# IMPLEMENTATION OF A DISTRIBUTED VIDEO CODEC

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CEM VEDAT IŞIK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2008

Approval of the thesis:

**IMPLEMENTATION OF A DISTRIBUTED VIDEO CODEC**

submitted by CEM VEDAT IŞIK in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University by,

Prof. Dr. Canan Özgen
Dean, Graduate School of Natural and Applied Sciences _____

Prof. Dr. İsmet Erkmen
Head of Department, Electrical and Electronics Engineering Dept. _____

Assoc. Prof. Dr. Gözde Bozdağı Akar
Supervisor, Electrical and Electronics Engineering Dept. _____

**Examining Committee Members:**

Assoc. Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Aydın Alatan
Electrical and Electronics Engineering Dept., METU _____

Assis. Prof. Dr. Elif Uysal-Bıyıkoğlu
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Erdal Arıkan
Electrical and Electronics Engineering Dept., Bilkent University _____

**Date:** February 7, 2008

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Cem Vedat IŞIK

Signature :

# ABSTRACT

# IMPLEMENTATION OF A DISTRIBUTED VIDEO CODEC

Işık, Cem Vedat
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Akar, Gözde Bozdağı

February 2008, 107 pages

Current interframe video compression standards such as the MPEG4 and H.264, require a high-complexity encoder for predictive coding to exploit the similarities among successive video frames. This requirement is acceptable for cases where the video sequence to be transmitted is encoded once and decoded many times. However, some emerging applications such as video-based sensor networks, power-aware surveillance and mobile video communication systems require computational complexity to be shifted from encoder to decoder. Distributed Video Coding (DVC) is a new coding paradigm, based on two information-theoretic results, Slepian-Wolf and Wyner-Ziv, which allows exploiting source statistics at the decoder only. This architecture, therefore, enables very simple encoders to be used in video coding. Wyner-Ziv video coding is a particular case of DVC which deals with lossy source coding where side information is available at the decoder only. In this thesis, we implemented a DVC codec based on

the DISCOVER (DIStributed COding for Video sERvices) project and carried out a detailed analysis of each block. Several algorithms have been implemented for each block and results are compared in terms of rate-distortion. The implemented architecture is aimed to be used as a testbed for future studies.

**Keywords:** Distributed Video Coding, Wyner-Ziv, Slepian-Wolf, low complexity encoding

# ÖZ

## BİR DAĞITIK VİDEO KODLAYICI - KODÇÖZÜCÜNÜN GERÇEKLEŞTİRİMİ

Cem Vedat IŞIK
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Danışmanı: Doç. Dr. Gözde Bozdağı Akar

Şubat 2008, 107 sayfa

Günümüzde kullanılan MPEG4 ve H.264 gibi video kodlama standartları, peş peşe gelen video çerçevelerinin birbirleri arasındaki benzerliklerinden yararlanmak için yüksek karmaşıklığa sahip kodlayıcı gerektirir. Bu gereksinim, aktarılacak videonun bir kere kodlanıp defalarca çözüldüğü sistemler için uygundur. Ancak, video tabanlı sensör ağları, güç tasarruflu gözetme sistemleri ve taşınabilir video iletişim sistemleri gibi gelişmekte olan bazı teknolojiler hesaba dayalı karmaşıklığın kodlayıcıdan çok çözücüye aktarılmasına ihtiyaç duyar. Dağıtık Video Kodlama (DVK), iki bilişim kuramsal sonuca, Slepian-Wolf ve Wyner-Ziv, dayanan ve kaynak istatistiğinin sadece çözücü tarafında kullanılmasını sağlayan yeni bir kodlama örneğidir. Bu mimari, böylece, video kodlamasında oldukça basit kodlayıcıların kullanımına olanak vermektedir. Wyner-Ziv kodlaması, DVK'nın bilgi kayıplı kaynak kodlaması ile ilgilenen ve yan bilginin sadece çözücü tarafında olduğu özel bir durumudur. Bu tez kapsamında, DISCOVER (DIStributed COding for Video sERvices) projesini temel alan bir DVK kodlayıcı-kodçözücüsü gerçekleştirilmiş ve her blok için detaylı bir çözümleme

yapılmıştır. Her blok için çeşitli algoritmalar gerçekleştirilmiş ve sonuçlar hız-bozulum açısından karşılaştırılmıştır. Gerçekleştirilen mimarinin gelecek çalışmalar için bir sınama ortamı olması amaçlanmıştır.

**Anahtar Sözcükler:** Dağıtık video kodlaması, Wyner-Ziv, Slepian-Wolf, düşük karmaşıklıkla kodlama

To My Sister Işıl

# ACKNOWLEDGEMENTS

I would like to thank all who have contributed one way or the other to make this thesis possible. Special thanks to Dr. Gözde Bozdağı Akar, supervisor of this thesis, for her guidance, enthusiasm, encouragement, outstanding support and patience. Thanks to Mehmet Oğuz Bici, for his unconditional friendship and words of incentive during this thesis. Thanks also for helping me in the development, for the valuable suggestions and technical support. I would like to thank Anıl Aksay, whose familiarity with the subject was invaluable.

Thanks to Murat Deniz Aykın, Emrah Bala, Berkan Solmaz and all colleagues from Multi Media Research Group, for their suggestions and availability to help me whenever it was necessary and also for the very pleasant working environment.

Warm words of gratitude are addressed to my parents Kezban - Vedat IŞIK and Funda, whose love, understanding, patience and encouragement has made this thesis possible.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| CRC | Cyclic Redundancy Check |
| DCT | Discrete Cosine Transform |
| DPCM | Differential Pulse Code Modulation |
| DSC | Distributed Source Coding |
| DVC | Distributed Video Coding |
| FEC | Forward Error Correction |
| FEP | Forward Error Protection |
| FPS | Frame per Second |
| GSM | Global System for Mobile Communications |
| GOP | Group of Pictures |
| IDCT | Inverse Discrete Cosine Transform |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| ITU | International Telecommunication Union |
| ITU-T | Telecommunication Standardization Sector |
| LDPC | Low Density Parity Check |
| LDPCA | LDPC Accumulate |
| MPEG | Moving Picture Experts Group |
| PSNR | Peak Signal-to-Noise Ratio |
| QCIF | Quarter Common Intermediate Format |
| RD | Rate Distortion |
| RS | Reed-Solomon |
| RSC | Recursive Systematic Convolutional |
| SAD | Sum of Absolute Differences |
| SI | Side Information |
| VCEG | Video Coding Experts Group |
| WZ | Wyner-Ziv |

# CHAPTER 1

# INTRODUCTION

Today's video compression standards such as ITU-T VCEG H.264 [1] and ISO/IEC MPEG4 [2] use interframe predictive coding. In this paradigm, encoding lies on Discrete Cosine Transform (DCT) and motion compensation in order to exploit the spatial and temporal correlations existing in a video sequence. In order to exploit correlations, the encoder, which is responsible for the motion estimation task, requires a higher computational complexity than the decoder. This kind of architecture is suitable for conventional video delivery systems such as video-on-demand systems and video broadcasting where the video sequence is encoded once but decoded many times. However, technologies like wireless surveillance systems, wireless video cameras, video-based wireless sensor networks and mobile video communication systems have been very popular in recent years due to increasing research and development activities on these technologies. These systems have different requirements than those of conventional video delivery systems. For applications, such as video-based wireless sensor networks, it is essential to have large number of sensor nodes, therefore low-cost and simple encoders. For applications such as mobile video communication systems, low power consumption for encoders is an important constraint. While fulfilling these requirements, achieving a coding performance closer to conventional interframe video coding standards is also essential.

In conclusion, a challenging problem emerged with new visual communication systems like stated above: "How can we achieve both

efficient and low-complexity video compression, which can not be obtained by traditional video coding systems?”

Several information theoretic results [3], [4] suggest that this question can be answered by exploiting source statistics, totally or partially, at the decoder only. These results drove the design of new types of coding algorithms, Distributed Video Coding (DVC) solutions which will be presented in the following chapters.

## 1.1 Applications

The "low-complexity encoder, at the expense of a higher decoder complexity" architecture of DVC, makes it a suitable solution for new emerging technologies such as, video-based sensor networks, wireless mobile video communication, low-power surveillance networks and multi-view video recording, where the encoder complexity, power capacity and/or cost are limited resources. Some applications to which DVC is a promising solution are described in the following:

### 1.1.1 Video-based Sensor Networks

By the advances in microelectronics and telecommunications, sensor networks and their applications become a new area for research. The main goal of a wireless sensor network is to set up a wireless network consisting of spatially distributed autonomous devices (sensor nodes) to cooperatively accomplish a certain task. The development of sensor networks was originated by military applications such as battlefield surveillance [5] [6] and underwater sensor networks [7]. However, today, some wireless sensor networks now have video acquisition capabilities for civilian application areas, which make it possible to track persons, vehicles or events throughout an environment, monitoring of activities, and triggering alarms, if needed. Also while decoding data received from sensor nodes, with Wyner-Ziv coding, independently encoded data from neighboring sensor nodes can be jointly decoded by

exploiting the correlation between neighboring nodes' data. In this architecture, the decoder will have high computational capabilities, therefore high-complexity, as a central data processor, in order to exploit the correlation among received data. Therefore, low-cost, power-aware and low-complexity sensor nodes can be developed and deployed. Such a system is illustrated at Figure 1.1.

