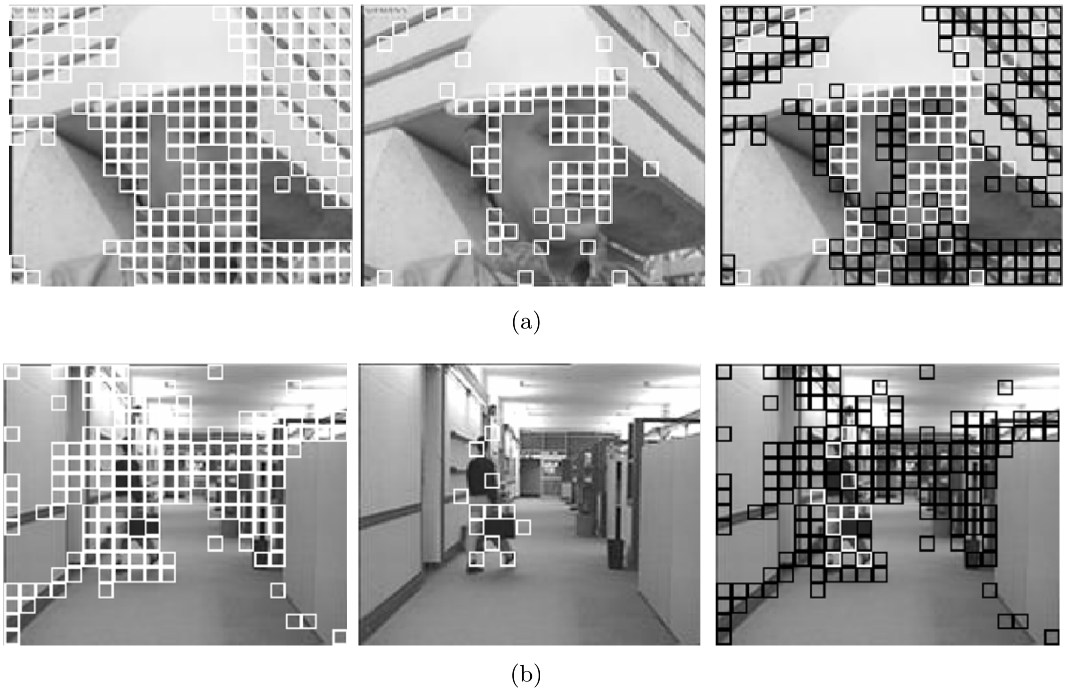
Previously, we introduced the global motion mask, with a threshold for unreliable pixels. To combine the pixel-based global mask with the block-based motion learning mask, we modify the global mask. In the block-based global mask each 8x8 block with at least two unreliable pixels from the pixel-based global mask is labeled unreliable. With such a low threshold, any blocks classified as reliable are most likely very well predicted.

Consequently, the global motion mask provides a good indication of well predicted areas. Furthermore, the motion learning mask gives a good indication of areas with high errors. With the complementary behavior the two masks can be used to derive a novel three classes mask with three reliability classes.

Very unreliable blocks Motion learning AND global motion mask indicate an unreliable block.

Relatively unreliable blocks Global motion mask indicates an unreliable block and motion learning mask indicates a reliable block.

Reliable blocks Motion learning AND global motion mask indicate a reliable block.



(a)

(b)

Figure 6.9.: From left to right: global motion mask, motion learning mask, 3 classes mask (white - very unreliable, black - relatively unreliable) for (a) Foreman and (b) Hall-monitor.

Example masks for the Foreman and the Hall-monitor sequence is given in Figure 6.9. The leftmost global mask indicates mostly blocks on edges as unreliable. The motion learning mask shows unreliable blocks mostly in the boundaries of the moving objects. For the rightmost three classes mask, white blocks are very unreliable, black blocks are relatively unreliable and the remaining blocks are reliable.

6.5. Evaluation of proposed classification schemes

In this section we compare and evaluate the classification schemes introduced in Section 6.4. First, we compare the two motion estimation schemes. Then, we focus on the motion learning and 3 classes masks.

6.5.1. Classification based on motion estimation

For this experiment we use real video sequences in CIF resolution with a frame rate of 30 frames per second. The sequences range from the low motion Hall-monitor sequence to the very large motion Stefan one. The varying sequence properties provide a representative subset of real video content. The side information Y is extrapolated and uses the block-based CARS approach.

Block-based versus global motion estimation

We consider two evaluation criteria to compare the classification quality. First, the difference in side information quality between reliable and unreliable regions in terms of PSNR. Second, the amount of pixels labeled as unreliable. The second evaluation criteria requires a ground truth to compare with. For that purpose we employ the oracle mask for $|X - Y| > 6$ introduced in Section 6.4.1.

For each video sequence in Table 6.1 we consider the side information Y with its respective PSNR. We then show how the PSNR differs between reliable and unreliable regions for each of the three classification schemes. In addition we provide the percentage of regions classified as unreliable.

Table 6.1.: Comparison of the region classification schemes [27].

sequence	area label	global motion	block-based	$\|X - Y\| > 6$
Stefan	PSNR unreliable [dB] (A)	20.5	20.8	19.6
(PSNR Y	PSNR reliable [dB]	26.1	25.4	40.1
=24.6 dB)	percentage of A	15.4%	14.6%	33.0%
Foreman	PSNR unreliable [dB] (A)	25.8	27.7	22.7
(PSNR Y	PSNR reliable [dB]	31.7	30.8	40.3
=30.2 dB)	percentage of A	13.0%	14.5%	17.5%
Coastguard	PSNR unreliable [dB] (A)	26.1	26.7	24.6
(PSNR Y	PSNR reliable [dB]	33.4	31.7	38.9
=31.3 dB)	percentage of A	13.6%	3.8%	19.8%
Hall-monitor	PSNR unreliable [dB] (A)	27.5	30.0	23.8
(PSNR Y	PSNR reliable [dB]	37.3	35.0	40.5
=34.7 dB)	percentage of A	9.5%	2.3%	6.0%

In Table 6.1, we compare the classification schemes. With regards to both criteria, the block-based approach performs quite poorly because of halo effect and poor motion estimation in homogeneous regions. The global motion approach shows better results, especially in terms of the PSNR differences. However, the percentage of unreliable regions indicates that the global motion approach suffers from its fixed threshold. As introduced in Section 6.4.1 the number of pixels classified as unreliable can vary from 0 to 20%. Consequently, it is hard to correctly classify unreliable regions if more than 20% pixels of a frame are unreliable as seen for the Stefan sequence with 33% unreliable pixels.

In conclusion, we prefer global motion estimation over block-based motion estimation for the purpose of classification. However, there is a large performance gap to oracle classification, indicating a large percentage of misclassified pixels.

Classification based on global motion estimation

To investigate the impact of the misclassified pixels we investigate the RD performance of the complete system. We focus on the preferred global motion classification **2 class global** and the oracle classification **2 class oracle**. Furthermore, we provide the one class case **1 class** for comparison. The parameters for all dependency models are estimated from the previously decoded frame.

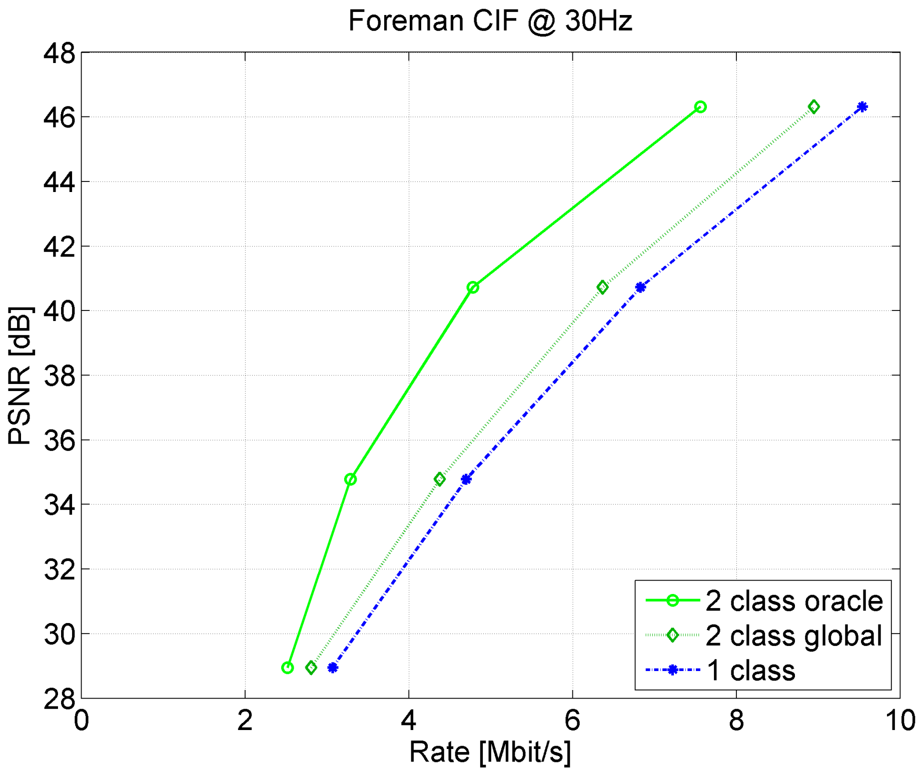
Figure 6.10 shows pixel domain results for the Foreman and Hall-monitor sequences. For the coarsely quantized Hall-monitor sequence we observe a performance degradation of the global classification **2 class global**. For that RD point the performance of the global classification is below the **1 class** case. The reason for this exception is a poor fit on coarsely quantized data due to a limited amount of pixels in small unreliable regions.

In Figure 6.10 (a), the oracle classification **2 class oracle** outperforms the one class model by 4 to 6 dB. For the global motion classification the performance gain to the one class model is reduced to less than 1 dB. The low-motion Hall-monitor sequence in Figure 6.10 (b) has only a small percentage of unreliable pixels, which reduces the performance gain of the oracle classification over the **1 class** model to 2 to 3 dB. Excect for the lowest quality, the global motion classification still maintains a performance that is up to 2 dB higher than the one class model.

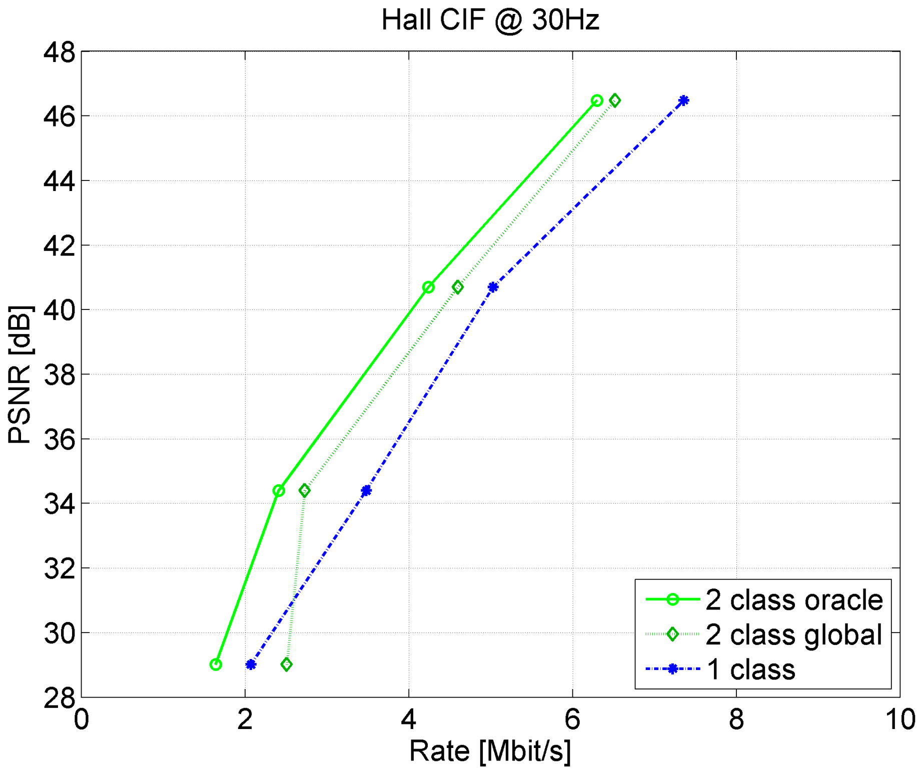
In conclusion, the global motion classification can provide a significant RD performance increase in pixel-based DVC. How the increase translates to the transform domain will be investigated by means of the block-based global motion classification adopted for the 3 classes mask.

6.5.2. Motion learning-based masks

For this experiment we use the transform domain code provided by Varodayan et.al. from [8]. Hence, the testing conditions have to be adapted. Due to the high decoding complexity we use QCIF resolution sequences Foreman and Hall-monitor at a frame rate of 30 frames per second. For Foreman we use frames 48 to 96 and for Hall monitor



(a)



(b)

Figure 6.10.: Coding rates by applying different classification masks: 1 class, 2 class oracle and 2 class global for Foreman (a) and Hall-monitor (b).

the first 50 frames. Initially, we perform an oracle estimation of the VDC. Later, we also show how the results change if we estimate the VDC parameters from the previous frame.

The schemes we compare are the following:

LDPC No mask is used and regular LDPC decoding performed.

Motion learning No mask is used, but motion learning is applied.

ML mask Motion learning mask with regular LDPC decoding.

GM mask Block-based global motion mask with regular LDPC decoding.

3 classes mask Combines the three classes mask with regular LDPC decoding.

Figure 6.11 compares the five schemes. For the Foreman sequence, **motion learning** provides a 0.4 dB gain over LDPC. For the Foreman sequence and the Hall-monitor sequence, we observe a 0.45 dB gain for the **GM mask** and **ML mask** over **LDPC**. The **3 classes mask** provides an additional 0.2 dB gain over the GM and ML mask.