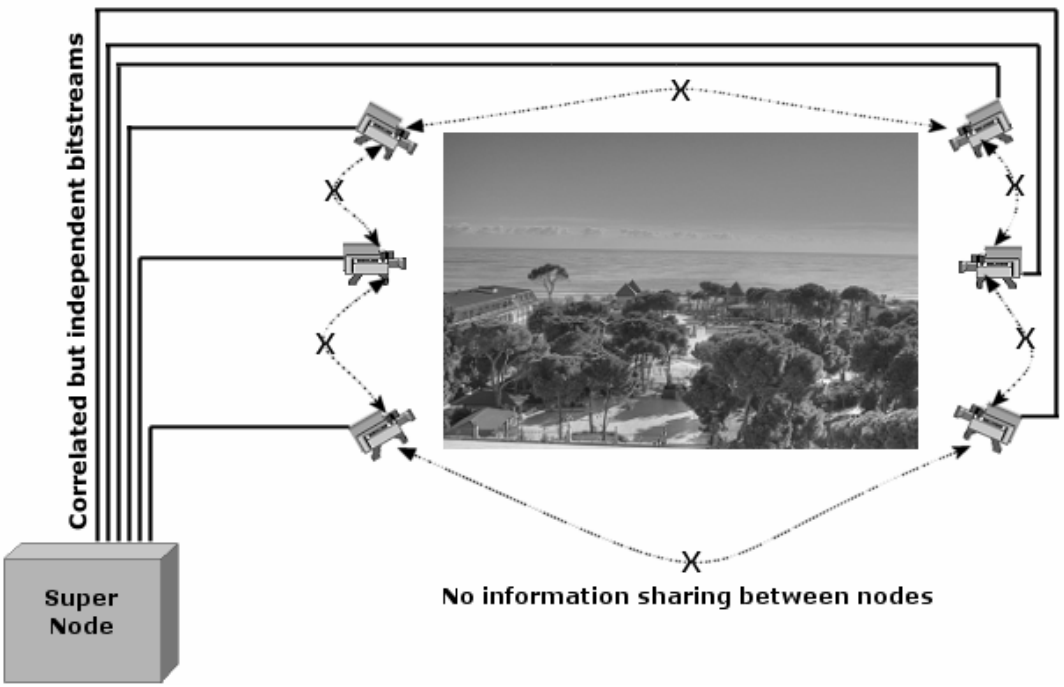


Figure 1.1: Low-complexity sensors transmitting data to a super node

### 1.1.2 Wireless Mobile Video Communication

Wireless video communication between two or more camera phones is another application that can benefit from the low-complexity encoder architecture of DVC. Since power consumption, therefore battery life, of terminals is a bottleneck in such systems, the main requirement is to have low-complexity encoder and low-complexity decoder in each

terminal. However, to make us of distributed video coding in such an application, it is necessary to have a high-complexity decoder somewhere in between. In such architecture, the high-complexity decoder and a transcoder can be located at a GSM base station. The base station will receive the low-complexity encoded bit stream (Wyner-Ziv bit stream) from one terminal, transcode it to a ISO/IEC MPEG or ITU-T standard H.26x bit stream and transmit it to other terminals with low complexity decoders. The architecture described here, allows a low-complexity encoder and low-complexity decoder to be deployed at each terminal. (Figure 1.2)

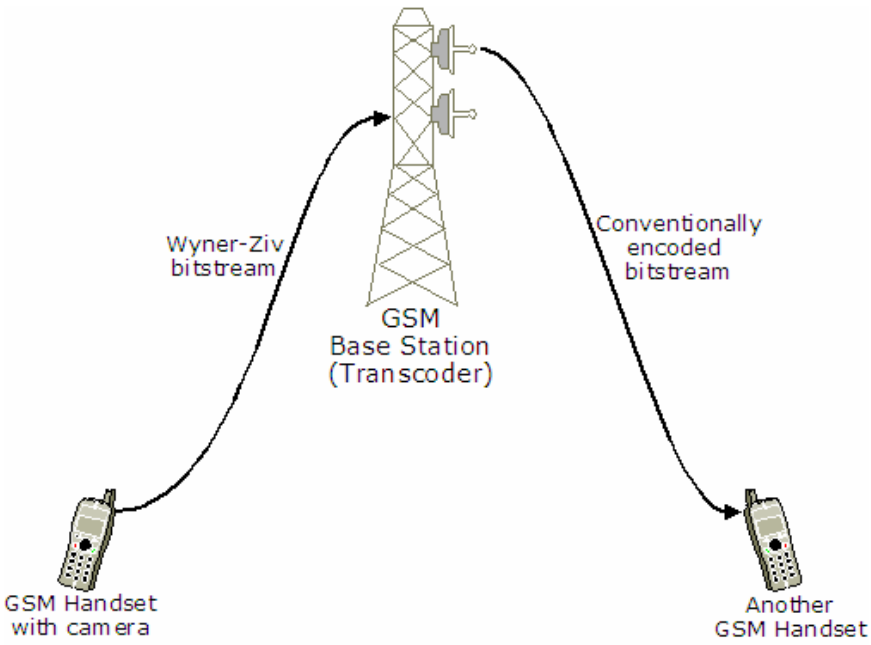


Figure 1.2: Video Conferencing between two mobile terminals

### 1.1.3 Low-Power Surveillance Networks and Multi-View Video Recording

Generally in a surveillance network system, events are sensed from different locations by multiple cameras and all video sequences are collected in one recorder system. Similar to the approach in Section 1.1.1, cameras that have overlapping line of sight, sense correlated events, therefore video sequences, in overlapped areas. Assuming that the video sequences encoded by low-complexity cameras are decoded in a single high-complexity central system, the number of encoders used is larger than the number of decoder which makes a suitable scenario for distributed video coding since it enables to exploit the correlation between the encoded sequences at the decoder, yielding low encoding complexity. A similar approach can be used in acquiring multi view video of an object or scene, where neighboring cameras will have correlated video sequences [8]. (Figure 1.3)

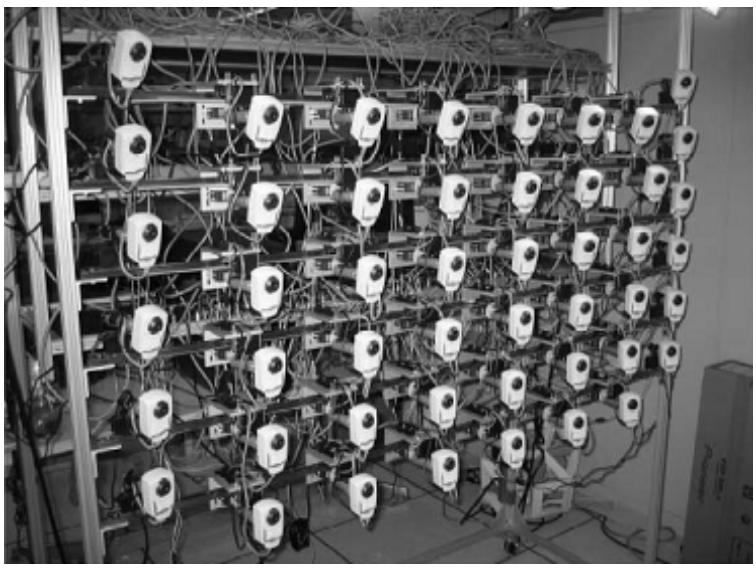

Figure 1.3: Large camera array of 48 cameras of Advanced Multimedia Processing Lab, Carneige Mellon [9]

### 1.1.4 Networked Broadcasting Video Recorders

Networks of broadcasting video recorders are used for shooting and recording in application contexts such as music news, concerts, sports, meetings etc. This application scenario – networked broadcasting video recorders – is mostly characterized by the usage of multiple video recorders shooting, recording and streaming the same scene, including the capability of later access on demand via wired or wireless channels to the views corresponding to any of the broadcasting video recorders. For broadcasting video recorders, the transmission capabilities can be quite demanding. Another possibility is to allow directors to remotely control the broadcasting video recorder in terms of shooting direction/angle, zooming, etc. These recorders may offer the ability to use intelligent image analysis functions and different types of networking protocol support. This implies for example that the broadcasting video recorders don't need to be transmitting continuously and simultaneously since they may be accessed one by one depending on the director's choice [10]. (Figure 1.4)

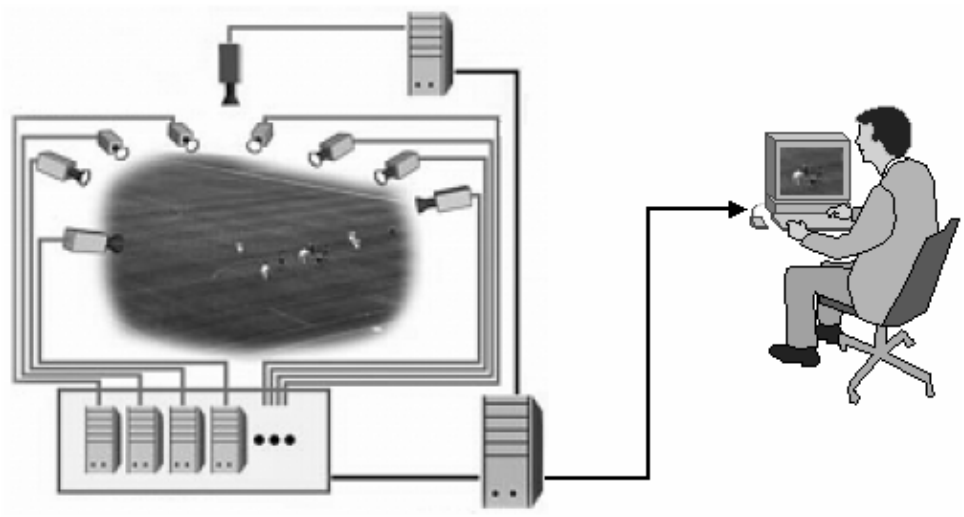

Figure 1.4: Networked Broadcasting Video Recorders

## 1.2 Objective of the thesis

The main objective of this thesis is to implement an end-to-end distributed video coding / decoding system and carry out a detailed analysis of each fundamental block of the system.
The implemented codec architecture for the DVC scenario is based on the DISCOVER [11] (DIStributed COding for Video sERvices), a European project which has been devoted to the advancement of Distributed Video Coding.

## 1.3 Outline of the Thesis

The motivation and context for this work and definition of main objectives together with its possible applications are presented in Chapter 1.
Chapter 2 summarizes the background and general idea behind distributed video coding. Also, a review of the most relevant distributed video coding schemes is presented.
Chapter 3 focuses on the Wyner-Ziv coding and the implemented Wyner-Ziv Codec Architecture based on the DISCOVER project.
Chapter 4 illustrates the results of the experiments performed to evaluate the performance of the implemented Wyner-Ziv Codec. The comparisons between the most relevant distributed video coding schemes are also presented in this chapter.
Finally, Chapter 5 gives conclusions of this thesis together with possible future work.

# CHAPTER 2

# DISTRIBUTED VIDEO CODING

## 2.1 Background on Distributed Video Coding

Distributed Source Coding (DSC) is a coding paradigm that is based on joint decoding of two or more independent but statistically correlated random sequences which are encoded by different encoders. The term "distributed" refers to the encoding mode of signals. Each correlated but independent encoded signal is sent to a single joint decoder which exploits the statistical dependencies between received signals. Distributed Video Coding (DVC) is a special video coding paradigm that is based on the independent encoding-joint decoding architecture of DSC. In DVC, statistical dependency of signals may refer to overlapping areas sensed by multiple cameras or spatial and temporal dependencies among successive video frames in a sequence and so. Thus, it is possible to reduce the encoding complexity of video signals by exploiting the redundancies among encoded signals at the decoder. As mentioned in the first chapter, the conventional techniques for encoding of video do not provide an acceptable solution for emerging visual communication systems. Traditional video coding schemes aim to find the minimum encoding rate, *R*, required such that two statistically dependent sequences *X* and *Y*, for instance the consecutive frames of a video sequence, can be perfectly recovered, i.e. without errors, by a joint decoder. From the information theory this rate is equal to joint entropy $H(X,Y)$. See Figure 2.1 below.