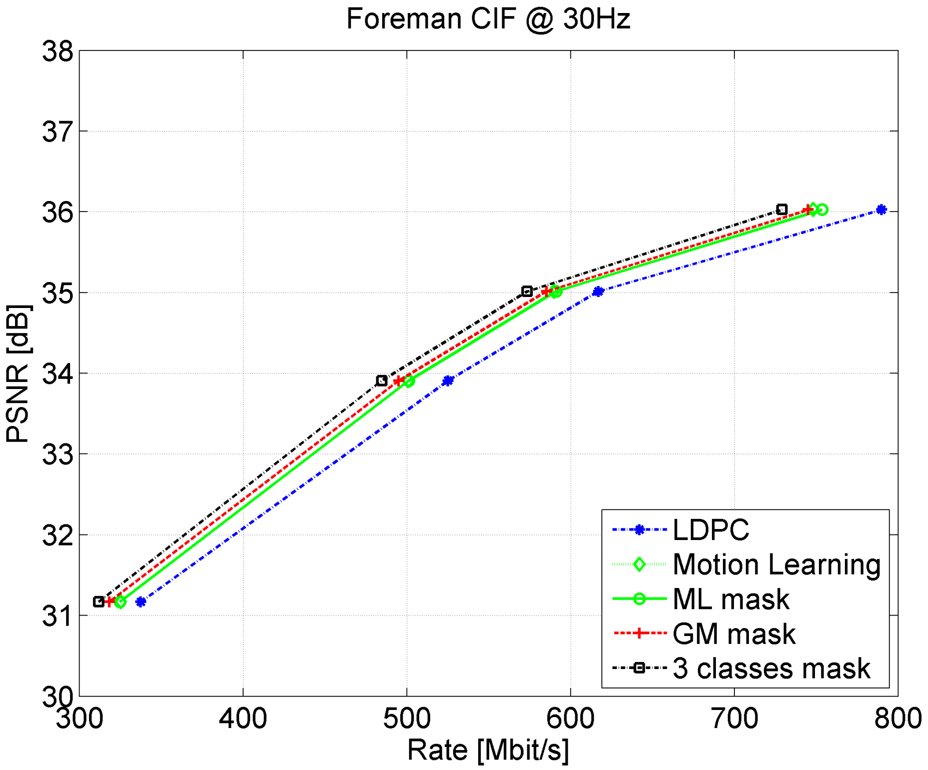
The benefit from using motion learning itself is sequence dependent. For the Foreman sequence **motion learning** outperforms **LDPC** by 0.4 dB. For the low motion Hall-monitor sequence, the side information is well predicted. For this sequence there is no performance gain of motion learning over regular LDPC coding.

Finally, if we do not use an oracle estimated VDC parameter but estimate it from the previous frame, all masking approaches lose in performance. As shown in Figure 6.12, the **3 classes mask** does not provide a noticeable improvement anymore. With its higher precision it is also most sensitive towards inaccurate modeling. As such it is not feasible to go beyond two models in a practical situation.

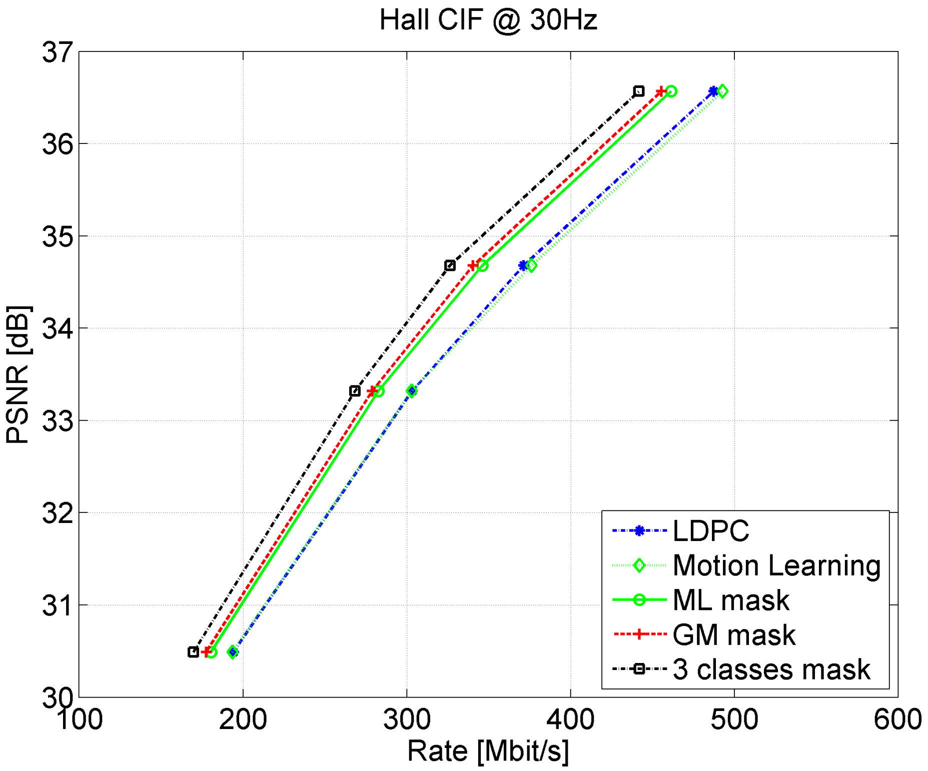
6.6. Discussion

Multiple channel models can help the VDC modeling
Since the VDC is non-stationary in nature due to events like occlusion, it is not possible to get the highest compression performance with a stationary VDC model. With manual classification we observe up to 30% bit rate reduction if we model two classes separately.

Classification oracle - large potential gain but sensitive to misclassification
We analyze the potential performance gain of accurate channel modeling in DVC by means of an oracle classification. A two class model distinguishes between reliable

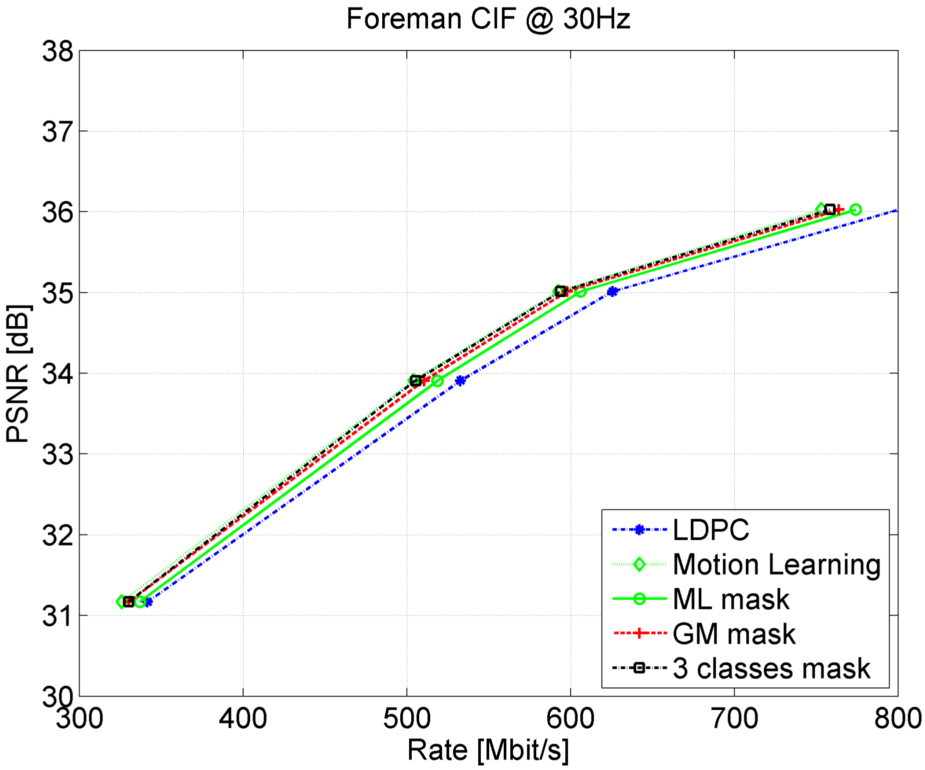


(a) Foreman

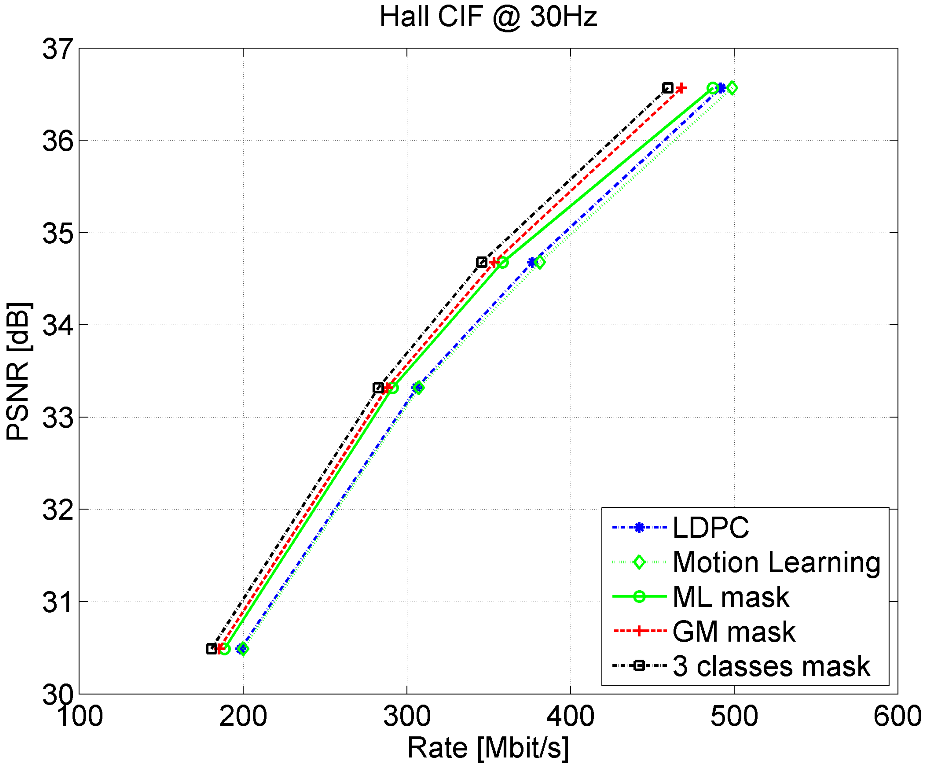


(b) Hall

Figure 6.11.: Coding rates by applying different (classification) schemes: LDPC, motion learning, ML mask, GM mask and 3 classes mask with oracle estimation of VDC parameters for Foreman (a) and Hall-monitor (b).



(a) Foreman



(b) Hall

Figure 6.12.: Coding rates by applying different (classification) schemes: LDPC, motion learning, ML mask, GM mask and 3 classes mask with VDC parameters estimated from previous frame for Foreman (a) and Hall-monitor (b).

and unreliable coefficients. The model reduces the bit rate by 40%, i.e. increases the RD performance by 3 dB. We then introduce misclassification. Already with 10% misclassified coefficients, we observe a RD performance decrease of 1.5 dB.

Since only the zero threshold shows a significant performance gain, it is necessary to identify perfectly predicted areas. It is questionable whether even sophisticated classification schemes can reduce misclassification below 10%. Already 30% misclassification is a challenge without access to the reference frame.

In addition we observe limits to the oracle classification itself. Since we only operate at the decoder side, it is not possible to skip blocks or complete frames. This limited adaptivity to non-stationarity in low complexity DVC is incurs a non-recoverable performance loss.

Classification based on motion estimation lacks reference information
For an automatic classification, we consider two motion estimation-based schemes, a block-based and a global motion-based approach. We find the global motion-based approach to provide a better class separation. The class separation increases the RD performance of a pixel-based DVC codec by up to 2 dB for the low motion Hall sequence. For the Foreman sequence the RD performance gain declines to 1dB. With larger motion, the classification accuracy suffers from a lack of access to reference information.

Motion learning provides partial reference information for the classification
To allow partial access to reference frame data, we employ a motion learning approach from literature [93]. The classification based on the motion learning shows a similar quality to the global motion-based classification. The motion learning is well suited for finding badly predicted areas. The global motion is well suited for finding well predicted areas. We then propose to combine the two approaches and get a novel three levels classification.

The three levels classification improves RD performance by 0.2 dB with accurate channel parameters. In practice, the channel model parameters need to be estimated from the previous frame. Then, the three levels classification only yields a negligible gain. We assert two classes are sufficient for further investigation.

Based on our findings regarding the high sensitivity to misclassification, we omit the classification approach from the benchmark in Chapter 7.

7. Performance benchmark of proposed DVC components

In the previous chapters we evaluated a realization for each of three studied components in a DVC system, namely channel coding, ME/MC and DCT quantization. The main focus of this chapter is then to provide a performance benchmark of the three components.

First, we report the latest performance of DISCOVER as state of the art in DVC. Further, we analyze the RD gap between DVC and conventional video coding. Finally, we discuss the trade-off between RD performance and encoder complexity, and conduct a benchmark of several state of the art schemes.

7.1. Background

The DISCOVER project [1] concludes that DVC-based architectures may present the following functional benefit [6]:
Flexible allocation of the overall video codec complexity *– Since the DVC approach allows moving part of the encoder complexity to the decoder, it clearly provides the benefit of a flexible allocation of the video codec complexity between the encoder and decoder. A particular case of the flexible allocation is the important case (for some applications) of low encoder complexity which may also imply lower encoder battery consumption, as well as cheaper and lighter encoders.*

Next to the processing during encoding, the transmission subsystem also incurs significant energy costs. Consequently, an increase in transmission bandwidth due to a decrease in processing complexity can incur higher cumulative energy costs. For instance Nimmagadda *et al.* present an adaptive image compression algorithm to save transmission energy through wireless networks in [68]. This method achieves 20% energy reduction on average by 35% reduction in transmission energy with 15% additional processing energy.

7.1.1. DISCOVER video codec as state of the art DVC

The state-of-the-art on DVC, by example of the DISCOVER video codec, is reviewed in [71]:

In terms of RD performance, the DISCOVER video codec already wins against the H.264/AVC intra codec for most test sequences, and for GOP=2. For more quiet sequences, the DISCOVER codec may even win against the H.264/AVC no motion codec. For longer GOP sizes, winning against H.264/AVC intra is more difficult, highlighting the importance and difficulty of getting good side information, notably when key frames are farther apart.

The DISCOVER encoding complexity is always much lower than the H.264/AVC intra encoding complexity, even for GOP=2 where it performs better in terms of RD performance. Since the DISCOVER codec performs better than H.264/AVC intra for GOP=2, for most sequences, this highlights that Wyner-Ziv coding is already a credible solution when encoding complexity is a very critical requirement (even if at the cost of some additional decoding complexity). Good examples for these applications may be deep space video transmission, video surveillance, and video sensor networks.

7.1.2. Encoder complexity and transmission bandwidth

The importance of power consumption in the case of wireless devices is well understood [45, 114, 48, 68]. Power is consumed by different tasks, namely, acquisition, processing, transmission and display. Consequently, less power consumption can increase battery lifetime of a constrained device [6].

The RD performance of a DVC system affects the amount of transmission bandwidth. The required bandwidth determines the energy cost for transmission. The processing cost incurred by processing during encoding has a measurable effect on the energy consumption [48].

First, we investigate the RD performance of the proposed DVC components. We compare each component with its conventional counterpart. Subsequently we focus on the encoder complexity and provide a benchmark that takes RD performance and encoder complexity into account.

7.2. RD performance comparison by component

The main contribution of this section is to analyze the performance gap by evaluating the three components investigated in this thesis. Contrary to previous chapters, the evaluation focuses not on the best RD performance in DVC, but on the RD performance gap between DVC and conventional video coding. For that purpose, we compare the RD performance of each DVC components with its counterparts in a state of the art video codec.

We chose H.263 as our reference scheme, more precisely the tmn 3.2 implementation [3]. Because of its modular behavior H.263 is more suited for analyzing separate components than its successor H.264. The components are listed in Table 7.1.

Table 7.1.: Respective H.263 and DVC components

Component	**H.263**	**DVC**	**Remark**
Transform	8x8 DCT	8x8 DCT	identical
Quantization	adaptive inter DCT coefficients	fixed per frame intra DCT coefficients	see Chapter 5
Encoding	source coding VLC	channel coding LDPC	see Chapter 3
ME	minimum residue with reference frame	true motion only previous frames	see Chapter 4
MC	P-frame MO	WZ-frame MX	see Chapter 4

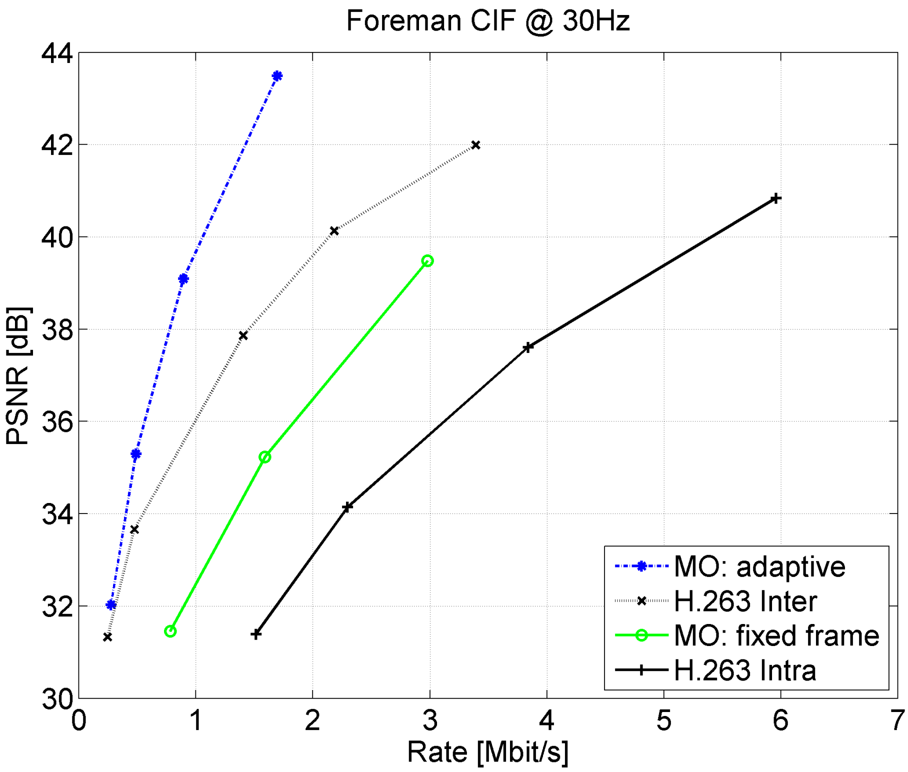
We investigate each component's influence on the RD performance. We use the CIF resolution Foreman and Hall-monitor sequences for a frame rate of 30 frames per second. Results for the Hall-monitor sequence are only listed when they present additional insights. As the results in this section are based on the "best" configurations from Chapter 3, 4 and 5, there is some overlap in the results.

7.2.1. Quantization

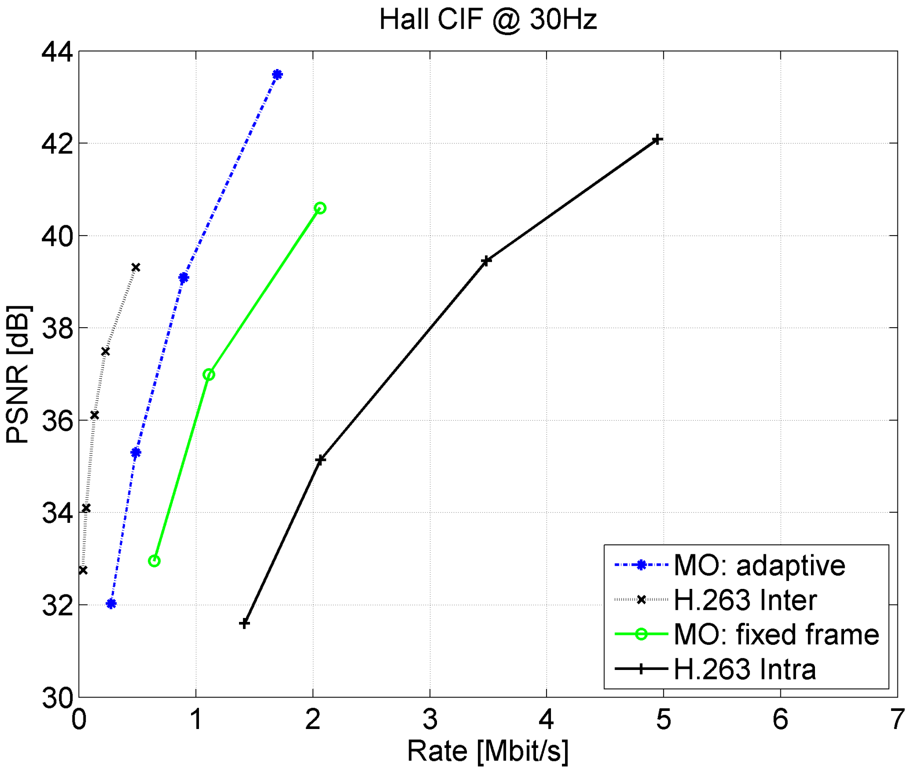
We include three DVC quantization schemes from Section 5.2 to the RD comparison. The first scheme is the fixed frame quantization we found to best compromise between adaptivity of quantization and overhead information required. The second scheme is the adaptive oracle quantization as an upper bound. Here, we omit the overhead information required to make this scheme practical.

To exclude the influence of the differences in ME/MC, we employ the motion oracle MO for the DVC quantization schemes. For conventional coding we provide RD results H.263 intra coding as a reference point and H.263 inter coding as the practical adaptive quantization scheme.

The first observation in Figure 7.1 is the performance of the adaptive DVC scheme. It is an oracle quantization and indicates the performance loss incurred by the reduced adaptivity in the fixed frame scheme. One question that might arise is how the **adaptive** scheme can outperform H.263 inter coding for the Foreman sequence. The



(a) Foreman



(b) Hall-monitor

Figure 7.1.: RD performance difference between quantization schemes for Foreman (a) and Hall-monitor (b) sequence.

reason is, that whereas H.263 has to encode zero runs, the adaptive DVC scheme does not. Especially at higher qualities with shorter zero runs this has a noticeable impact on the RD performance.

For the Hall-monitor sequence in Figure 7.1 (b) H.263 still outperforms the **adaptive** scheme by 3 dB. The reason for the better H.263 inter performance was introduced in Section 6.3.1, namely that H.263 can efficiently implement a skip mode for coefficients, blocks and complete frames. In addition, the inter DCT coefficients in H.263 inter have more zero coefficients than the intra DCT coefficients in the **adaptive** DVC quantization.

The H.263 inter coding outperforms the practical DVC **fixed frame** quantization by 3 dB. Since quantization and encoding are difficult to separate, the 3 dB loss for the Foreman sequence is caused by the combination quantization+LDPC for DVC and quantization+VLC for H.263 inter. For the Hall-monitor sequence the performance gap between H.263 inter coding and the DVC **fixed frame** quantization is larger with 6 dB.

7.2.2. Encoding

For the second comparison we restrict **VLC** and **LDPC** to the fixed quantization from [26], that is the **set globally** scheme from Section 5.2. As a consequence, the reconstructed frame suffers from clipping artifacts and the **VLC** itself suffers from a higher variance in the residue. Hence the performance of **VLC** with fixed quantization drops by 3dB compared to using adaptive quantization.

Figure 7.2 shows the respective RD performances. For the Foreman sequence **LDPC** loses almost 2 dB in PSNR. In comparison the quality loss for the Hall-monitor sequence is approximately 3 dB. This observation holds for both ME/MC schemes, **MX** and **MO**. The main difference between the sequences is a better prediction for the Hall-monitor sequence. A better prediction results in a lower conditional entropy. With a lower entropy the LDPC performance worsens [50].

In conclusion, the RD performance loss from fixed frame quantization and LDPC coding is significant. Compared to H.263 inter, the RD performance loss can add up to 3 dB for the Foreman sequence, of which quantization contributes 1 dB and LDPC contributes 2 dB to the performance loss. The RD performance loss is 6 dB for the Hall-monitor sequence, of which quantization and LDPC both contribute 3 dB to the performance loss.

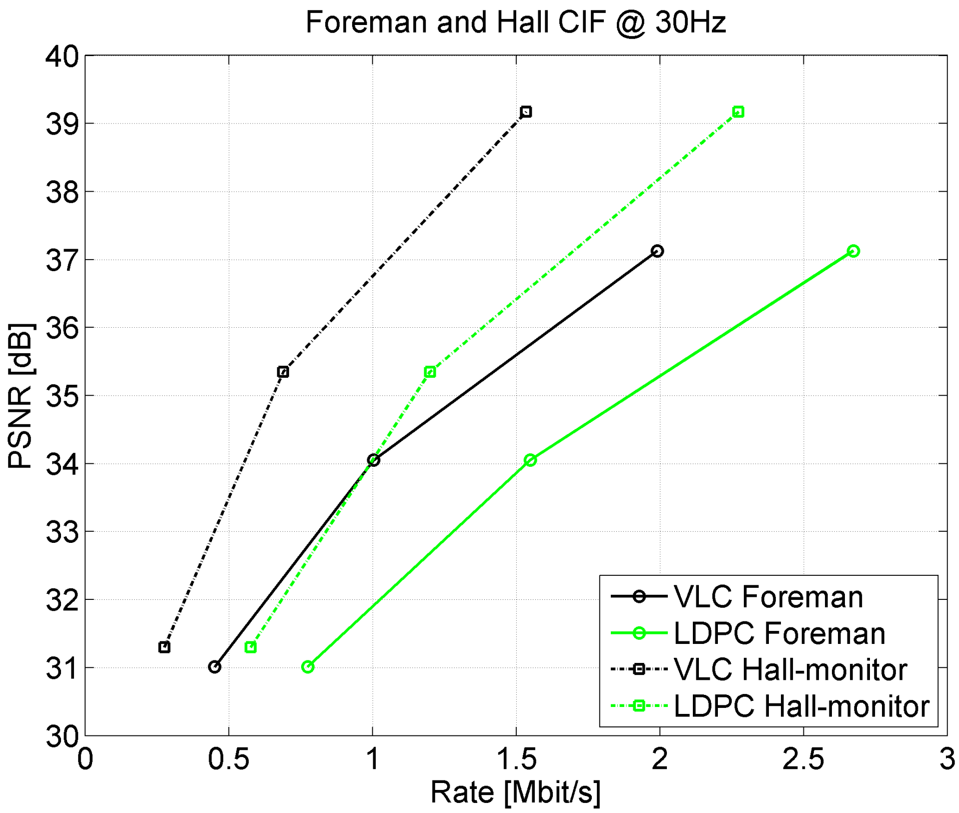


Figure 7.2.: RD performance difference between VLC and LDPC with fixed quantization for Foreman and Hall-monitor sequence.

7.2.3. Motion estimation and compensation

Finally we compare the side information quality. At first we investigate the trade-off between the prediction quality loss in DVC and the motion vector rate in H.263. For that purpose we focus on the difference between the motion oracle **MO** and extrapolation **MX**. From Section 5.3.1 we know that for the Foreman sequence the PSNR of the **MO** side information is 3 to 4 dB higher than the **MX** side information.

Figure 7.3 shows that the vector cost is small and does not increase for higher qualities. The motion vector overhead is negligible compared to the RD performance loss from the motion oracle to extrapolation. The RD gap between **MO** and **MX** also worsens for higher qualities.