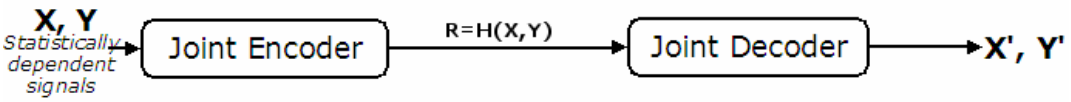


Figure 2.1: Traditional Coding Example

### 2.1.1 Slepian-Wolf Theorem

What if, two statistically dependent signals, *X* and *Y,* are separately encoded but the decoding process is performed jointly? This is the Distributed Source Coding.

Consider again two correlated signals, *X* and *Y*. Let the decoder have access to one of them, say *Y* as illustrated in Figure 2.2. In this case, we desire to find the rate required to transmit *X* with asymptotically small probability of reconstruction error. If we do not exploit the fact that *Y* is available at the decoder, then by the source coding theorem a rate $R = H(X)$ is sufficient to transmit *X* noiselessly. If *Y* were also made available at the encoder as illustrated in Figure 2.3, then by the source coding theorem, *X* could be transmitted noiselessly at a smaller rate $R = H(X|Y)$. Slepian-Wolf [3] result shows that even when *Y* is known only at the decoder, with the knowledge of joint statistics alone, the encoder of *X* can perform just as well as the case when *Y* is available to both of them (i.e. transmit *X* noiselessly at a rate $R = H(X|Y)$). (See Figure 2.2). A consequence of the above result is that this performance can be achieved independent of the actual realization of *Y* since the encoder of *X* uses only the joint statistics.

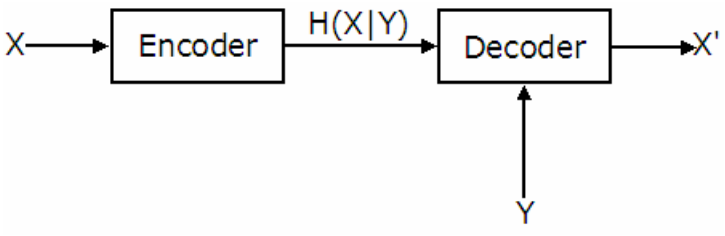


Figure 2.2: Only decoder has access to *Y*. The Slepian-Wolf theorem says that *X* can still be described at *H(X|Y)* bits per sample.

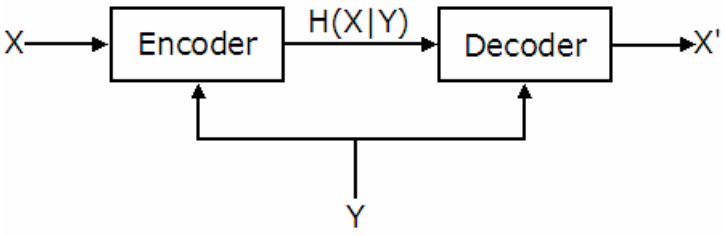


Figure 2.3: Both encoder and decoder have access to side information *Y* which is correlated to *X*. *X* can be described by *H(X|Y)* bits per sample

The achievable rates for recovering *X* and *Y* with arbitrarily small error probability in separate encoding - joint decoding case are illustrated in Figure 2.4.

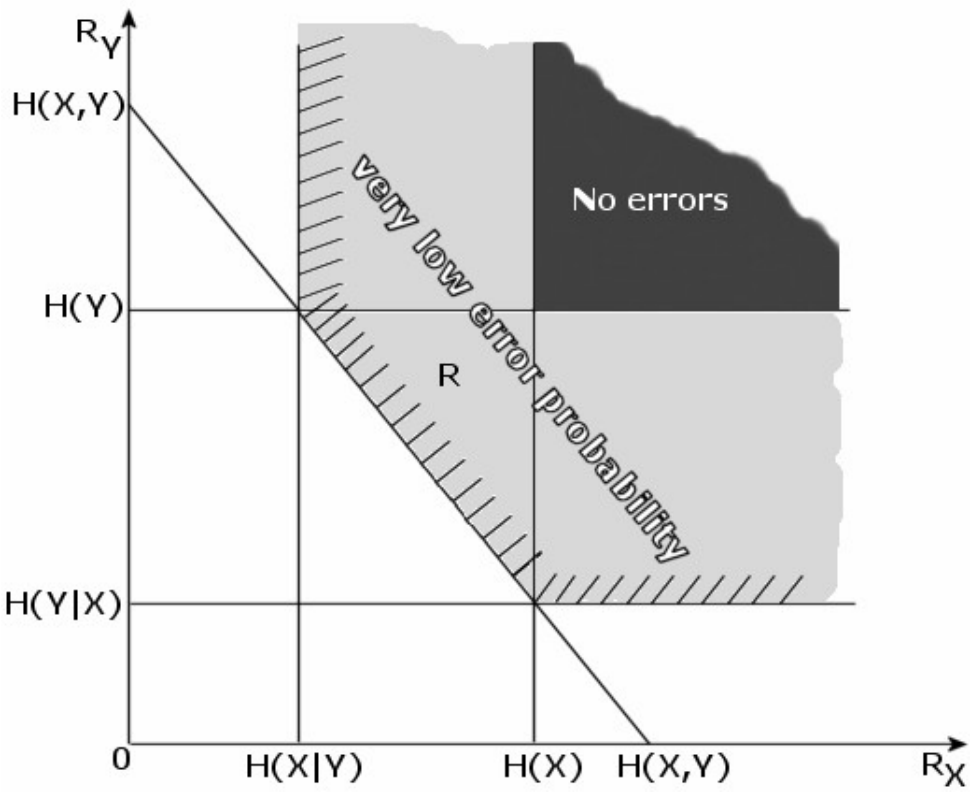


Figure 2.4: Achievable rate regions stated by Slepian and Wolf

Slepian and Wolf stated that independent encoding in distributed source coding paradigm does not need to have any compression efficiency loss when compared to joint encoding which is used in the traditional video coding paradigm. In literature the term "Slepian-Wolf Coding" is generally used for the lossless distributed source coding since it assumes that the two statistically correlated sequences which are independently encoded are reconstructed with an arbitrarily small error probability at the joint decoder. Lossless in this context is different from mathematically "lossless" term, since a very small controlled amount of error is considered.

### 2.1.2 Wyner-Ziv Theorem

Wyner and Ziv [4] studied a particular case of Slepian-Wolf coding corresponding to the rate point *(H (X|Y), H(Y))* in Figure 2.4. This particular case deals with the source coding of the *X* signal considering the *Y* signal, known as side information, is available at the decoder. This is also referred as "*lossy compression with decoder side information*". Figure 2.5 illustrates such a scenario. $R_{WZ}(d)$ is the minimal rate at which one source (*X*) can be transmitted at a given distortion.

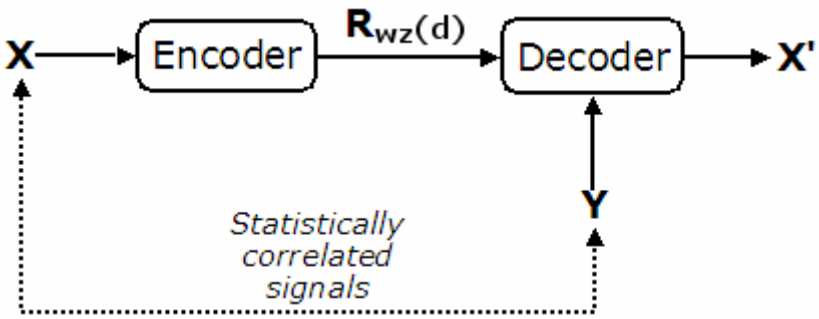


Figure 2.5: Lossy compression with side information at decoder

The term *lossy compression* is due to Wyner and Ziv having considered an average, acceptable distortion, between the signals *X*, and its decoded version *X'*. Several realistic scenarios, e.g. multi-camera systems (surveillance scenario) and video coding can be characterized by the illustration in Figure 2.5.

The Wyner and Ziv work establishes the minimum rate $R_{WZ}(d)$ necessary to encode *X* guaranteeing its reconstruction with an average distortion below *d*, assuming that the decoder has the side information *Y* available. Let $R_{X|Y}(d)$ be the rate-distortion function which results

when the encoder as well as the decoder has access to the side information *Y*. In nearly all cases it is shown that

$$R_{WZ}(d) \geq R_{X|Y}(d), \qquad d > 0 \tag{2.1}$$

So that knowledge of the side information at the encoder permits transmission of the *X* at a given distortion level using a smaller transmission rate. This is in contrast to the situation treated by Slepian and Wolf [3] where, for arbitrarily accurate reproduction of *X*, i.e., d = $\varepsilon$ for any $\varepsilon$ > 0, knowledge of the side information at the encoder does not allow a reduction of the transmission rate.

In literature, there are two major areas where distributed video coding finds application, notably low-complexity video coding (as in [16], [19] and [20]) and robust video transmission (as in [15] and [18]). While in the former case the main goal is to compress video using a low complexity encoder, in the latter case an additional bitstream is produced in order to correct transmission errors in a traditionally coded video signal. The first Wyner-Ziv practical schemes in two major areas used pixel by pixel encoding and decoding, also called pixel domain Wyner-Ziv coding (e.g. [12], [13], [14] and [21]). Later on, some Wyner-Ziv codecs (e.g. [11], [15], [16] and [22]) appeared including a block-based transform module, after transform coding has been studied in the context of distributed source coding. It has been realized that pixel domain Wyner-Ziv coding performance [20] can come close to that of transform domain solutions. In the following sections, the most relevant and cited examples of Wyner-Ziv video coding solutions will be presented, namely, i) PRISM Solution, ii) Stanford Image, Video, and Multimedia Systems Group Solutions and iii) DISCOVER.

## 2.2 The PRISM Solution

"PRISM: **P**ower-efficient, **R**obust, h**I**gh-compression, **S**yndrome-based **M**ultimedia coding" was first presented by Puri and Ramchandran [18].

The main goal of PRISM solution is to combine intraframe coding features such as robustness to transmission errors and low-complexity encoding with the compression efficiency of interframe coding. The PRISM solution proposes a video encoding scheme based on Wyner-Ziv coding which is illustrated in Figure 2.6.

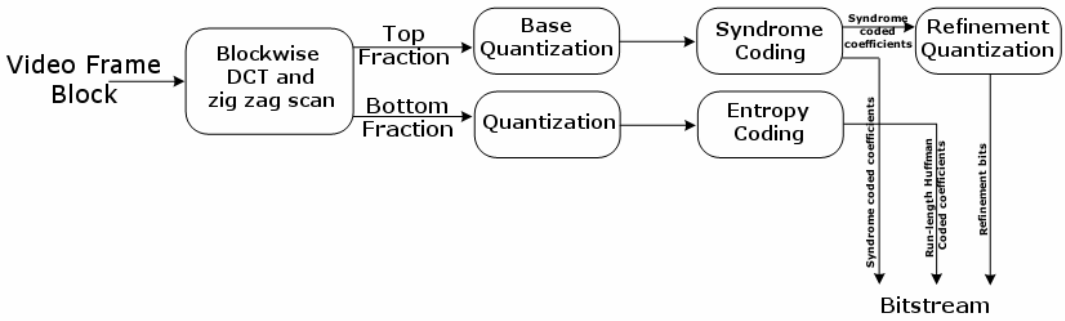


Figure 2.6: PRISM Encoder Architecture [18]

In this scheme several side information candidates [18] substitute the 'single' side information that characterizes Wyner-Ziv coding. Figure 2.6 and Figure 2.7 illustrate the architectures of the PRISM encoder and decoder, respectively.