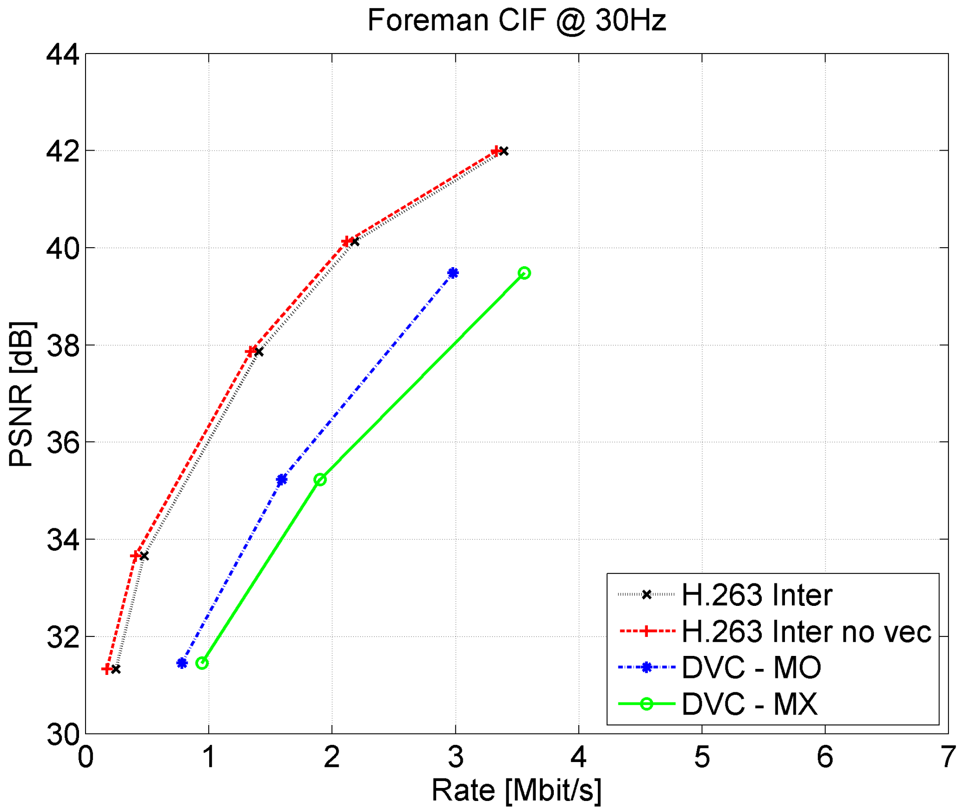
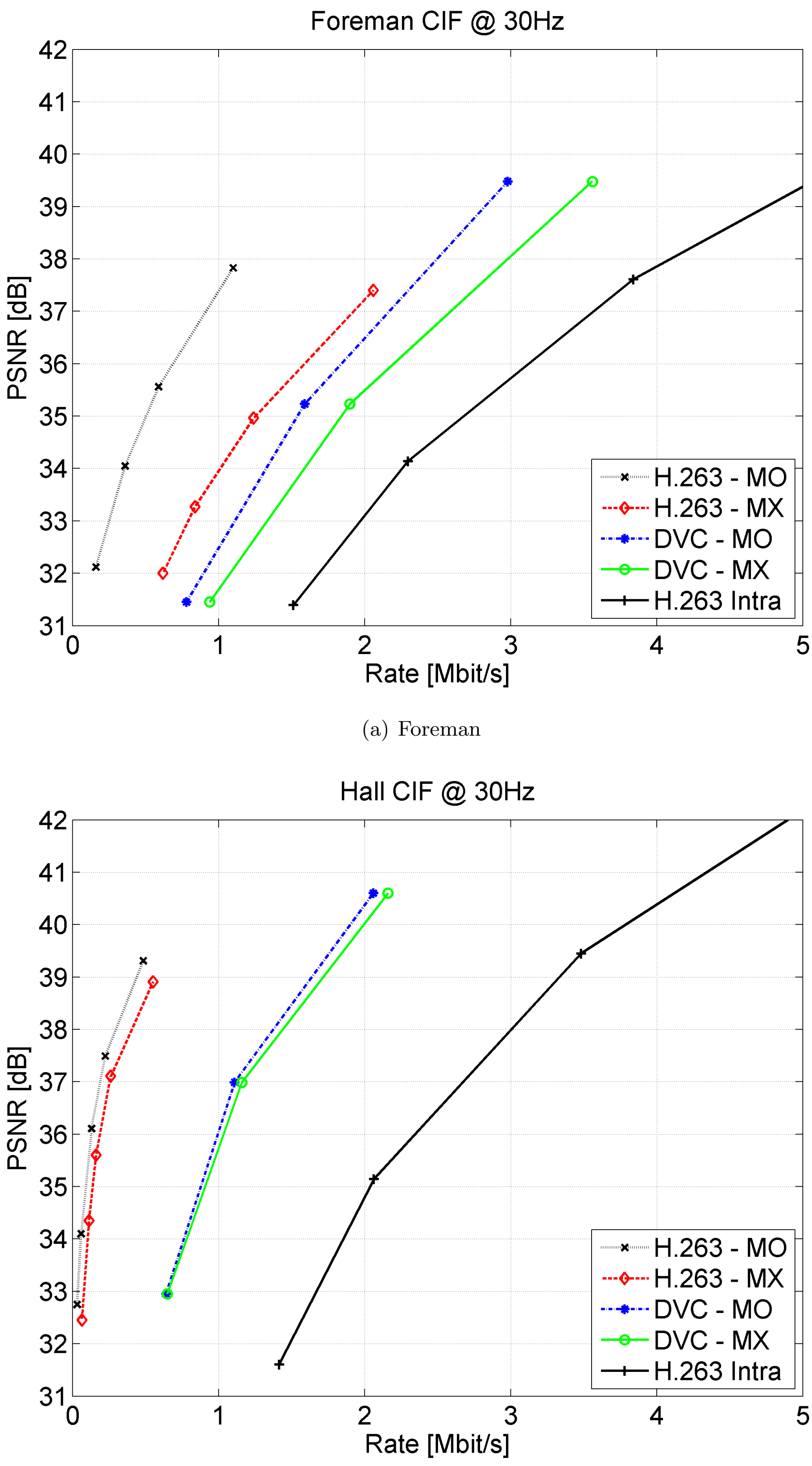


Figure 7.3.: RD performance difference between sending motion vectors and loss from MO to MX for Foreman sequence.

To analyze how the difference in side information PSNR propagates to the RD performance of the complete system we compare the RD performance of **MX** and **MO** in Figure 7.4. For H.263 the motion oracle outperforms extrapolation by almost 4 dB. There is a small additional loss in RD performance since the Huffman tables are optimized for the motion oracle and not extrapolation.

The difference in side information PSNR propagates fully to the RD performance difference between **MX** and **MO** for H.263. By contrast, the RD performance **MX** and **MO** only differs by 1 to 1.5 dB in DVC, i.e. the side information PSNR difference does only propagate partially.

For the low motion Hall-monitor sequence, the prediction quality is almost similar for motion oracle **MO** and extrapolation **MX**. This propagates to the RD performance



(a) Foreman

(b) Hall-monitor

Figure 7.4.: Performance difference between the two predictions MX and MO.

of H.263 and DVC. In both coding schemes, the RD performance of extrapolation and motion oracle differ by less than 1 dB.

In conclusion, the RD performance loss from motion oracle to extrapolation is significant for the Foreman sequence. Compared to H.263 inter, the RD performance loss can add up to 6 dB for the Foreman sequence, of which the extrapolation contributes 3 dB.

7.2.4. Discussion of observed RD performance losses

We analyzed the RD performance losses incurred by the 3 investigated DVC components. We observe that the losses incurred are not orthogonal, i.e. the components are not independent. We observe the dependency between components for instance in Figure 7.4.

For the Foreman sequence the ME/MC in DVC causes more than half the total RD performance loss of 6 dB to H.263 inter. Although the RD performance loss is also 6 dB for the Hall-monitor sequence, the ME/MC contributes less than 0.4 dB. Consequently, for the low motion Hall-monitor sequence fixed frame quantization and LDPC contribute almost 6 dB to the RD performance loss while contributing less than 3 dB to the RD performance loss for the Foreman sequence.
In conclusion, the RD performance loss per component varies with the video sequence properties. Nevertheless, the cumulative RD performance loss compared to conventional inter coding is in general large.

7.3. Trade-off between RD performance and encoding complexity

DVC coders were initially motivated by the low encoder complexity. Most state of the art encoder have added quite some complexity to the encoder compared to the initial Stanford DVC codec. As a consequence, the PSNR performance has gone up. There has been little study of the trade-off between DVC encoder complexity and performance. For that purpose we consider a benchmark, to review the matured DVC solutions and their status compared to state of the art conventional codecs.

The intention of such a benchmark is to provide a representative comparison of the participating systems. For conventional codecs, we consider H.263 and H.264/AVC. For DVC, we consider the latest DISCOVER codec, available at [1], and a system build from the components proposed in this thesis. The comparison then focuses on both RD performance and encoder complexity.

7.3.1. Considered complexity measure

While measuring the RD performance is simple, finding an objective complexity measure is not. Physical measures like run-time and memory requirements change with advances in computer technology. The encoding time is highly dependent on the used hardware and software platforms. The encoding time depends on the instruction set, microarchitecture details such as techniques to exploit instruction-level parallelism, pipeline depth, memory structure and bandwidth and the compiler optimization such as software pipelining and loopunrolling etc. [57].

A more standardized measure is the number of elementary computer operations it takes to solve the problem in the worst case [5]. In view of this particular benchmark however, it is difficult to obtain the number of operations. For some codecs only an executable is available, making it difficult to measure the number of operations reliably.

The considered DVC schemes rely on a feedback channel. Thus, there are at least two main problems for applying these codecs in practice. First, the decoder complexity in DVC is significant and real time decoding beyond todays technology. Second, even if the decoder complexity could be reduced, there is still the round trip delay to consider. While waiting for the feedback, the encoder would have to continuously buffer video data. Adding an encoder-based rate control, as considered in [29], can circumvent the problem. Yet, at the same time it increases encoder complexity and/or decreases the RD performance.

In view of these practical issues we follow the benchmark from [1]: we measure the complexity by means of the encoding time for the full sequence, in seconds, under controlled conditions. Even though simple, it is the first time such a comparison is done for a wide range of schemes.

7.3.2. Considered conventional video coding and DVC schemes

The schemes we compare are the following:

H.263 intra *I-I-I-I* Low complexity, straightforward intra coding. Expected to have both the lowest RD performance and one of the lowest complexities.

H.263 inter *I-P-P-P* Mid complexity, straightforward inter coding. Expected to have second highest RD performance and reasonable encoding complexity.

H.264 intra *I-I-I-I* Mid complexity, sophisticated intra coding. Expected to have robust RD performance and reasonable encoding complexity. Main difference towards H.263 intra coding is an improved VLC design with context adaptive VLC tables and variable macroblock sizes [102].

H.264 inter *I-P-P-P* High complexity, sophisticated inter coding. Expected to have best RD performance and highest encoding complexity. Main difference towards H.263 intra coding is an improved VLC design, variable macroblock sizes, higher accuracy during motion estimation and multi-frame motion compensation [102]. We use the Enhanced Predictive Zonal Search (EPZS) instead of full search for the motion estimation. While decreasing the RD performance slightly, the encoding complexity is reduced significantly.

DISCOVER codec *I-WZ-I-WZ* Low-Mid complexity, combining sophisticated H.264 intra coding and WZ coding. Expected to have best RD performance of DVC schemes and encoding complexity somewhat higher than half that of H.264 intra (due to GOP size of 2). It should be noted that we use a modified codec version which reduces the encoding time to less than 25 % of the original codec from [1]. The H.264 intra coding is similar to the **H.264 intra** scheme in this benchmark. Compared to the DISCOVER codec it uses a more recent intra codec and low complexity oriented configuration parameters, for instance baseline instead of main profile.

proposed DVC components oracle *I-WZ-WZ-WZ* Low complexity, combining straightforward H.263 intra and WZ coding. Expected to have reasonable RD performance and one of the lowest encoding complexities. We assume an oracle rate control at the encoder, i.e. it is not practical but provides a lower bound for encoding complexity. It only does the LDPC encoding once for the correct rate. RD performance will be denoted TU Delft jointly with proposed DVC components feedback as it is identical.

proposed DVC components feedback *I-WZ-WZ-WZ* Low complexity, straightforward H.263 intra and coding. Expected to have reasonable RD performance and low encoding complexity. The codec does not assume encoder rate control and encodes all possible codes in sequence (65) [8], then relying on a feedback channel to choose the right code. RD performance will be denoted TU Delft jointly with proposed DVC components oracle as it is identical.

7.3.3. Considered settings

The settings for the conventional codecs were chosen with emphasis on low encoding complexity. The detailed configuration settings can be found in the Appendix B. The specific algorithms in use are the following. For H.263 we use the tmn 3.2 implementation [3] on which also the proposed DVC components is based. For H.264 we use reference software JM 16.1 [4]. The DISCOVER codec originally employs reference software JM 9.5 in its framework but in our modification we use JM 16.1 instead. It should be noted, that all WZ codecs only encode the luminance.

We kept the hard- and software conditions constant over all schemes. For all results, presented in the following, we used a Linux 64 bit compute server with a quad core Intel Xeon 5160 at 3.0 GHz and 32GB of RAM. It should be noted that none of the codecs exploited multicore features. Still, a constrained device on which these codecs have to work in practice poses very different limitations than a compute server. Especially memory and buffer size can become a limiting factor. Consequently, the benchmark only provides a general feeling of encoder complexity. It does not address practical issues like for instance the question of how to simultaneously watch while recording on a hand held device.
The code was compiled with gcc-3.4.6-9-x86_64. Next to each sequential benchmark, no other processes were running on the compute server, giving the results comparative value. Nevertheless, some conditions like the amount of optimized code could not be taken into account. Optimized code can have a large impact on the encoding complexity. For instance Denolf *et al.* show optimized code that results in a speed up factor of 6.0 to 19.5 for a video decoder [43].

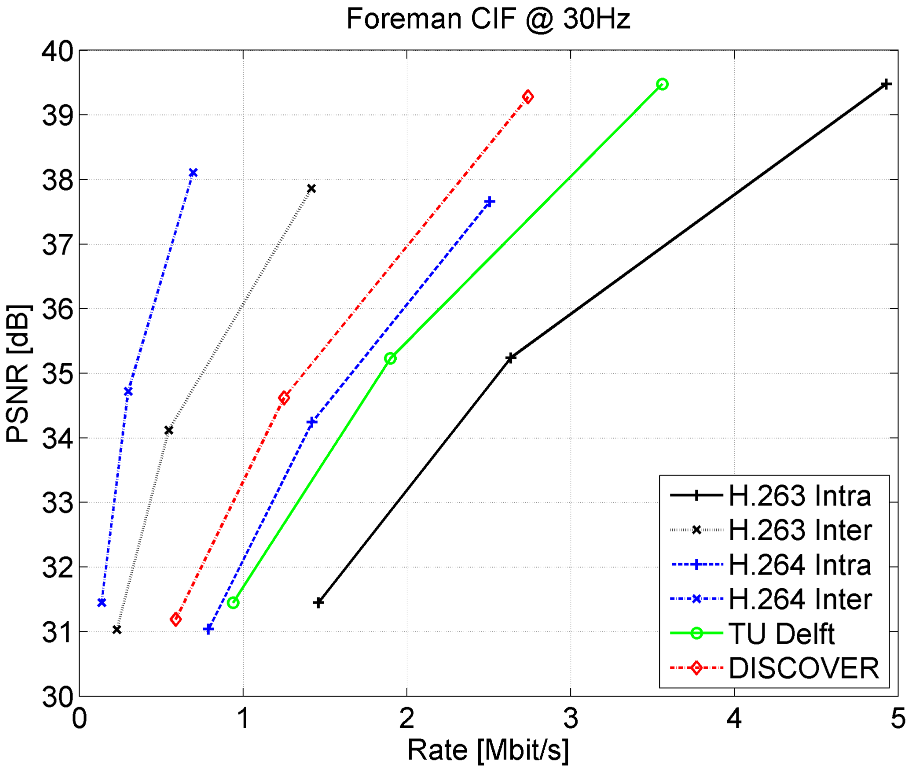
7.4. RD performance versus encoding complexity