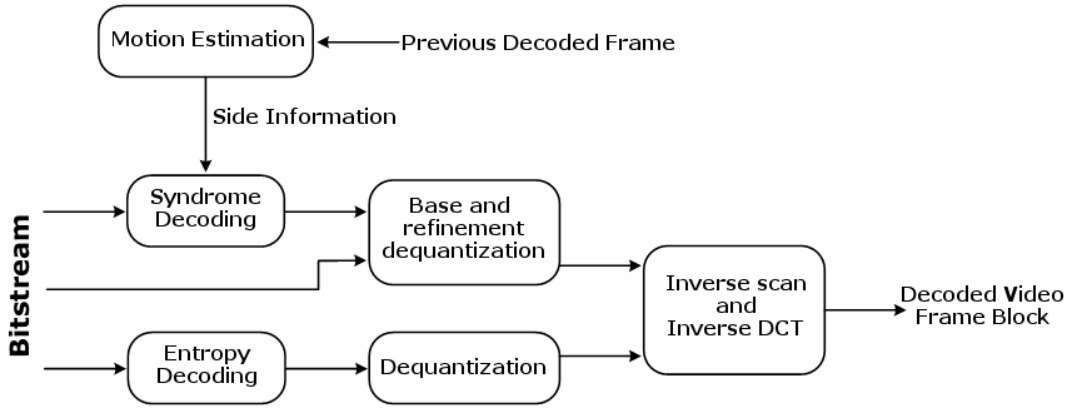


Figure 2.7: PRISM Decoder Architecture [18]

### 2.2.1 Encoding in PRISM

Illustrated in Figure 2.6, each video frame is divided into 8x8 or 16x16 samples blocks. Thus, different frame regions may be described by different amounts of motion; different frame regions may therefore be characterized by different correlation intensities. Each obtained 8×8 or 16×16 sample block is classified into one of several pre-defined classes according to the statistical dependency between the current frame block and the co-located block in the previous frame. In [18], the correlation intensity has been modeled through the squared error between each current frame block and the co-located one in the previous frame. The decision of what kind of encoding is well-suited for each block of the current frame is made in this stage, either no coding (class: skip) or traditional coding (class: intracoding) or syndrome coding (class: syndrome coding). The current frame blocks classified in the skip class are not encoded and the blocks classified in the intra coding class are traditionally encoded. The blocks classified in the syndrome coding class constitute the PRISM novelty; therefore, a special attention is given to the encoding-decoding procedures of those blocks. The encoding modes classes selected for the current frame blocks are then transmitted to the decoder as header information. To

each obtained current frame sample block, a blockwise DCT is applied and the resulting transform coefficients are then zig zag scanned. Most of the block's energy is concentrated in a small number of transform coefficients corresponding to the lower frequency coefficients. Relying on this idea, for the blocks classified in the syndrome coding mode class, the PRISM solution encodes the low-frequency coefficients using syndrome coding while the high-frequency coefficients are traditionally encoded (i.e. quantized and entropy encoded). Typically, many of the high-frequency coefficients have low or near-zero values and therefore entropy coding uses few bits to send those transform coefficients. On the other hand, the low-frequency coefficients have high values and the syndrome encoding will allow reducing the bitrate needed to transmit them; bitrate reduction is achieved since instead of transmitting each individual codeword corresponding to a quantized transform coefficient, syndrome coding only transmits the index of the set containing that codeword. The zig zag scanned transform coefficients are then quantized to generate quantized codewords. The DC coefficient (the lowest frequency transform coefficient) and a small number of AC coefficients near the DC (in a zig zag scan order) are quantized in the base quantization architectural module; these transform coefficients are called Wyner-Ziv coefficients. The choice of the base quantization step size is determined by the correlation level of each 8×8 or 16×16 block within the current frame determined in the classification stage. As can be seen Figure 2.6 the remaining high-frequency transform coefficients are fed into the other quantization module; the quantization step size corresponds to the desired distortion in the reconstruction. Hence, the base quantization and the quantization architectural modules Figure 2.6 differ in the quantization step size that typically assumes different values. The quantized codewords space is divided into several groups of codewords called cosets. Each coset has an index label associated; this index label, known as syndrome, points out the coset to which the codeword corresponding to a quantized

transform coefficient belongs to. Since the number of syndromes is lower than the number of codewords, the number of bits required to encode a syndrome is less than the number of bits needed to encode a codeword. Using syndrome coding, the transmission of individual codewords associated to the quantized transform coefficients is replaced by the transmission of syndromes and therefore compression is achieved. In the PRISM solution, a trellis-based syndrome code (128-state rate-½ trellis code) is applied to quantized low-frequency transform coefficients of each 8×8 or a 16×16 block. The resulting syndrome bits are incorporated in the bitstream syntax at the block level. As mentioned when describing the quantization process in Figure 2.6, low-frequency and high-frequency transform coefficients of 8×8 or 16×16 blocks are quantized with different quantization step sizes. Compressing a signal with different quantization step sizes corresponds to attaining different signal reconstruction quality levels at the decoder. Therefore to attain a desirable block reconstruction quality, it is essential to choose a specific quantization step. For syndrome encoded low-frequency coefficients, the choice of the quantization step size depends on the correlation between a block and the co-located one in the previous frame, determined at the classification stage; by doing this, the trellis codes decoding error probability is minimized. In order to attain a global desirable reconstruction quality, a refinement of the base quantization step size is performed. The refinement quantization process corresponds to sub-partitioning the base quantization interval in order to obtain the quantization step size corresponding to the desirable reconstruction quality. The sub-partitions within the base quantization interval are called refinement intervals; each refinement interval has an index associated to it. The refinement bits associated to the refinement interval index are transmitted to the decoder; these bits are another component of the bitstream syntax at the block level. The quantized transform coefficients corresponding to the high-frequency coefficients that have not been syndrome encoded are then

traditionally entropy encoded using run-length Huffman coding. The resulting bits, called pure source coded bits, are incorporated in the bitstream syntax at the block level. Beyond the five steps, DCT, quantization, syndrome coding, refinement quantization, and entropy coding, performed by the PRISM encoder, a cyclic redundancy check (CRC) of the base quantized transform coefficients is also computed and transmitted to help the decoder performing the motion estimation task. As illustrated in Figure 2.8, the bitstream syntax at the block level encloses syndrome bits, CRC bits, refinement bits and pure source coding bits.


Figure 2.8: Bitstream syntax at the block level [18]

The complexity burden of PRISM encoding is similar to that of a conventional intraframe encoding solution because of the transform and the entropy coding modules as in traditional intraframe encoding. [18]

### 2.2.2 Decoding in PRISM

The PRISM decoder architecture is presented in Figure 2.7. For the frame blocks which are classified in the skip class, i.e. not encoded frame blocks, the co-located blocks in the previous reconstructed frame are used as reconstructed blocks; the blocks which are classified in the intra coding class are decoded using inverse operations to those performed at the encoder: entropy decoding and dequantization.
The motion estimation task provides the necessary information to decode the received syndrome bits. In the PRISM decoding

architecture, this information consists of several candidates to prediction block (*i.e. more than one side information*) instead of the "single" side information that characterizes the basic Wyner-Ziv coding scenario. The candidate predictors are obtained from the previous reconstructed frame by half-pixel motion estimation. A full motion search motion estimation algorithm is used with half-pixel accuracy, i.e. all neighboring blocks within a search range are used as candidate blocks which is very similar to what is done at the encoder side in traditional video codecs. For each 8×8 or 16×16 block, the received syndrome bits together with one of the candidate predictors are used to decode quantized codewords sequence. From the received syndrome bits, it is possible to obtain several quantized codeword sequences. To find out which is the closest sequence to the candidate predictor within this set of quantized codeword sequences, the Viterbi algorithm is used [18]. If the identified closest sequence does not match the CRC received, the syndrome decoding process is performed again using another candidate predictor which has been generated by the motion search. The syndrome decoding process stops when the closest sequence identified matches the CRC received. The CRC is used as a unique signature for each block and allows identifying the best candidate predictor. The base dequantization is performed after the quantized coefficients are reconstructed. In order to achieve better reconstruction quality, the base dequantization is followed by refinement dequantization using the refinement bits which are transmitted by the encoder. At this stage of the PRISM decoding process, there are two estimates for the Wyner-Ziv coefficients: i) the coefficients of the prediction block found in the motion estimation stage and ii) the coefficients obtained through syndrome decoding, base dequantization and refinement dequantization. The Wyner-Ziv coefficients final estimate is obtained by employing a linear estimation algorithm. The received pure source coded bits, corresponding to the quantized high-frequency transform coefficients, are decoded using

inverse operations to those performed at the encoder: entropy decoding and dequantization. After inverse zig zag scanning of the decoded transform coefficients, the inverse discrete cosine transform (IDCT) is then applied. So, the PRISM decoding process is completed.

### 2.2.3 Experimental results of PRISM reported in literature

In order to evaluate the performance of the proposed system, the authors coded several sequences among which only Carphone sequence results will be mentioned. The first frame of each video sequence is fully intra mode encoded, i.e. each block of the frame is encoded in intra mode for both the PRISM and H.263+ coders. Figure 2.9 show the rate-distortion performance comparison of PRISM system with the rate-distortion performance of a H.263+ video coder, when there is no frame loss. As it can be noticed from Figure 2.9 and other rate-distortion comparisons of [18], the PRISM rate-distortion performance is between the inter and the intra coding modes of the H.263+ coder, independently of the motion content associated to the video sequence. Even though the performance of PRISM is not as good as interframe coding, one of the main advantages is that PRISM has a negligible effect on the quality of the decoded video, since the error propagation is stopped due to the absent of a prediction loop at the PRISM encoder. However, in traditional video coding schemes, errors accumulate and propagate to the following frames of the video sequence.

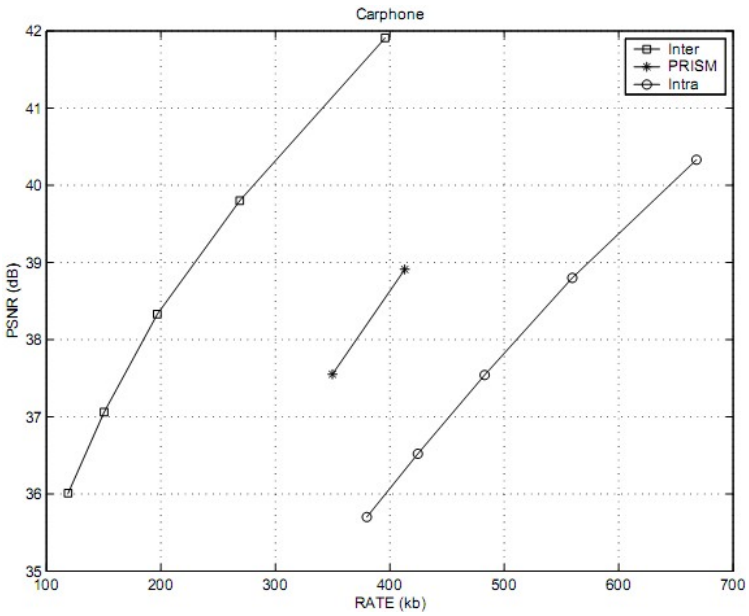


Figure 2.9: RD performance of PRISM with Carphone sequence [18]

### 2.2.4 Remarks on PRISM

There are three main architectural goals of PRISM; i) flexible distribution of complexity without compromising the compression performance, between encoder and decoder ii) robustness to drift between encoder and decoder caused by lack of synchronization due to channel loss and iii) a very light and powerful syntax that is well-suited for standardization and seamless innovation within the standard. The PRISM concept applies even if only a subset of these goals is desired. For example, if complexity is not the main problem but robustness is, the PRISM approach can deployed to make standard fragile prediction-based video streams robust, as a replacement for FEC-based error-resiliency solutions. At the other extreme, robustness may be less important than low-complexity encoding, for example in high-end video camera applications, where full-motion MPEG represents a processing bottleneck due to a high frame rate acquisition. However, the full power of PRISM is expressed when both low-complexity and robustness

are needed, for example in video-over-wireless applications like surveillance cameras or video telephony, where device memory and/or battery life are important, and the transmission environment causes packet/frame drops. In addition to that, PRISM employs syndrome encoding modes with different rates, depending on the estimated frame difference energy and the rate control scheme of PRISM does not require online decoding and a feedback channel, thus making it suitable for storage applications. However, this mode decision mechanism brings additional complexity to PRISM Encoder.

## 2.3 Stanford Video Coding Solutions

Several distributed video coding schemes based on Slepian-Wolf and Wyner-Ziv theorems have been proposed by the Stanford Image, Video, and Multimedia Systems Group [24]. These approaches can be summarized in three architectures; Low-Complexity Video Coding, Robust Video Coding and Residual Video Coding. The main ideas behind these approaches are low-complexity and error resilient video coding. They achieved these in three different architectures as will be described in the following sections.

### 2.3.1 Low-Complexity Video Coding Solution

The first low-complexity video codec presented by the Stanford Image, Video, and Multimedia Systems Group is [12], where a video sequence is divided into key frames (the odd frames of the video sequence) and Wyner-Ziv frames (the even frames of the same sequence). Then each Wyner-Ziv frame is pixel by pixel intra encoded, independently of the key frames and other even frames. To decode a Wyner-Ziv frame, the side information which is an estimate of the Wyner-Ziv frame, is generated through frame interpolation techniques using the key frames which are assumed to be available without any loss at the decoder. A more flexible approach was presented in [14] where the number of

Wyner-Ziv frames between key frames, GOP size, may vary, which differs from [12], the key frames are traditionally intraframe encoded with a H.263+ coding standard and the Wyner-Ziv frames are encoded as in [12]. Using previously reconstructed both Wyner-Ziv and key frames, by frame interpolation or extrapolation, the side information is generated. In [16], an architecture similar to the one in [14] is proposed with a major difference that in [16] transform coding is considered in Wyner-Ziv frame coding instead of pixel by pixel coding, and again, the Wyner-Ziv frames are even frames of the video sequence and the remaining frames are the key frames, GOP size is two. In [19], the idea of sending robust hash codewords from the encoder, in addition to the Wyner-Ziv bits to aid the decoder in estimating the motion and generate the side information was presented. In [23], the system in [19] has been improved by treating the high frequency components of the frame as the hash. All the approaches which are briefly described above are based on an intraframe encoder-interframe decoder system. Namely, each Wyner-Ziv frame is encoded independently of the other Wyner-Ziv frames and key frames, i.e. similarities with other video frames are not exploited at the encoder, but the decoding is performed jointly. In intraframe encoder-interframe decoder architecture, high compression efficiency may be achieved in the joint decoding process since correlations between frames are explored at the decoder only through frame interpolation or extrapolation techniques.

The most recent Wyner-Ziv low-complexity video coding solution originating from the Stanford group is [23] which is proposed by Aaron *et al*. In this solution, Wyner-Ziv coding is only performed on low frequency coefficients of the frame, which tend to have significant correlation with the corresponding coefficients from the previous frame. High frequency components of the frame can be transmitted as supplementary information (hash) about the current Wyner-Ziv frame. The high frequency coefficients, if sent, are compressed by efficient

run-length coding and are used at the decoder in the inverse transform and in estimating the motion. Since the high frequency components contain important edge information, relying only on these coefficients for the motion search still results in accurate motion estimation. The motion-compensated previous frame is then used as side information for Wyner-Ziv decoding the low frequency coefficients. Figure 2.10 illustrates the architecture proposed in [23].

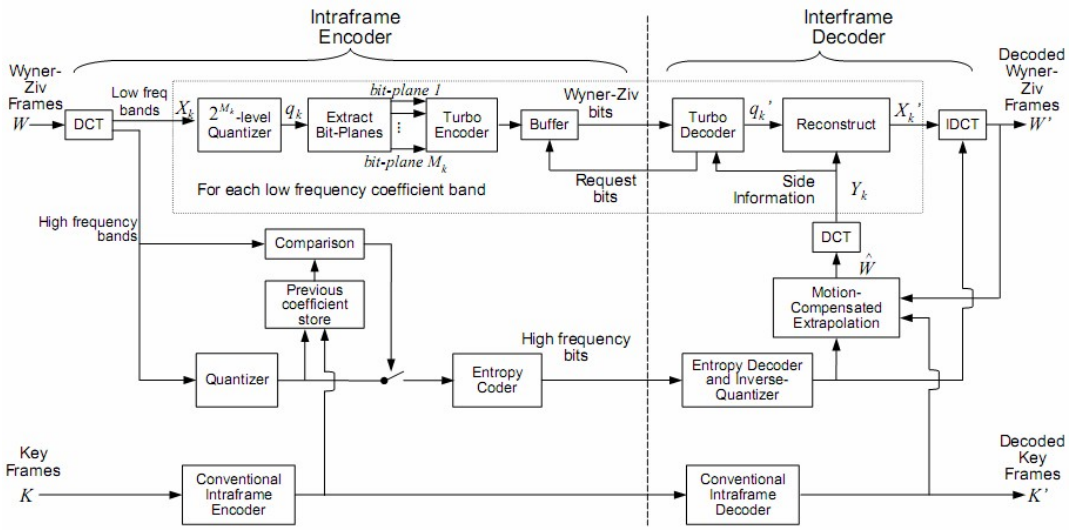


Figure 2.10: Intraframe encoder-Interframe Decoder Architecture [23]

In the proposed solution [23], the frames of a video sequence are organized into Groups of Pictures (GOPs). Each GOP is constituted by a key frame (the first frame of the GOP) and by Wyner-Ziv frames (the remaining frames until the next key frame). The key frames, represented in Figure 2.10 by *K*, are intraframe encoded and decoded using a traditional H.263+ video coding standard. The Wyner-Ziv frames, *W*, are intraframe encoded but interframe decoded.

#### 2.3.1.1 Intraframe Encoding

There are two different encoding procedures in Figure 2.10, namely the encoding of key frames and the encoding of Wyner-Ziv frame. Every 8$^{th}$ frame is considered to be a key frame. If the frame to be encoded is a key frame, *K*, it is traditionally intraframe encoded using the H.263+ standard. If the frame to be encoded is a Wyner-Ziv frame, the first step to encode a Wyner-Ziv frame, *W,* is transform coding. An 8x8 block-based Discrete Cosine Transform (DCT) is applied over frame *W*. The transform coefficients of the whole frame *W* are then divided into a low frequency and a high frequency set. Only the low frequency coefficients are coded using Wyner-Ziv coding. The low frequency transform coefficients are grouped together to form coefficient bands $X_k$ in Figure 2.10, where $X_k$ is the $k^{th}$ transform coefficient band of *W*. Each transform coefficient band $X_k$ is then uniformly quantized with $2^{M_k}$ levels producing the quantized symbol stream $q_k$. Over the resulting quantized symbol stream $q_k$ associated to the transform coefficient band $X_k$, bitplane extraction is performed; this means, the $X_k$ band quantized symbols $q_k$ are converted to fixed-length binary codewords and corresponding bit-planes of the same importance are grouped together forming the corresponding bitplane array with $M_k$ bitplane vectors. Each bitplane is then independently fed into the turbo encoder, which plays the role of the Slepian-Wolf encoder in Figure 2.10, the Slepian-Wolf codec is built based on a Rate Compatible Punctured Turbo (RCPT) code structure [25]. The RCPT, combined with feedback, provides rate flexibility which is essential in adapting to the changing statistics between the side information and the frame to be encoded. The parity bits produced by the turbo encoder are stored in a buffer which transmits a subset of these parity bits to the decoder upon request. The parity bits sent from the encoder buffer constitute the Wyner-Ziv bits. For some bitplanes, however, the correlation with the side information is very small so the bit-plane is sent uncoded. The encoder stores for the quantized high frequency coefficients of the

previous frame. For a given block, the distance of the current coefficients from the corresponding quantized coefficients of the previous frame is calculated. If the distance is smaller than a threshold, a "no high frequency bits" codeword is sent. The encoder is no longer an intraframe coder because of the distance calculation. However, storing the quantized high frequencies of the previous frame is a negligible burden, compared to conventional frame store and encoder-based motion estimation. If the distance exceeds the threshold, the block's high frequency coefficients are compressed using run-length and Huffman coding and are sent to the decoder.

The encoder complexity of the proposed codec is similar to that of conventional intraframe encoding. For the Wyner-Ziv frames, turbo coding (composed of interleaving and convolutional coding) replaces conventional entropy coding. Storing a quantized version of the high frequency coefficients requires minimal memory and computation [23].