7.4.1. RD performance comparison

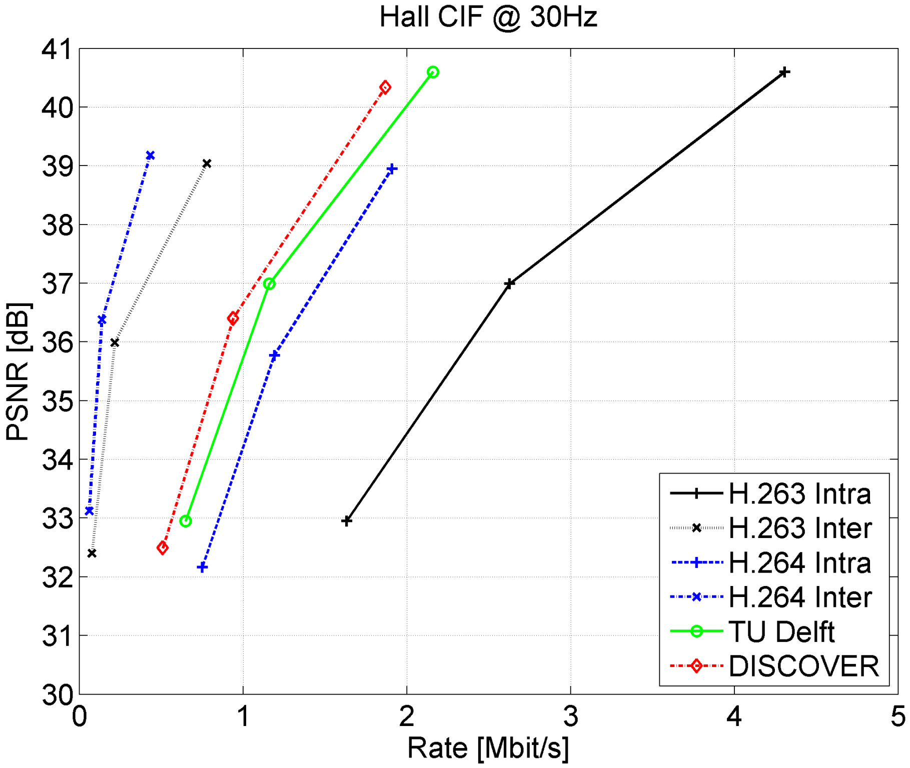
Figure 7.5 gives an overview of the RD performance for Foreman and Hall-monitor sequence. For both sequences, we find there is significant spatial correlation to be exploited. Consequently, the sophisticated intra coding in H.264 outperforms its H.263 counterpart by 3 to 4dB. For H.264 inter coding, the performance gain is 2dB over H.263 inter coding for the Foreman sequence. This gain is reduced to 1dB for the Hall-monitor sequence, where the motion is comparably small.

For the DVC codecs, also shown in Figure 7.5, the DISCOVER codec consistently outperforms the proposed DVC components by at least 1dB. The DISCOVER codec combines the higher side information quality of interpolation with a GOP size of two with a comparably low key frame cost due to efficient H.264 intra coding. For the Foreman sequence, H.264 intra alone also outperforms the proposed DVC components. Nevertheless, both conventional inter prediction schemes outperform the DISCOVER codec by 2 to 7 dB.

Figure 7.6 provides a similar overview of the RD performance for Coastguard and Stefan sequence. For these sequences, we find that exploiting the spatial correlation is far less efficient. Consequently, the gap of 2 dB between the two conventional intra coding schemes is not as pronounced as in Figure 7.5. In terms of temporal correlation,

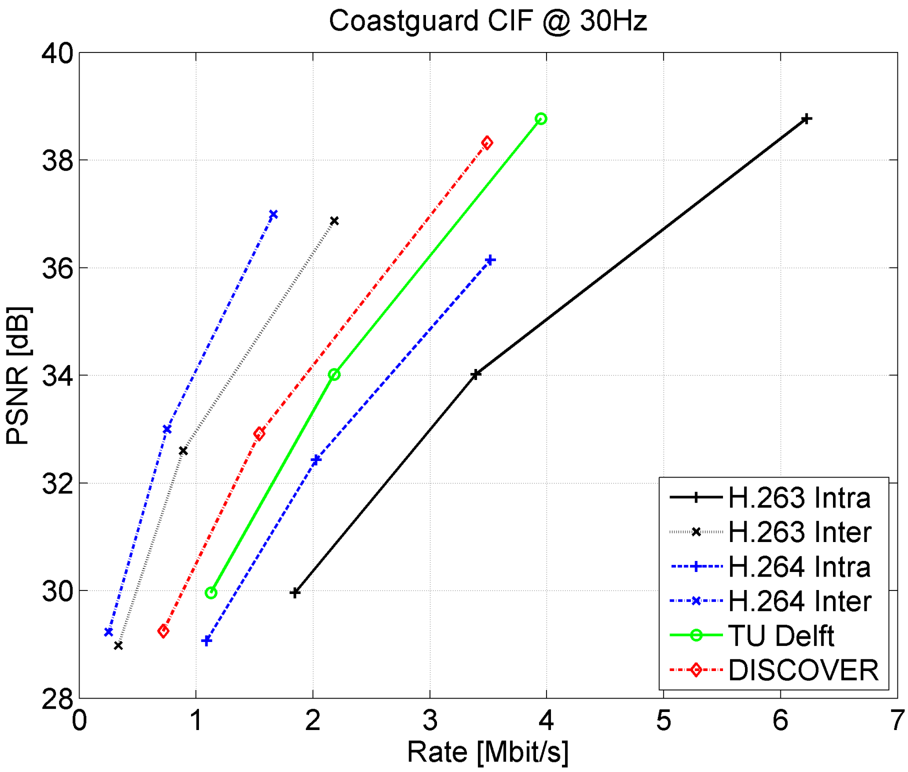


(a) Foreman

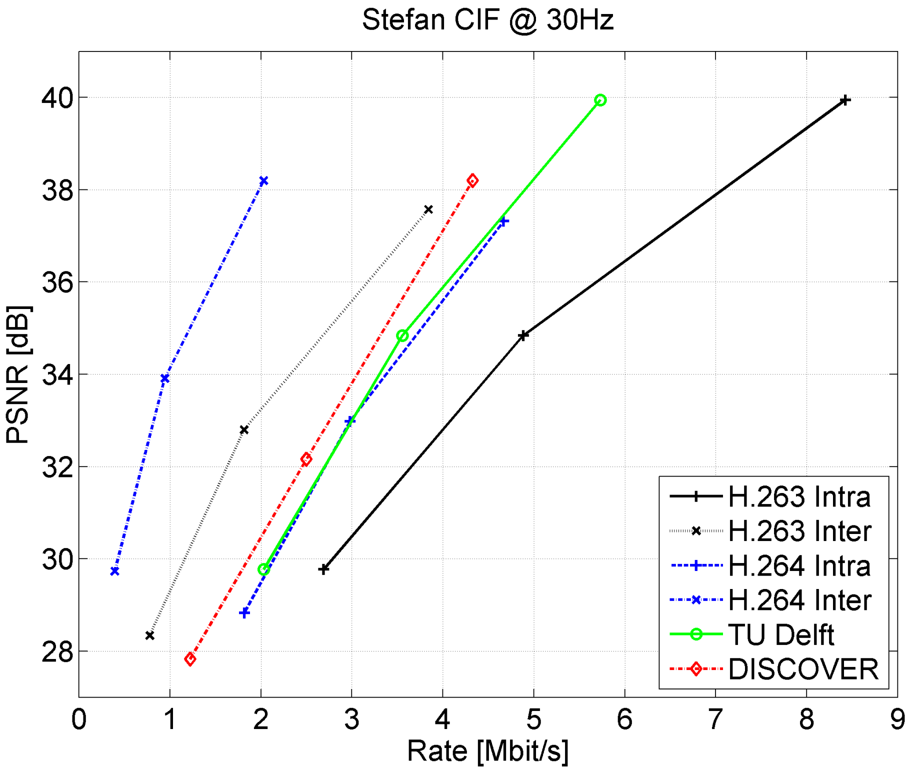


(b) Hall

Figure 7.5.: RD performance comparison of conventional and DVC coders for (a) Foreman and (b) Hall-monitor (fix legend H.).



(a) Coastguard



(b) Stefan

Figure 7.6.: RD performance comparison of conventional and DVC coders for (a) Coastguard and (b) Stefan.

we observe a gain of 5 dB when going from H.263 inter to H.264 inter for the large motion Stefan sequence. This gain is reduced to 1 to 2 dB for Coastguard.

For the WZ codecs, also shown in Figure 7.6, the DISCOVER codec still outperforms the proposed DVC components. However, the gap between the two DVC approaches is limited to about 1 dB in PSNR. The only exception is the Foreman sequence, where the gap between the proposed components and the DISCOVER codec accounts for 2 dB. Both DVC schemes are consistently outperformed by the conventional inter prediction schemes.

7.4.2. Encoding complexity comparison

The encoding complexity has been measured in seconds to encode the full CIF resolutions sequences, i.e. including all frames. Each sequence contains 300 frames. The investigated qualities correspond to the RD points in Section 7.4.1 from lowest PSNR at Quality 1 to highest PSNR at Quality 3. As the encoding times do not fluctuate significantly over the four sequences, Figure 7.7 provides the average encoding time over all 4 sequences. For the complexity per sequence we refer the reader to the Appendix C.

Ranging from lowest to highest complexity we observe the following in Figure 7.7:

proposed DVC components oracle This scheme needs least time of all schemes with an encoding time average of 3 seconds per sequence. All three blocks at the encoder are simple, which while reducing the RD performance increases the speed. Compared to H.263 intra, for which DCT transform and quantization are identical, the LDPC encoding for the correct rate is less complex than VLC coding. Due to the oracle rate control not a practically feasible scheme.

H.263 intra This conventional intra scheme needs an encoding time average of 5 seconds per sequence. Slightly more complex compared to the proposed DVC components oracle due to the VLC coding including rate control as opposed to LDPC with oracle rate.

proposed DVC components feedback This feedback-based DVC scheme needs an encoding time average of 14 seconds. The complexity from the components oracle is increased since LDPC coding is performed for all possible rates. Instead of only LDPC coding the correct rate, the LDPC coding is done for 65 possible rates.

H.263 inter This conventional inter scheme needs an encoding time average of 15 seconds per sequence. More complex than H.263 intra since ME/MC is performed at the encoder.

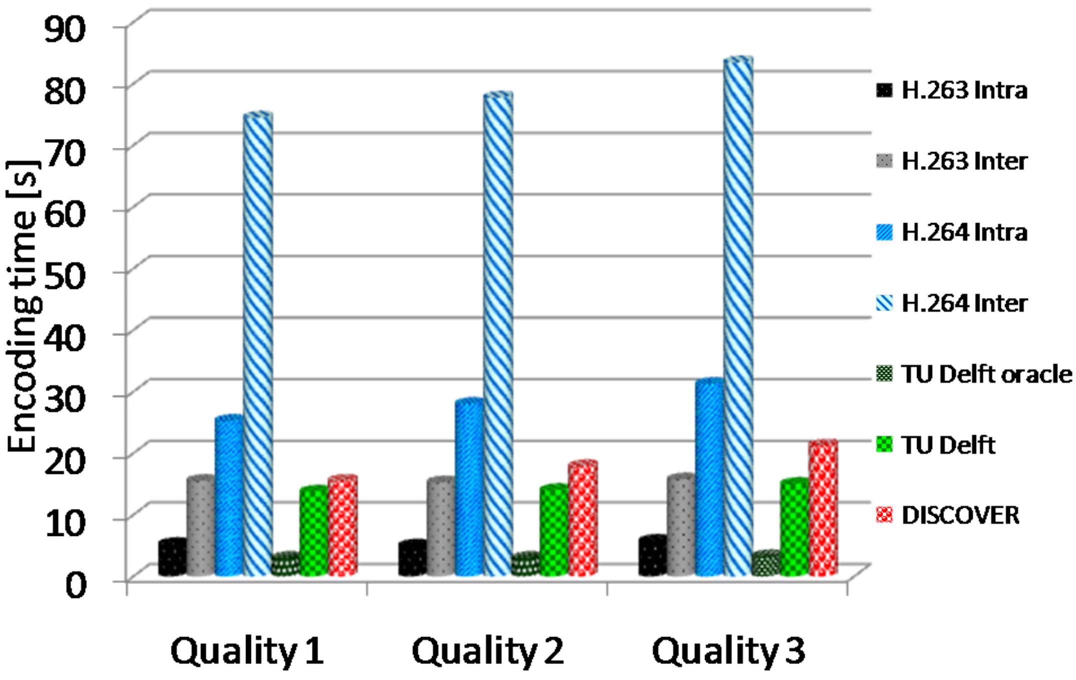


Figure 7.7.: Encoder complexity comparison of conventional and DVC coders, averaged over all 4 sequences.

DISCOVER WZ codec This feedback-based DVC scheme needs an encoding time average of 18 seconds. The coding of the WZ frames is faster than for the proposed DVC components. We expect code optimization to be responsible for the speed improvement. Most of the encoding time is spent on encoding the 50% intra coded key frames.

H.264 intra The conventional intra scheme needs an encoding time average of 28 seconds. Considerably more complex than H.263 intra due to the RD performance improvements introduced in Section 7.3.

H.264 inter The conventional inter scheme needs an encoding time average of 79 seconds. Considerably more complex than H.263 inter due to the RD performance improvements introduced in Section 7.3.

The proposed DVC components oracle scheme shows that the low complexity concept of DVC has merit by combining a better RD performance with lower complexity than intra coding. But in conclusion, the complexity increase from feedback-based DVC coding to H.263 inter coding is too small to outweigh the corresponding loss in RD performance of up to 7 dB.

7.5. Discussion

Non orthogonal performance losses observed
We have investigated the performance gap between DVC and conventional coding by analyzing three components, namely quantization, encoding and ME/MC. All three components incur losses dependent on the video content.

The three components respective contributions to the total performance loss varies with the video properties. For low motion sequences the first two components, quantization and LDPC coding dominate. For difficult sequences the motion estimation alone is responsible for more than half the loss.

DVC has a better RD performance vs encoding complexity trade-off than conventional intra coding
We provide a benchmark on RD performance and encoding complexity of conventional and DVC coders. The latter are able to outperform intra coding for most sequences, depending on the temporal correlation even by a large margin. At the same time, it is possible to keep the encoder complexity low.

To further improve the RD performance of DVC is most effectively tackled by increasing the encoding complexity. But considering the encoding complexity of benchmarked conventional coders there is not much room to increase the encoder complexity of DVC coders without getting in the complexity range of for instance H.263 inter.

Based on these findings, we conclude that DVC is only feasible where the encoding complexity is severely restricted and a feedback channel present.