#### 2.3.1.2 Interframe Decoding

As shown in decoding architecture of Figure 2.10, one of two decoding procedures is performed, in accordance to the encoding procedure described in Section 2.3.1.1. For key frame *K*, a traditional intraframe decoder using the H.263+ standard is employed; the decoded key frame is then used in the next temporally adjacent Wyner-Ziv frame decoding process to generate $\hat{W}$ (an estimate of the W frame) by means of motion estimation. For the Wyner-Ziv frame *W*, the decoder performs frame extrapolation using the received high frequency bits and the previous reconstructed frame (Wyner-Ziv frame or key frame) to generate an estimate of frame W, called $\hat{W}$. For a given block of the current W frame, if no high frequency bits are sent, the corresponding block in the $\hat{W}$ frame is filled with the co-located samples block from the previous reconstructed frame i.e. the co-located block from the previous frame is used as the side information. If the decoder receives high frequency bits, the decoder reconstructs these coefficients and

utilizes them in a motion search based on the received high frequency bits to generate the best side information block $\hat{W}$ from the previous reconstructed frame. A block-based 8×8 DCT is then performed over the $\hat{W}$ frame to obtain the side information transform coefficient bands $Y_k$, corresponding to the transform coefficient bands $X_k$. To make the side information useful to the following stages (turbo decoding and reconstruction), a statistical dependence model between corresponding coefficients in $X_k$ and $Y_k$ must be considered. In [23], the authors assume that the residual distribution between the corresponding elements in $X_k$ and $Y_k$ is modeled by a Laplacian distribution. The decoded quantized symbol stream $q_k'$ associated to the transform coefficient band $X_k$ is obtained through a turbo decoding procedure. For each transform coefficient band, the turbo decoder starts decoding the most significant bitplane followed by the sequential decoding of the remaining bitplanes. Each transform coefficient band bitplane is decoded using the received Wyner-Ziv bits associated to that bitplane and the side information $Y_k$. When the received Wyner-Ziv bits together with $Y_k$ are not sufficient to provide a reliable decoding of the current bitplane, more bits are requested via the feedback channel by the decoder; the feedback channel is thus necessary to adapt to the changing statistics between the side information and the frame to be encoded. After the additionally requested Wyner-Ziv bits are received, a new attempt to decode the relevant bitplane is performed. The requests and following decoding operations are executed until the current bitplane error probability, $P_e$, is lower than $10^{-3}$; in this case, the turbo decoding of a transform coefficient band bitplane is considered to be successful. An ideal error detection capability is assumed at the decoder to determine the current bitplane error probability of a given transform coefficient band, i.e. the turbo decoder is able to measure in a perfect way the transform coefficient band current bitplane error probability. In general, due to the availability of the side information, the number of Wyner-Ziv bits required to

determine in which quantization interval (level) a transform coefficient is mapped to, $R_k$, from the $2^{M_k}$ possible levels, is lower than $M_k$ and thus compression efficiency achieved. Notice that the more accurate the side information is, the higher is the compression efficiency since fewer Wyner-Ziv bits are required to provide a reliable decoding. After turbo decoding the $M_k$ bitplanes associated to the DCT band $X_k$, the bitplanes are grouped together to form the decoded quantized symbol stream $q_k'$. Given the reconstructed quantized symbol stream $q_k'$ and the side information $Y_k$, the reconstruction of each transform coefficient band, $X_k'$, is computed through the conditional expectation $E(X_k \mid q_k', Y_k)$. Assuming that $q_k'$ is error-free, this reconstruction function has the advantage of bounding the magnitude of the reconstruction distortion to a maximum value, determined by the quantizer coarseness. This property is desirable since it eliminates large positive or negative errors for a given transform coefficient. These large errors tend to be very perceptible and annoying to the viewer [23]. After all low and high frequency coefficient bands are reconstructed, a block-based 8×8 inverse-DCT is performed and the reconstructed $W$ frame, $W'$, is obtained. For the blocks with no transmitted high frequency bits, the high frequency coefficients of the side information frame are used.

#### 2.3.1.3 Experimental results of Low-Complexity Video Coding Solution reported in literature

In order to evaluate the performance of the system described, the authors considered several video sequences at 10 fps, among which only Foreman sequence will be considered here. Figure 2.11 shows the rate-distortion performance.

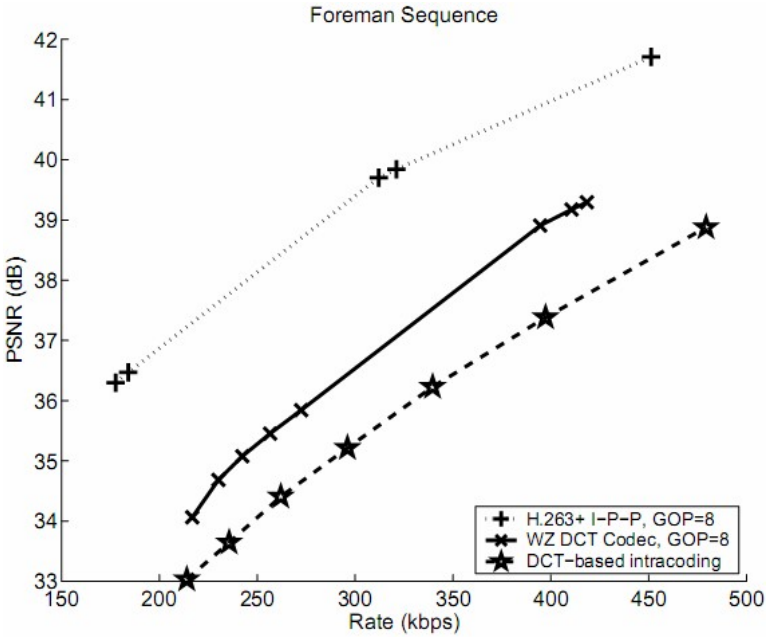


Figure 2.11: RD performance with Foreman sequence [23]

For the Foreman sequence, proposed codec introduces 1.5 dB improvement in PSNR (fixed bitrate) and about 15 to 20% in bit rate savings (fixed PSNR). The observed improvement over conventional intraframe coding is less, due to high motion throughout the sequence, thus, the accuracy of the side information generated is not adequate. Most of the high frequency coefficients are sent to decoder; therefore the performance is close to that of simply intracoding the frame. For all the sequences, the gaps from H.263+ interframe coding ranges from 2 to 3 dB.

#### 2.3.1.4 Remarks on Low-Complexity Video Coding Solution

In this architecture, only the low frequency coefficients of a frame are Wyner-Ziv coded. The high frequency coefficients are run-length coded and used at the decoder to achieve accurate motion estimation. Thus, recursive decoding a series of Wyner-Ziv frames by performing motion compensation of the previous frame to generate the side information is

possible like the I-P-P structure used in conventional interframe video coding. The I-P-P dependency is only meaningful at the decoder because the frames are still encoded independently at the encoder. This codec shows impressive gains over conventional DCT-based intraframe coding while having comparable encoding complexity. There is still a performance gap from H.263+ interframe coding.

### 2.3.2 Residual Video Coding Solution

Residual Video Coding solution is the most recent scheme proposed by Stanford Image, Video, and Multimedia Systems Group. Since it is mainly based on pixel domain Wyner-Ziv Coding Structure [12] which is proposed by the same group, only major differences in the architecture will be mentioned in this section.

In this study, the previous Wyner-Ziv video codec [12] is extended by Wyner-Ziv encoding the residual of a frame with respect to an available reference frame at the encoder. For the sake of low-complexity requirement no motion compensation is performed for finding frame difference. Wyner-Ziv coding is applied to the pixels of frame and the previous frame pixels are used as encoder reference. The encoder simply exploits some of the similarities between the current frame and the previous frame, while the decoder can use both the previous frame and the more computationally complex motion-compensated side information for decoding. Authors state that, allowing the encoder perform frame storing and frame subtraction, the pixel domain Wyner-Ziv residual coder achieves better rate-distortion performance compared to their previous pixel-domain schemes and similar performance as transform-domain Wyner-Ziv video codecs. [20]

The architecture proposed is illustrated in Figure 2.12

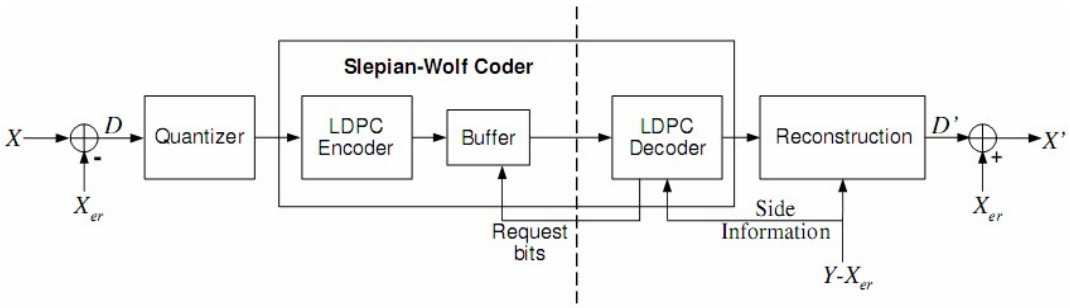


Figure 2.12: LDPC-based Wyner-Ziv residual video codec [20]

The encoder reference frame $X_{er}$, can be any frame that is easy to generate by encoder and to avoid drift at decoder, $X_{er}$ should be replicable by decoder. Thus, the selection of $X_{er}$ can be the reconstructed previous frame or the average of two or more adjacent reconstructed frames [20]. The generation of $X_{er}$ brings extra complexity to encoder, when compared to [12], due to frame storing and frame subtraction operations. However, $X_{er}$ can be considered as second side information which is available both at encoder and the decoder. If $X_{er}$ and $X$ are statistically dependent to $Y$, then using $X_{er}$ can reduce the encoding rate of the system. [20]

$Y$, the side information is generated at the decoder by computationally complex motion-based techniques. The Wyner-Ziv decoder uses $Y$ and $X_{er}$, for the reconstruction of $X'$.

### 2.3.2.1 Encoding Procedure

At encoder, difference between $X$, current frame, and $X_{er}$, the encoder reference frame is quantized using a deadzone uniform scalar quantizer. The bit-planes of the quantized symbols are encoded by a low-density parity check (LDPC) code and the accumulated syndrome bits are stored in an encoder buffer as in [27]. As described in [27], the accumulation of the syndrome bits allows rate-adaptivity which is necessary for varying frame statistics. The encoder transmits a subset

of these bits to the decoder upon request. At this point, as in [19], hash information can be calculated and sent to aid the decoder in performing motion estimation.

#### 2.3.2.2 Decoding Procedure

The decoder generates the side information *Y* by applying one of several techniques, namely, motion-compensated interpolation, motion-compensated extrapolation or hash-based motion estimation on previously reconstructed adjacent frames. It also reconstructs the encoder reference frame $X_{er}$. Decoding of the bit-planes is performed iteratively starting with the most significant bit-plane. LDPC decoder takes the received subset of accumulated syndrome bits corresponding to the bit-plane and the residual side information $Y\text{-}X_{er}$ to decode the current bit-plane. LDPC decoder may request additional accumulated syndrome bits from buffer, if received bits can not be reliably decoded. This feedback mechanism is used, until an acceptable probability of symbol error is achieved. When all the bit-planes are decoded, the decoded symbols and the residual side information $Y\text{-}X_{er}$ are used by the reconstruction block to reconstruct *D'*. To generate *X', D'* is added to $X_{er}$.

#### 2.3.2.3 Experimental results of Stanford Residual Video Coding Solution reported in literature

The simulations have been performed on the first 100 frames of the Salesman and Foreman QCIF sequences at 15 fps. For the pixel-domain Wyner-Ziv schemes, a rate-adaptive LDPC accumulate code of block length 25344 bits. For the DCT-domain systems, the LDPC accumulate code applied to each coefficient band has a block length of 396 bits. In both cases key frames are H.263+ DCT-based intracoded. Results of simulation sets can be grouped in i) Side Information Generation by Motion-Compensated Interpolation ii) Hash-based Side Information Generation.

**i) Side Information Generation by Motion-Compensated Interpolation**

Three Wyner-Ziv coding schemes, namely, pixel-domain Wyner-Ziv residual coding, pixel-domain Wyner-Ziv coding and DCT-domain Wyner-Ziv coding, are compared in Figure 2.13. In every three case, odd frames of the sequence are Wyner-Ziv frames, while the even ones are key frames (i.e. GOP size is two). Side information is generated by frame interpolation of the previous and next key frames.

a) Pixel-domain Wyner-Ziv residual coding: The frame is encoded using the architecture illustrated in Figure 2.12. The previous reconstructed key frame has been selected as the $X_{er}$, encoder reference.

b) Pixel-domain Wyner-Ziv coding: The frame is encoded using the architecture proposed in [12].

c) DCT-domain Wyner-Ziv coding: The frame is encoded using the architecture proposed in [16]. An 8x8 DCT is applied to the frame and the ten lowest frequency coefficient bands are independently compressed using Wyner-Ziv coding. High frequency coefficients are quantized and compressed with H.263+ zero run-length coding and entropy coding.

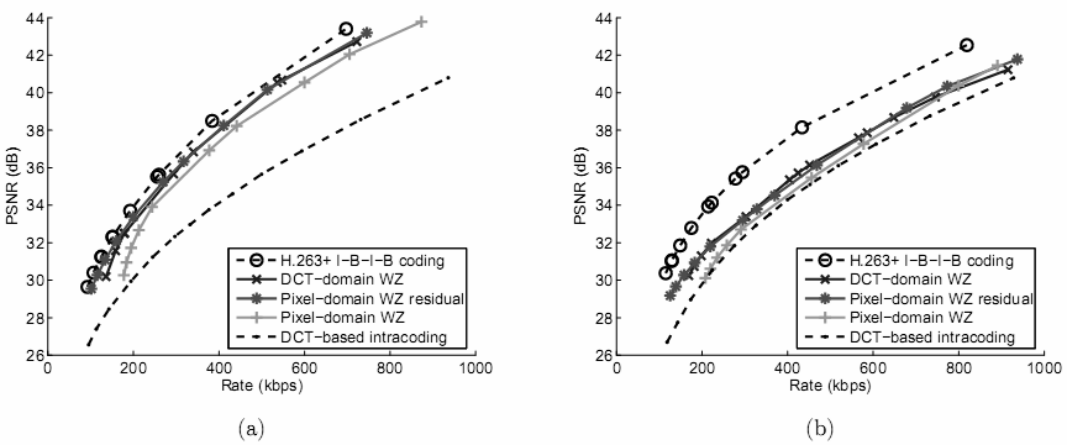


Figure 2.13: Rate vs. PSNR curves for (a) Salesman, (b) Foreman, GOP = 2 [20]

RD performance of these schemes are compared with the H.263+ intraframe coding (all I frames) and H.263+ interframe coding with an I-B- I-B predictive structure. The rate and PSNR values in Figure 2.13 are averaged over both key frames and Wyner-Ziv frames. Results show that, the pixel-domain Wyner-Ziv residual coding scheme has slightly better compression performance (less than 1 dB in most cases) than the non-residual pixel-domain system. For (a), all Wyner-Ziv coding schemes have significantly better RD performance than intraframe coding. However, for (b), which has high motion throughout the frame, this is not the case. [20]

**ii) Hash-based Side Information Generation**

Three Wyner-Ziv coding schemes, namely pixel-domain Wyner-iv residual coding with pixel residual hash, pixel-domain Wyner-Ziv coding with pixel hash and DCT-domain Wyner-Ziv coding with DCT hash are compared in Figure 2.14. In three cases, every $8^{th}$ frame is considered as a key frame, while the remaining frames are Wyner-Ziv coded, i.e. GOP size is eight. For every Wyner-Ziv frame, a hash is generated by encoder as in [19]. The hash mechanism, i.e. the sending decision of hash and the behavior of decoder upon receiving hash, is same as in [19].

a) Pixel-domain Wyner-Ziv residual coding with pixel residual hash: The hash for each block is a quantized subsample of the pixels in $Y$-$X_{er}$. Previous frame has been selected as $X_{er}$. The entropy of the sent hash symbols are calculated and added to the rate shown in Figure 2.14. To encode the Wyner-Ziv frames the system in Figure 2.12 is used, with the previous frame as encoder reference. [20]

b) Pixel-domain Wyner-Ziv coding with pixel hash: A quantized subsample of $X$ is used as a hash in this scheme. The entropy of the sent hash symbols are calculated and added to the rate shown Figure 2.14. Each Wyner-Ziv frame is encoded using the simple pixel-domain Wyner-Ziv system [12].

c) DCT-domain Wyner-Ziv coding with DCT hash: The Wyner-Ziv scheme described in [23] is used in this scheme. An 8x8 DCT is applied to the frame and the ten lowest frequency coefficient bands are independently compressed using Wyner-Ziv coding. High frequencies are quantized and compressed using H.263+ zero run-length coding and entropy coding. These high frequencies, if sent, serve as the hash at the decoder. [20]

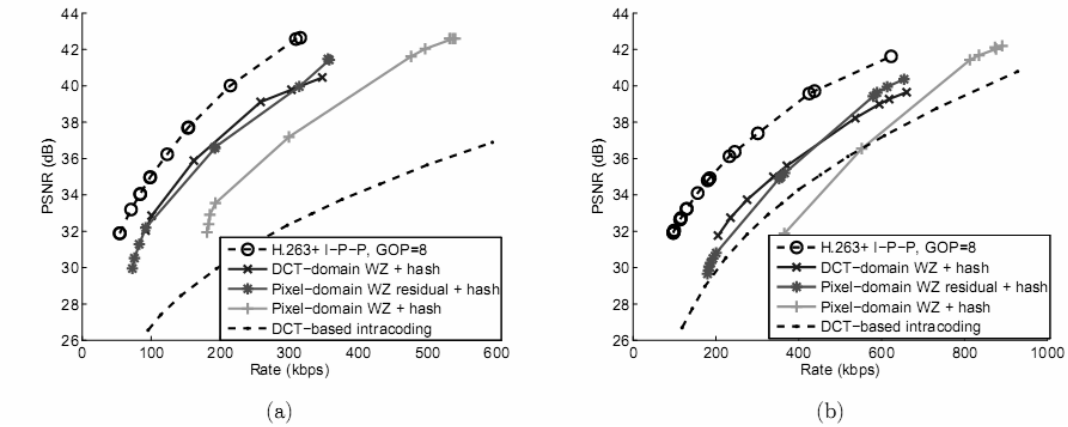


Figure 2.14 Rate vs. PSNR curves for (a) Salesman (b) Foreman, GOP = 8 [20]

RD performance of these schemes are compared with the H.263+ DCT-based intraframe coding and H.263+ interframe coding with an I-P-P predictive structure with GOP Size = 8. Results show that, the pixel-domain Wyner-Ziv residual scheme has significantly better performance (2 to 3 dB) than the original pixel-domain system. Authors state that this performance improvement is due to both reduction of the hash bitrate and the Wyner-Ziv bits sent for a given quantizer step size. Both set of simulation results show that, pixel-domain Wyner-Ziv residual coding exhibits similar compression performance as DCT-

domain Wyner-Ziv coding with the advantage of not needing the optimal rate allocation between different coefficient bands. [20]

#### 2.3.2.4 Remarks on Stanford Residual Video Coding Solution

In this work Wyner-Ziv coding is performed on the residual pixels of a frame with respect to a simple encoder reference. In this scheme, the encoder can exploit part of the temporal and spatial redundancies among successive frames, while the decoder uses more sophisticated motion estimation techniques to conditionally decode the frames using better side information. The scheme requires the additional complexity of frame store and frame subtraction at the encoder, but demonstrates an improvement in compression efficiency (up to 3 dB) compared to simply Wyner-Ziv encoding the pixels of a frame. [20]

### 2.3.3 Robust Video Coding Solution

In order to achieve more robustness to transmission errors in distributed video coding, Stanford Image, Video, and Multimedia Systems Group proposed [13], [15] and [21] to protect a video bitstream from transmission errors. The results obtained with these approaches show that an additional Wyner-Ziv bitstream can be used to simultaneous achieve strong protection against channel errors and graceful degradation of the video quality.

The most recent solution proposed by this group is [26] and illustrated in Figure 2.15. The goal of this solution is to describe a scheme for error-resilient video broadcasting, using Wyner-Ziv coding, instead of conventional forward error correction, which can achieve graceful degradation of the decoded video quality without the need for a layered representation.

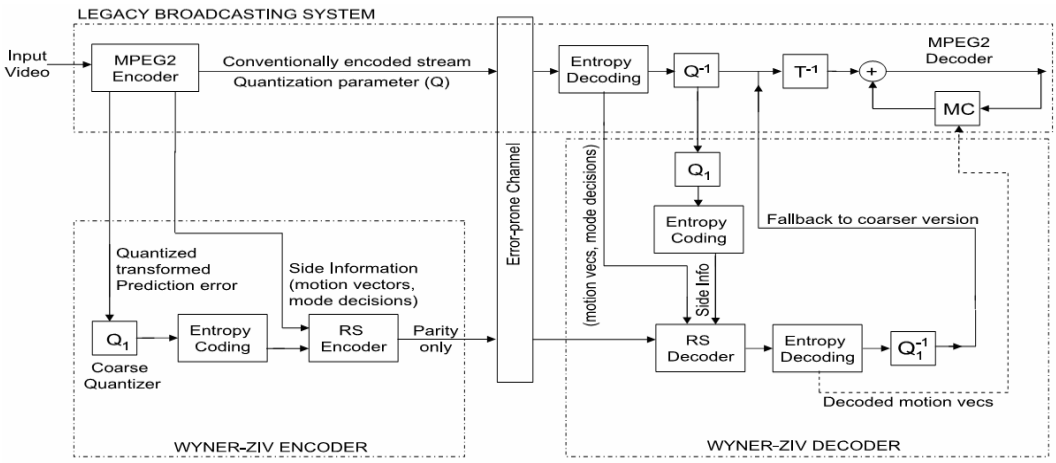


Figure 2.15: Systematic forward error protection (FEP) by combining MPEG coding and Reed-Solomon (RS) codes across slices [26]

Systematic lossy Forward Error Protection (FEP) scheme is the terminology used in the literature to refer to the coding architecture in Figure 2.15. For the concept of systematic lossy forward error protection MPEG2 video compression is used as an example.