8. Discussion

8.1. Summary of results

Distributed video coding is an approach to low complexity video encoding. For capturing video on constrained media devices the complexity is a strong limitation. The DVC components presented in this thesis have in common that they focus on a very low encoder complexity. The presented solutions keep the encoder complexity minimal and focus on improving the RD performance of DVC at the decoder side. The focus of this thesis is not on tuning and optimizing the proposed DVC components, but we are interested in their inherent performance limitations compared to conventional video coding.

The underlying theory of DVC states that there is no coding efficiency loss when performing independent encoding with side information under certain conditions. These conditions were later narrowed down to one condition, namely that the difference between reference frame X and side information Y is assumed to be of Gaussian distribution. However, this Gaussian assumption does not hold for video prediction errors.

In Chapter 3 we show, that the Gaussian distribution is not well suited to modeling the $X - Y$ difference. Consequently, we employ different distributions to model this difference, i.e. the virtual dependency channel. First, for symbol-based LDPC coding in the pixel domain we find the two-sided Gamma distribution to have the best general performance. Second, for bit plane-based LDPC coding in the transform domain we follow literature and use the Laplacian distribution. The latter approach provides a similar RD performance, but greatly decreases the decoder complexity and is thus our preferred choice.

The properties of the VDC depend on how the side information Y is generated and on how X is quantized. The side information Y is generated by combining motion estimation and compensation. In Chapter 4 we find, that the combination of true motion estimation and motion compensated extrapolation outperforms minimum residue ME and motion compensated interpolation in terms of RD performance. However, without access to the reference frame X during motion estimation, our proposed DVC solution loses RD performance compared to conventional video coding.

The quantization in DVC is similarly impaired without access to side information Y at the encoder. In addition, the fixed-rate LDPC encoder limits the adaptivity of quantization. In Chapter 5 we propose a DVC quantization scheme that combines limited adaptivity with a small overhead. Further, we propose a side information update scheme based on the iterative decoding of DCT coefficients. With partial access to reference frame X at the decoder, a more reliable side information can be generated. However, the impact on the RD performance is small and highly sequence dependent.

In Chapter 6 we revisit the VDC modeling. Here, we aim to improve the LDPC decoding by taking reliability information of the side information Y into account. The video prediction errors in Y and hence the VDC properties are non-stationary. By means of a classification into a reliable and an unreliable class we take the non-stationarity into account. With an oracle classification we verify the non-stationarity and observe a significant RD performance increase compared to a stationary VDC model. But given misclassification, the RD performance degrades rapidly. The proposed practical classification schemes are not able to solve the classification problem accurately enough.

8.2. DVC versus conventional video coding

The two primary options for capturing video on constrained media devices are to either make DVC schemes competitive in terms of RD performance or to reduce the encoding complexity of conventional video coding. Conventional video codecs are well established in practice and have mature solutions for each component. By contrast, DVC is still in the research phase and induces drawbacks to practical deployment, e.g. the often assumed availability of a feedback channel.

From Chapters 3, 4 and 5 we may generalize, that each investigated DVC component incurs a RD performance loss when compared to conventional video coding. For the proposed low complexity DVC components we analyze this RD performance gap between DVC and conventional video coding in Chapter 7. From literature it is known that the RD performance losses in DVC can most effectively be tackled at the encoder side. The trade-off between RD performance and encoder complexity is then discussed as final question in Chapter 7.

When referring to the RD performance of conventional video coding, we refer to inter coding. With intra coding there is a very low complexity alternative. At the cost of RD performance, the encoder complexity is significantly reduced. We show that with a similar encoding complexity DVC can provide a better RD performance than conventional intra coding schemes. However, the best overall trade-off between encoding complexity and RD performance is provided by the well established conventional H.263 inter codec.

Considering the encoding complexity of benchmarked conventional coders in Chapter 7, there is not much room for increasing the encoder complexity of DVC coders without getting in the complexity range of for instance H.263 inter.

8.3. Outlook

Low complexity encoding of video will still be relevant in the future. Although the definition of what is low complexity might change with advances in processing power, i.e. hardware design, the advances in battery capacity are slow. Hence, there is still a need for efficient algorithms to increase battery lifetime. We expect the solution for consumer video encoding to be found in optimized conventional video coders or their hardware implementations rather than in DVC.

In the upcoming years, we then expect DVC research to move to more specialized application fields focusing on extremely low complexity. Special attention might be paid to inherently distributed systems like wireless sensor networks. Here, DVC may provide a successful best effort scheme. In addition DVC might provide further insights into related fields by means of its components. In this thesis we provided insights into channel coding, ME/MC and quantization, all of which are relevant in many application fields.

To make DVC competitive to conventional video coders there are still unsolved problems. Below we list a few of these.

1. While the presence of a feedback channel is implicitly assumed in this thesis, for practical deployment such a feedback channel is not feasible. Consequently, a low complexity encoder based rate control is necessary. We do not see a solution to this problem without either decreasing the RD performance as reported in [29] or increasing the encoder complexity to at least enable frame differencing [29, 59].

2. A further increase in the adaptivity of the DCT quantization while keeping the overhead low would be beneficial. A solution to this problem may be found by considering existing literature. Rateless LDPC codes [50, 59] remove the need for fixed length input to the LDPC decoder. Consequently, adaptive run length and amplitude pairs similar to conventional video coding may be used.

3. Especially the quality of the initial side information is crucial to the RD performance. Hence more sophisticated motion estimation schemes, for instance optical flow-based, may be applied. Optical flow can be thought of as close to true motion [94].

In addition more research on extrapolation is required, especially occlusion handling. In the context of DVC a soft probability based hole filling may be beneficial.

4. To better take non-stationarity into account, we see two possible directions for further research. The first direction is towards better classification at the decoder by for instance pattern recognition [44]. The second direction is towards a more robust alternative to LDPC coding. Such an alternative would have an impact far wider than just on DVC. In this context, compressive sensing [32] may turn out to be a promising research field.

A. Derived probabilities for bit plane-based dependency models

To implement dependency models in a practical LDPC coder, we need estimates of the probabilities $P(Q|Y)$, $P(Q^b|Y^b)$, $P(Q^b|Y)$ and $P(Q^b|Y, Q^{b+1}, ..., Q^{L-1})$. Here, b denotes the current bit plane and L the total number of bit planes. The probabilities have been derived in [97] and are listed below:

$$P[Q^{(b)} = b_q|Y^{(b)} = b_q] = \sum_{\forall Y} P[Q^{(b)} = b_q|Y = y, Y^{(b)}] \cdot P[Y = y|Y^{(b)} = b_q], \qquad \text{(A.1)}$$

where $P[Y = y|Y^{(b)} = b_q]$ is 0 or 1 depending on whether the bit plane of Y equals b_q.

$$P[Q^{(b)} = b_q|Y = y, Y^{(0)} = b_q] = \sum_{m=-q^-}^{q^+} \sum_{n=0}^{r_{max}} P(m \cdot d + (n - r(y, 2^b))), \qquad \text{(A.2)}$$

with $q^- = q(y, 2^{(b+1)})$, $q^+ = q((2^L - 1) - y, 2^{(b+1)})$, $r_{max} = 2^b - 1$, $d = 2^{b+1}$ and where the functions $r(a, b)$ and $q(a, b)$ are the remainder and quotient of the division between two integer values a and b.

$$P(Q^{(b)} = 0|Y = y, Q^{(b+1)}, ..., Q^{(L-1)}) = \sum_{i=0}^{2^b-1} P_N(q(x_p, 2^{b+1}) \cdot 2^{b+1} + i - y), \qquad \text{(A.3)}$$

with $x_p = \sum_{i=b+1}^{L-1} Q^{(i)} \cdot 2^i$.

B. Configuration settings for benchmark

B.1. H.263 tmn

We ran the tmn encoder with the following settings/command line options:

H.263 intra: tmn -i foreman_cif.yuv -x 3 -b 299 -k 0 -c 0 -q 16 -A 16 -f 1 -w -o foreman_QP16.yuv
Quality 1: q=A=16
Quality 2: q=A=8
Quality 3: q=A=4

H.263 inter: tmn -i foreman_cif.yuv -x 3 -b 299 -k 0 -c 0 -q 16 -A 16 -f 0 -w -o foreman_QP15.yuv
Quality 1: q=A=16
Quality 2: q=A=8
Quality 3: q=A=4

B.2. H.264 JM 16.1

To consider a low complexity encoder profile we changed the following parameters from the default encoder.cfg:

```
#################################################################
# Files
#################################################################


InputFile1     = "foreman_cif.yuv"       # Input sequence

SourceWidth    = 352    # Source frame width
SourceHeight   = 288    # Source frame height

OutputWidth    = 352    # Output frame width
```

```
OutputHeight   = 288    # Output frame height

################################################################
# Encoder Control
################################################################

ProfileIDC     = 66 # Profile IDC (66=baseline; FREXT: 100=High)

QPISlice       = 26  # Quant. param for I Slices (0-51)
QPPSlice       = 26 # Quant. param for P Slices (0-51)

QPISlice       = 26  # Quant. param for I Slices (0-51)
QPPSlice       = 26  # Quant. param for P Slices (0-51)

SearchRange    = 16  # Max search range

NumberReferenceFrames = 2   # Number of previous frames
used for inter motion search (0-16)

################################################################
# Output Control, NALs
################################################################

SymbolMode     =  0  # Symbol mode (Entropy coding method: 0=UVLC, 1=CABAC)

################################################################
#Fast Motion Estimation Control Parameters
################################################################

SearchMode     = 3    # Motion estimation mode
                      # -1 = Full Search
                      #  0 = Fast Full Search (default)
                      #  1 = UMHexagon Search
                      #  2 = Simplified UMHexagon Search
                      #  3 = Enhanced Predictive Zonal Search (EPZS)
```

We then considered the following quality settings:
H.264 intra:

```
IntraPeriod     = 1   # Period of I-pictures   (0=only first)
```

Quality 1:

```
QPISlice              = 38  # Quant. param for I Slices (0-51)
```

Quality 2:

```
QPISlice              = 33  # Quant. param for I Slices (0-51)
```

Quality 3:

```
QPISlice              = 28  # Quant. param for I Slices (0-51)
```

H.264 inter:

```
IntraPeriod     = 0   # Period of I-pictures   (0=only first)
```

Quality 1:

```
QPISlice              = 36  # Quant. param for I Slices (0-51)
QPPSlice              = 36  # Quant. param for P Slices (0-51)
```

Quality 2:

```
QPISlice              = 31  # Quant. param for I Slices (0-51)
QPPSlice              = 31  # Quant. param for P Slices (0-51)
```

Quality 3:

```
QPISlice              = 26  # Quant. param for I Slices (0-51)
QPPSlice              = 26  # Quant. param for P Slices (0-51)
```

B.3. DISCOVER codec

The H.264 codec used in our modified version uses configuration identical to the H.264 encoder profile described in Section B.2.
The DISCOVER encoder configuration was the following:

```
# DISCOVER-encoder configuration
# <ParameterName> = <ParameterValue>
# All non integer values must be contained within quotation

##############################################################################
# Files
##############################################################################
InputFile         = "foreman_cif.yuv" # Input sequence, YUV 4:2:0
```

```
FramesToBeEncoded = 299 # Number of frames (WZ and Intra) to be encoded
SequenceSize      = "CIF" # QCIF: 176x144; CIF: 352x288
KeyFrameInfo      = "foreman_keyframe.cfg" # Key frame (intra) configuration file
WZBitstreamFile   = "bitstreamQ6.wz" # Bitstream file

##############################################################################
# Encoder Control
##############################################################################
IntraPeriod       =  2  # Period of I-frames (it must be > 1)
QIndex            =  6  # Quantisation index (Range between 1 and 8)
AdaptiveGOP       =  0  # 0 - off, 1 - on
```

We then considered the following quality settings:
DISCOVER codec:
Quality 1: QIndex = 3
Quality 2: QIndex = 6
Quality 3: QIndex = 8

The DISCOVER decoder configuration was the following:

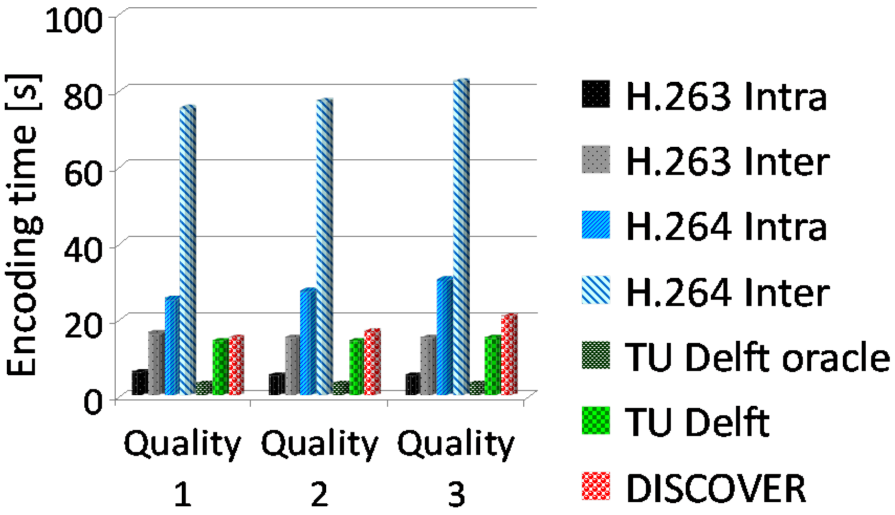
```
# DISCOVER-encoder configuration
# <ParameterName> = <ParameterValue>
# All non integer values must be contained within quotation

###############################################################################
# Files
###############################################################################
OriginalSequence = "foreman_cif.yuv" # Original sequence (used for PSNR calculation
WZBitstream      = "bitstreamQ8.wz" # Bitstream file
OutputSequences  = "foreman_decodedQ8"   # Decoded file (without extension)
PSNRFile         = "foreman_psnr_data_Q8a.txt" # PSNR trace file
KeyFrameInfo     = "foreman_keyframe.cfg" # Key frame (intra) configuration file
FrameRate  = 30 # Frame rate (used for rate calculation)
```

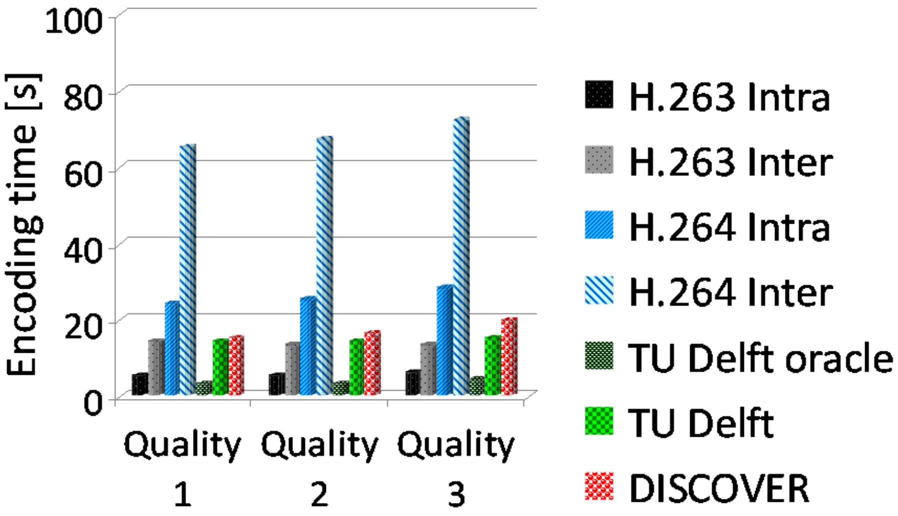
To run the DISCOVER codec with JM 16.1, it is necessary to generate the PSNR trace file foreman_psnr_data_Q8a.txt, i.e. we modified the JM 16.1 code to do so.

C. Encoding time benchmark results per sequence

As the encoding times do not fluctuate significantly over the four sequences, Section 7.4.2 provides the average encoding time over all 4 video sequences (Foreman, Hall-monitor, Coastguard, Stefan). The complexity per sequence is presented in the following.

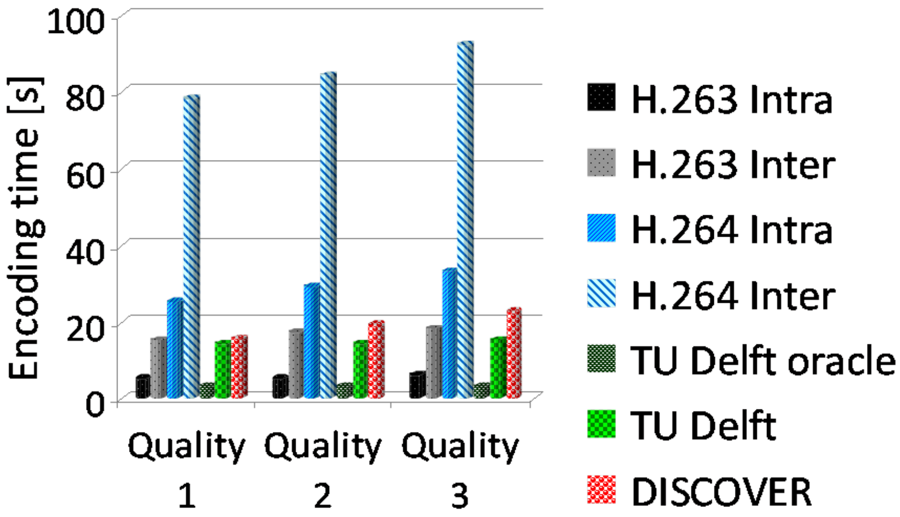


(a) Foreman

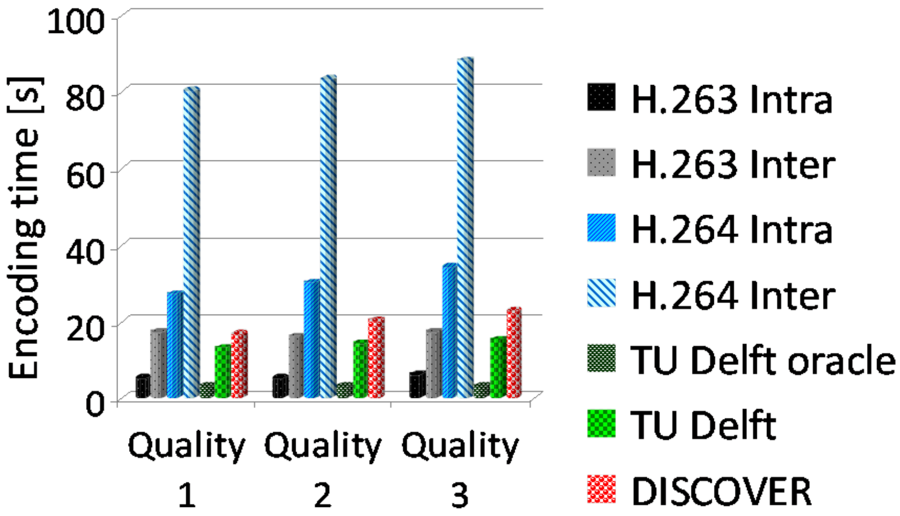


(b) Hall-monitor

Figure C.1.: Encoder complexity comparison of conventional and DVC coders for (a) Foreman and (b) Hall monitor.



(a) Coastguard



(b) Stefan

Figure C.2.: Encoder complexity comparison of conventional and DVC coders for (a) Coastguard and (b) Stefan.

Bibliography

[1] Discover project page. http://www.discoverdvc.org/.

[2] Discover test conditions. http://www.img.lx.it.pt/~discover/test_conditions.html.

[3] H.263+ reference software tmn 3.2.

[4] http://iphome.hhi.de/suehring/tml/.

[5] http://users.forthnet.gr/ath/kimon/cc/ccc1b.htm.

[6] http://www.discoverdvc.org/deliverables/discover-d19.pdf.

[7] http://www.faqs.org/patents/app/20090161010.

[8] http://www.stanford.edu/ divad/software.html.

[9] http://www.stanford.edu/ dmchen/dvc.html.

[10] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proceedings IEEE Data Compression Conference*, pages 252–261, April 2002.

[11] A. Aaron and B. Girod. Wyner-ziv video coding with low encoder complexity. In *Proceedings International Picture Coding Symposium, PCSÕ04, invited paper*, San Francisco,CA, December 2004.

[12] A. Aaron, S. Rane, E. Setton, and B. Girod. Transform-domain wyner-ziv codec for video. In *Proceedings Visual Communications and Image Processing*, San Jose,CA, January 2004.

[13] A. Aaron, S. Rane, R. Zhang, and B. Girod. Wyner-ziv coding for video: applications to compression and error resilience. In *Proc. of the Data Compression Conference (DCC 2003)*, pages 93–102, March 2003.

[14] A. Aaron, D. Varodayan, and B. Girod. Wyner-ziv residual coding of video. In *Proceedings Picture Coding Symposium*, Beijing, China, April 2006.

[15] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The discover codec: Architecture, techniques and evaluation. In *26th Picture Coding System*, Lisbon, Portugal, November 2007.

[16] X. Artigas, S. Malinowski, C. Guillemot, and L. Torres. Overlapped quasi-arithmetic codes for distributed video coding. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pages II –9–II –12, 16 2007-Oct. 19 2007.

[17] J. Ascenso, C. Brites, and F. Pereira. Content adaptive wyner-ziv video coding driven by motion activity. In *IEEE International Conference on Image Processing*, October 2006.

[18] A. Avudainayagam, J.M. Shea, and D.P. Wu. Hyper-trellis decoding of pixel-domain wyner-ziv video coding. 18(5):557–568, May 2008.

[19] Yukihiro Bandoh, Kazuya Hayase, Seishi Takamura, Kazuto Kamikura, and Yoshiyuki Yashima. Mode decision for h.264/avc based on spatio-temporal sensitivity. In *26th Picture Coding System*, Lisbon, Portugal, November 2007.

[20] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti. Statistical analysis of the 2d-dct coefficients of the differential signal for images. 4(6):477–488, November 1992.

[21] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. A soft-input soft-output maximum a posteriori (map) module to decode parallel and serial concatenated codes. In *TDA Progress Report*, volume 42, pages 1–20, November 1996.

[22] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070 vol.2, May 1993.

[23] M. Bhaskaranand and J.D. Gibson. Distributions of 3d dct coefficients for video. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 793–796, April 2009.

[24] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. Improving motion compensated extrapolation for distributed video coding. In *Thirteenth Annual Conference of the Advanced School for Computing and Imaging*, pages 291–297, June 2007.

[25] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. On extrapolating side information in distributed video coding. In *26th Picture Coding System*, Lisbon, Portugal, November 2007.

[26] S. Borchert, R.P. Westerlaken, R.K. Gunnewiek, and R.L. Lagendijk. Motion compensated prediction in transform domain distributed video coding. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 332–336, Oct. 2008.

[27] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. On the generation of side information for dvc. In *Twenty-eigth Symposium on Infomation Theory in the Benelux*, pages 1141–1148, May 2007.

[28] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. Analysis of performance losses in distributed video coding. In *Picture Coding Symposium*, Chicago, Illinois, USA, May 2009.

[29] C. Brites and F. Pereira. Encoder rate control for transform domain wyner-ziv video coding. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pages II –5–II –8, 16 2007-Oct. 19 2007.

[30] Catarina Brites and Fernando Pereira. Correlation noise modeling for efficient pixel and transform domain wyner-ziv video coding. *IEEE Trans. Circuits Syst. Video Techn.*, 18(9):1177–1190, 2008.

[31] J. Ascenso C. Brites and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, July 2005.

[32] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21 –30, march 2008.

[33] J. Chen and M.P.C. Fossorier. Near optimum universal belief propagation based decoding of low density parity check codes. *IEEE Transactions on Communications*, 50(3):406–414, March 2002.

[34] W.H. Chen and W.K. Pratt. Scene adaptive coder. 32:225–232, 1984.

[35] Yen-Kuang Chen. True motion estimation - theory, application, and implementation, 1998.

[36] Yen-Kuang Chen, A. Vetro, Huifang Sun, and S.Y. Kung. Frame-rate upconversion using transmitted true motion vectors. In *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pages 622–627, Dec 1998.

[37] W.J. Chien, L.J. Karam, and G.P. Abousleman. Distributed video coding with 3d recursive search block matching. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, page 4, May 2006.

[38] M. Dalai, R. Leonardi, and F. Pereira. Improving turbo codec integration in pixel-domain distributed video coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 537–540, Toulouse, France, 14-19 May 2006.

[39] G. Dane and T.Q. Nguyen. Motion vector processing for frame rate up conversion. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–309–12 vol.3, May 2004.

[40] M.C. Davey and D. MacKay. Low density parity check codes over gf(q). *IEEE Communication Letters*, 2(6):165–167, June 1998.

[41] G. de Haan, P.W.A.C. Biezen, H. Huijgen, and O.A. Ojo. True-motion estimation with 3-d recursive search block matching. *Circuits and Systems for Video Technology, IEEE Transactions on*, 3(5):368–379, 388, Oct 1993.

[42] N. Deligiannis, A. Munteanu, T. Clerckx, J. Cornelis, and P. Schelkens. On the side-information dependency of the temporal correlation in wyner-ziv video coding. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 709–712, April 2009.

[43] Kristof Denolf, Peter Vos, Jan Bormans, and Ivo Bolsens. Cost-efficient c-level design of an mpeg-4 video decoder. In *PATMOS '00: Proceedings of the 10th International Workshop on Integrated Circuit Design, Power and Timing Modeling, Optimization and Simulation*, pages 233–242, London, UK, 2000. Springer-Verlag.

[44] Robert P. W. Duin and Elċbieta P Ekalska. The science of pattern recognition. achievements and perspectives. In *in Challenges for Computational Intelligence, Studies in Computational Intelligence Series*, pages 221–259, 2007.

[45] A. Ephremides. Energy concerns in wireless networks. *Wireless Communications, IEEE*, 9(4):48–59, Aug. 2002.

[46] R.G. Gallager. Low density parity check codes. *IRE Trans.Inform.Theory*, IT-8:21–28, January 1962.

[47] B. Girod, A.M. Aaron, S. Rane, and D. Rebello-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, January 2005.

[48] Song Guo and Oliver W.W. Yang. Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. *Computer Communications*, 30(9):2129 – 2148, 2007.

[49] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(10):1025–1039, Oct 1998.

[50] D.-k. He, A. Jagmohan, L. Lu, and V. Sheinin. Wyner-Ziv video compression using rateless LDPC codes. In *Proc. Visual Communications and Image Processing*, San Jose, CA, USA, 2008.

[51] Shih-Yu Huang, Jin-Rong Chen, Jia-Shung Wang, Kuen-Rong Hsieh, and Hong-Yih Hsieh. Classified variable block size motion estimation algorithm for image sequence coding. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 3, pages 736–740 vol.3, Nov 1994.

[52] P. Ishwar, V. Prabhakaran, and K. Ramchandran. Towards a theory for video coding using distributed compression principles. In *Proc. International Conference on Image Processing*, volume 3, pages 687–690, September 2003.

[53] R.L. Joshi and T.R. Fischer. Comparison of generalized gaussian and laplacian modeling in dct image coding. *Signal Processing Letters, IEEE*, 2(5):81–82, May 1995.

[54] Bonghoe Kim and Hwang Soo Lee. Reduction of the number of iterations in turbo decoding using extrinsic information. In *TENCON 99. Proceedings of the IEEE Region 10 Conference*, volume 1, pages 494–497 vol.1, 1999.

[55] Ouyang Kun, Ouyang Qing, Zhou Zhengda, and Li Zhitang. Fast motion estimation for real time h.264 video encoder. In *MultiMedia and Information Technology, 2008. MMIT '08. International Conference on*, pages 310–313, Dec. 2008.

[56] E.Y. Lam and J.W. Goodman. A mathematical analysis of the dct coefficient distributions for images. *Image Processing, IEEE Transactions on*, 9(10):1661–1666, Oct 2000.

[57] G. Landge, M. van der Schaar, and V. Akella. Complexity metric driven energy optimization framework for implementing mpeg-21 scalable video decoders. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages ii/1141–ii/1144 Vol. 2, March 2005.

[58] Z. Li, L. Liu, and E.J. Delp. Rate distortion analysis of motion side estimation in wyner–ziv video coding. *IEEE Transaction on Image Processing*, 16(1):98–113, 2007.

[59] Limin Liu, Dake He, Ashish Jagmohan, Ligang Lu, and Edward J. Delp. A low-complexity iterative mode selection algorithm for wyner-ziv video compression. In *IEEE International Conference on Image Processing*, September 2008.

[60] D. J. C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.

[61] David J.C. MacKay and Radford M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32:1645–1646, 1996.

[62] J.L. Martinez, G. Fernandez Escribano, H. Kalva, W.A.R.J. Weerakkody, W.A.C. Fernando, and A. Garrido. Feedback free dvc architecture using machine learning. pages 1140–1143, 2008.

[63] P.F.A. Meyer, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. Distributed source coding of video with non-stationary side-information. In *Proc. Visual Communications and Image Processing (VCIP)*, July 2005.

[64] F. Muller. Distribution shape of two-dimensional dct coefficients of natural images. *Electronics Letters*, 29(22):1935–1936, Oct. 1993.

[65] F. Muller. On the motion compensated prediction error using true motion fields. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 3, pages 781–785 vol.3, Nov 1994.

[66] Stefaan Mys, Jürgen Slowack, Jozef Škorupa, Peter Lambert, and Rik Van de Walle. Introducing skip mode in distributed video coding. *Image Commun.*, 24(3):200–213, 2009.

[67] L. Natário, C. Brites, J. Ascenso, and F. Pereira. Extrapolating side information for low-delay pixel-domain distributed video coding. In *International Workshop on Very Low Bitrate Video Coding*, Sardinia, Italy, September 2005.

[68] Y. Nimmagadda, K. Kumar, and Yung-Hsiang Lu. Energy-efficient image compression in mobile devices for wireless transmission. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1278–1281, 28 2009-July 3 2009.

[69] EePing Ong, Hua Wang, and Ping Xue. Video coding based on true motion estimation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 3, pages III–409–12 vol.3, April 2003.

[70] F. Pereira. Distributed video coding: Basics, main solutions and trends. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1592–1595, 28 2009-July 3 2009.

[71] Fernando Pereira, Catarina Brites, and João Ascenso. Distributed video coding: Basics, codecs, and performance. In *Distributed Source Coding*, pages 189 – 245. Academic Press, Boston, 2009.

[72] S.S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding and its extension to the side information case. *Information Theory, IEEE Transactions on*, 49(5):1181–1203, May 2003.

[73] John G. Proakis. *Digital communications / John G. Proakis.* McGraw-Hill, 2001.

[74] R. Puri, A. Majumdar, and K. Ramchandran. Prism: A video coding paradigm with motion estimation at the decoder. *IEEE Transactions on Image Processing*, 16:2436–2448, October 2007.

[75] R. Puri and K. Ramchandran. Prism: an uplink-friendly multimedia coding paradigm. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2003 (ICASSP '03)*, volume 4, pages 856–859, April 2003.

[76] G. de Haan R.A. Braspenning. Efficient motion estimation with content-adaptive resolution. In *Proceedings of ISCE 2002*, pages E29–E–34, September 2002.

[77] G. de Haan R.A. Braspenning. True-motion estimation using feature correspondences. *Security, Steganography, and Watermarking of Multimedia Contents VI. Proceedings of the SPIE*, 5308:396–407, January 2004.

[78] G.N. Rao, R.S.V. Prasad, D.J. Chandra, and S. Narayanan. Real-time software implementation of h.264 baseline profile video encoder for mobile and handheld devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V, May 2006.

[79] R. Reininger and J. Gibson. Distributions of the two-dimensional dct coefficients for images. *Communications, IEEE Transactions on*, 31(6):835–839, Jun 1983.

[80] S. Sánchez, S. Borchert, R.P. Westerlaken, and R.L. Lagendijk. Non-stationary channel model based on unsupervised motion learning in distributed video coding. In *30-th Symposium on Information Theory in the Benelux*, Eindhoven, The Netherlands, May 2009.

[81] Khalid Sayood. *Introduction to Data Compression.* Morgan Kaufmann, 1996. pp 389-399.

[82] Khalid Sayood. *Introduction to Data Compression.* Morgan Kaufmann, 1996. pp 182.

[83] Khalid Sayood. *Introduction to Data Compression.* Morgan Kaufmann, 1996. pp 384-385.

[84] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[85] B. Sklar and F.J. Harris. The abcs of linear block codes. *Signal Processing Magazine, IEEE*, 21(4):14–35, July 2004.

[86] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

[87] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira. Exploiting spatial redundancy in pixel domain wyner-ziv video coding. In *IEEE International Conference on Image Processing*, pages 253–256, Atlanta,GA,USA, October 2006.

[88] M. Tagliasacchi, S. Tubaro, and A. Sarti. On the modeling of motion in wyner-ziv video coding. In *IEEE International Conference on Image Processing*, pages 593–596, Atlanta,GA,USA, October 2006.

[89] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. pages 839–846, 1998.

[90] Claudia Tonoli, Pierangelo Migliorati, and Riccardo Leonardi. Error resilience in current distributed video coding architectures. *J. Image Video Process.*, 2009:1–18, 2009.

[91] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive distributed source coding using low-density parity-check codes. In *Proceedings Asilomar Conference on Signals, Systems, and Computers*, pages 1203–1207, October 2005.

[92] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive codes for distributed source coding. *EURASIP Signal Processing Journal, Special Section on Distributed Source Coding*, 86(11):3123–3130, November 2006.

[93] David Varodayan, David Chen, Markus Flierl, and Bernd Girod. Wyner-ziv coding of video with unsupervised motion vector learning. *Image Commun.*, 23(5):369–378, 2008.

[94] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:490–498, 1989.

[95] Branka. Vucetic and Jinhong Yuan. *Turbo codes : principles and applications / Branka Vucetic, Jinhong Yuan.* Kluwer Academic, Boston ; London :, 2000.

[96] Zhenyu Wei, Kai Lam Tang, and K.N. Ngan. Implementation of h.264 on mobile device. *Consumer Electronics, IEEE Transactions on*, 53(3):1109–1116, Aug. 2007.

[97] R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, and R.L. Lagendijk. Analyzing symbol and bit plane-based ldpc in distributed video coding. In *Proceedings 2007 IEEE International Conference on Image Processing, ICIP '07*, pages II 17–20, San Antonio, USA, September 2007.

[98] R.P. Westerlaken, S. Borchert, R. Klein Gunnewiek, and R.L. Lagendijk. Dependency channel modeling for a ldpc-based wyner-ziv video compression scheme. In *Proceedings 2006 IEEE International Conference on Image Processing, ICIP '06*, pages 277–280, Atlanta, USA, October 2006.

[99] R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk. The role of the virtual channel in distributed source coding of video. In *IEEE International Conference on Image Processing*, volume 1, pages 581–584, September 2005.

[100] R.P. Westerlaken and R.L. Lagendijk. Towards h.263-like wyner-ziv coding of video. In *Preprint submitted to Elsevier*.

[101] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan. Rate-constrained coder control and comparison of video coding standards. 13(7):688–703, July 2003.

[102] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions On Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

[103] Wikipedia. Additive white gaussian noise — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Additive_white_Gaussian_noise&oldid=319441033.

[104] Wikipedia. Bch code — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=BCH_code&oldid=322126054.

[105] Wikipedia. Binary symmetric channel — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Binary_symmetric_channel&oldid=318328598.

[106] Wikipedia. Channel (communications) — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Channel_(communications)&oldid=322938495.

[107] Wikipedia. H.264/mpeg-4 avc — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=H.264/MPEG-4_AVC&oldid=306298035.

[108] Wikipedia. Low-density parity-check code — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Low-density_parity-check_code&oldid=322464507.

[109] Wikipedia. Probability density function — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Probability_density_function&oldid=334230566.

[110] Wikipedia. Turbo code — wikipedia, the free encyclopedia, 2009. http://en.wikipedia.org/w/index.php?title=Turbo_code&oldid=323217191.

[111] Wikipedia. Mpeg-2 — wikipedia, the free encyclopedia, 2010. http://en.wikipedia.org/w/index.php?title=MPEG-2&oldid=343208792.

[112] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[113] Shuiming Ye, M. Ouaret, F. Dufaux, and T. Ebrahimi. Improved side information generation with iterative decoding and frame interpolation for distributed video coding. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2228–2231, Oct. 2008.

[114] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[115] G.S. Yovanof and S. Liu. Statistical analysis of the dct coefficients and their quantization error. In *Signals, Systems and Computers, 1996. 1996 Conference Record of the Thirtieth Asilomar Conference on*, volume 1, pages 601–605 vol.1, Nov 1996.

Summary

The main focus of video encoding in the past twenty years has been on video broadcasting. A video is captured and encoded by professional equipment and then watched on varying consumer devices. Consequently, the main objective was to increase the compression efficiency and to decrease the decoder complexity. More recently, we observe a shift in user behavior, from solely consuming video to also producing and sharing video. For video encoding with constrained media devices instead of professional cameras, the encoder complexity becomes an important limitation.

To greatly reduce the encoder complexity it is possible to employ intra coding, that is to ignore the motion and encode images independently. Since temporal information is no longer taken into account, the problem with intra coding is the limited compression efficiency. This thesis addresses Distributed Video Coding (DVC) as a possible solution for very low complexity video encoding. Straightforward intra coding at the encoder is combined with including motion information at the decoder. In particular, the thesis focuses on the problems that typically emerge when exploiting temporal correlation solely at the decoder.

The focus of this thesis are the inherent performance limitations of DVC with low encoder complexity. Here, we only consider intra coding without any inter operability at the encoder. The thesis covers performance limitations of different DVC aspects, namely channel coding, motion estimation at the decoder and quantization. All proposed schemes focus on allowing real-time encoding. In channel coding, we investigate decoder-based modeling. In motion estimation at the decoder, we focus on true motion-based extrapolation. In quantization, we propose a trade-off between adaptivity and overhead.

The first discussed DVC aspect is the channel coding. To use state of the art LDPC codes efficiently, the behavior of the Virtual Dependency Channel (VDC) needs to be modeled accurately. The VDC is a virtual channel and comprises the prediction errors in the motion compensated prediction. We investigate the applicability of different channel models. The accuracy of these models is important to ensure efficient LDPC coding. We observe non-stationary behavior and show possibilities and limitations of to take it into account by means of decoder based classification.

The second discussed DVC aspect is the motion estimation at the decoder. In this context we find true motion estimation more suited to DVC than minimum residue motion estimation from conventional video coding. After the motion estimation, motion compensation is used to generate the side information. Literature mainly focuses on the side information quality and hence favors motion compensated interpolation. By contrast, we investigate the trade-off between side information quality and key frame cost. We find motion compensated extrapolation to be more suited to low complexity DVC.

The third discussed DVC aspect is the DCT quantization. The difference between different quantization methods is their adaptivity. Hence, we focus on the trade-off between adaptivity and required overhead to achieve the adaptivity in practice. We propose to use a scheme with both limited adaptivity and overhead.

Finally, we compare the derived solutions for each DVC aspect with its counterpart in conventional video coding. We find that DVC can outperform intra coding with a similar encoder complexity. However, for a less constrained encoder complexity conventional inter coding outperforms DVC by a large margin.

Sammenvatting

In de afgelopen twintig jaar heeft op het gebied van video codering de nadruk gelegen op het uitzenden van video. Videobeelden worden opgenomen en gecodeerd door professionele apparatuur en daarna bekeken op een verscheidenheid aan consumentenelektronica. Daarom was het doel de inpakefficiëntie te verhogen en de complexiteit van de decoder te verlagen. In de huidige tijd zien we een verschuiving in het gedrag van de gebruiker die naast het bekijken zich ook bezighoudt met het maken en delen van videobeelden. Voor het coderen van video op apparaten met beperkte mogelijkheden in plaats van professionele camera's wordt de complexiteit van de codering een belangrijke limitering.

Om de codeercomplexiteit de verlagen kan intra coding gebruikt worden, wat inhoudt dat de beweging in de beelden wordt genegeerd en de beelden onafhankelijk van elkaar worden gecodeerd. Het probleem van intra coding is de beperkte inpakefficiëntie doordat van tijdsafhankelijke informatie geen gebruik wordt gemaakt. Dit proefschrift stelt Distributed Video Coding (DVC) als een mogelijke oplossing om lage complexiteit videocodering te bewerkstelligen. Standaard intra coding bij de encoder wordt gecombineerd met het toevoegen van bewegingsinformatie bij de decoder. In het bijzonder legt dit proefschrift de nadruk op de problemen die ontstaan wanneer tijdscorrelatie alleen aan de decodeerkant wordt toegepast.

De nadruk van dit proefschrift ligt op de inherente prestatiebeperkingen van DVC met lage codeercomplexiteit. Hierin beperken we ons tot intra coding zonder enige interoperabiliteit aan de codeerkant. Het proefschrift bespreekt de prestatiebeperkingen van verschillende aspecten van DVC, namelijk kanaalcodering, bewegingsschatting aan de decodeerkant en kwantisatie. Al de voorgestelde schema's laten codering in realtime nadrukkelijk toe. Voor kanaalcodering onderzoeken we het modeleren aan de decodeerkant. Bij de bewegingsschatting aan de decodeerkant benadrukken we de extrapolatietechniek gebaseerd op true motion. Bij de kwantisatie ten slotte stellen we een afweging voor tussen adaptiviteit en overhead.

Het eerste aspect van DVC dat besproken wordt is kanaalcodering. Om state of the art LDPC codes efficiënt te kunnen gebruiken moet het gedrag van de Virtual Dependency Channel (VDC) nauwkeurig gemodelleerd worden. Het VDC is een virtueel kanaal en omvat de voorspellingsfouten in de bewegingsgecompenseerde voorspelling. We

onderzoeken de toepasbaarheid van verschillende kanaalmodellen. De nauwkeurigheid van deze modellen is belangrijk om zeker te zijn van een efficiënte LDPC codering. We nemen niet-stationair gedrag waar en laten hiervan de mogelijkheden en beperkingen zien, zodat er rekening mee gehouden kan worden door middel van decoder gebaseerde classificatie.

Het tweede aspect van DVC dat besproken wordt is de bewegingsschatting aan de decodeerkant. In deze context constateren wij dat de true motion schatting meer geschikt is voor DVC dan de minimumresidu bewegingsschatting van conventionele videocodering. Na bewegingsschatting wordt bewegingscompensatie toegepast om bij-informatie te genereren. In de literatuur ligt de nadruk voornamelijk op de kwaliteit van de bij-informatie en bestaat er een voorkeur voor bewegingsgecompenseerde interpolatie. Wij onderzoeken daarentegen de afweging tussen de kwaliteit van de bij-informatie en de keyframe kosten. Volgens ons is bewegingsgecompenseerde extrapolatie meer geschikt voor DVC met een lage complexiteit.

Het derde besproken aspect van DVC is de DCT kwantisatie. Het verschil tussen de verschillende kwantisatiemodellen zit in de adaptiviteit. We leggen de nadruk op de afweging tussen adaptiviteit en de benodigde overhead om de adaptiviteit in de praktijk te behalen. We stellen dat een schema gebruikt moet worden met zowel beperkte adaptiviteit als beperkte overhead.

Als laatste vergelijken we de bepaalde oplossingen voor elk aspect van DVC met zijn tegenhanger in conventionele videocodering. We vinden dat DVC beter kan presteren dan intra coding met een vergelijkbare codeercomplexiteit. Voor een minder beperkte codeercomplexiteit presteert conventionele inter coding echter een ruime marge beter dan DVC.