### 2.3.3.1 FEP Encoding

At the transmitter, the input video sequence is compressed by an MPEG-2 video coder. Since the MPEG-2 video bitstream is generated without consideration of the error resilience provided by the Wyner-Ziv coder, the overall scheme is called as systematic source-channel coding [26]. Input video sequence, is also encoded with a Wyner-Ziv encoder independent of MPEG-2 Encoding. This encoder uses a coarse quantizer, entropy coder and a Reed-Solomon (RS) encoder [15]. Coarse quantizer operates on the quantized transformed prediction error signal generated by the conventional MPEG-2 encoder. The output of the coarse quantizer is entropy coded. After entropy coding procedure, the resulting bitstream (i.e. the entropy coded output of the coarse quantizer along with the motion vectors and mode decisions

inherited from the conventional MPEG-2 encoder) is input to a Reed-Solomon (RS) encoder which applies systematic RS codes with byte-long symbols, across the slices of an entire frame. *Only the RS parity symbols* are then transmitted to the receiver, and these constitute the Wyner-Ziv bitstream. The systematic portion of the RS encoder output is discarded. If there are no transmission errors, the RS parity symbols do not provide any additional information. (i.e. similar to the inefficiency of traditional forward error correction when there are no errors.) When transmission errors occur, the decoder generates a coarsely quantized version of the received prediction error signal. This coarse version is actually an error-prone copy of the Wyner-Ziv description, which serves as side information for the RS decoder. [26]

#### 2.3.3.2 FEP Decoding

The compressed legacy system bitstream received is decoded through operations corresponding to the inverse of those performed in the MPEG-2 encoder. Due to channel transmission errors, some slices of input video may be incorrectly decoded. Previous frame error concealment is then used to conceal erroneous slices however some errors still subsist after the error concealment operation. The RS decoder uses the parity symbols and error-prone Wyner-Ziv description to obtain the error-free Wyner-Ziv description. Since the location of the lost slices is known, the RS decoder can perform erasure decoding across the error-prone slices. The erroneous slices can be “filled” with their correct but coarser versions. Transmission errors are therefore corrected up to a certain residual distortion. This is called the "fallback" mechanism [26]. After decoding, the coarse fallback causes some prediction mismatch which propagates to the subsequent frames, but visual examination of the decoded sequence shows that this small error is imperceptible. Thus, the receiver obtains a video sequence of superior visual quality. This system includes FEC as a special case, if the “coarse” quantizer uses the same quantization parameter as the main MPEG-2 encoder. This scheme applies Wyner-Ziv decoding to the

received prediction error signal, as opposed to generating side information by re-encoding the output of the MPEG decoder, as was done in [15]. Authors state that the decoder implementation to be very simple, with the coarse quantizer ($Q_1$ in Figure 2.15) and extra entropy coding adding a negligible complexity overhead with respect to conventional FEC techniques [26].

#### 2.3.3.3 Remarks on Stanford Robust Video Coding Solution

In this work a practical scheme which applies Wyner-Ziv coding ideas for error-resilient digital video broadcasting is presented. Experimental results show that a supplementary bitstream generated using Wyner-Ziv coding of the source sequence can be used to correct transmission errors in the received video signal, up to a certain residual distortion. In return for some imperceptible residual distortion in the case of channel errors, the above scheme can potentially achieve a much lower bitrate than a conventional channel coder which protects the bits produced by the source coder. Equivalently, if higher distortion is allowed, stronger error protection can be achieved at the same bitrate [26]. On the other hand, generation of supplementary bitstream brings additional complexity to the encoder, which compromises the motivation of low-complexity encoding. This scheme is suitable when robustness to channel errors is the main concern and encoder complexity is not the main issue.

## 2.4 DISCOVER Project

DISCOVER [28] is a project under European Commission IST FP6 programme and has been devoted to the advancement of Distributed Video Coding. In this section, the monoview Wyner-Ziv Video codec architecture of DISCOVER (**DIS**tributed **CO**ding for **V**ideo s**ER**vices) [11] will be explained. Since the implemented Wyner-Ziv Video Codec within the context of this thesis is based on DISCOVER, architecture and encoder / decoder specific tools of DISCOVER will be briefly

mentioned in this section. A detailed analysis and definition for each building block will be given in following chapter. The DISCOVER came out with the objectives such as proposing new video coding schemes and tools in the area of Distributed Video Coding with a strong potential for new applications in mind. Error resiliency, scalability and coding efficiency were other important key points in the evolution of DISCOVER. The basic architecture used in this project is given in Figure 2.16.

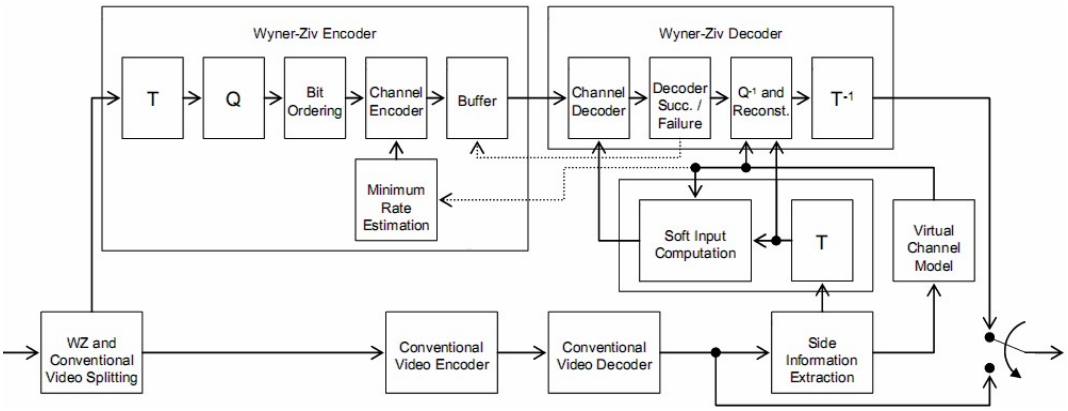


Figure 2.16: DISCOVER Codec Architecture [11]

The details of the DISCOVER project will be given in Chapter 3 since we have implemented a codec based on DISCOVER architecture within the context of this thesis.

### 2.4.1 Experimental results of DISCOVER reported in literature

To obtain the RD plots in Figure 2.17 and Figure 2.18, "Hall Monitor" and "Foreman" sequences have been used, respectively. Both sequences are QCIF size and frame rate is 15 fps. GOP Size is selected as two, i.e. odd frames are key frames and even frames are Wyner-Ziv. Key frames are always encoded with H.264/AVC Intra (Main Profile).

The RD performance of DISCOVER has been compared with standard low complexity encoders together with H.263+ Intra and H.264/AVC Intra, which are two very well known codecs where no temporal correlation between sequence frames is exploited. DISCOVER has also been compared with the H.264/AVC with no motion (IBI GOP structure), which has a lower encoding complexity compared to the full H.264/AVC Inter codec [11]. Some DISCOVER codec RD performance results are presented in Figure 2.17 and Figure 2.18.

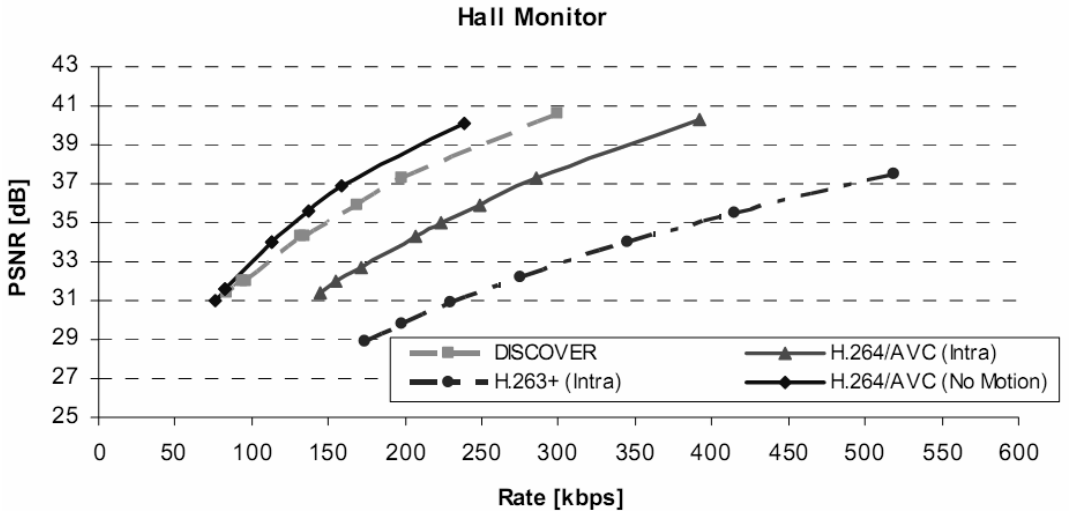


Figure 2.17: DISCOVER codec RD performance for Hall Monitor sequence, GOP = 2 [11]

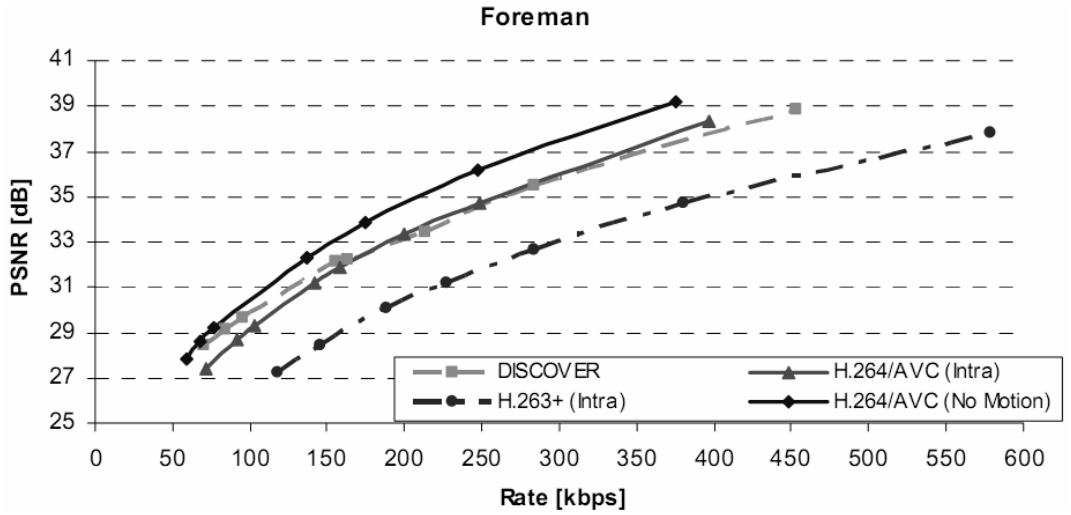


Figure 2.18: DISCOVER codec RD performance for Foreman sequence, GOP = 2 [11]

As illustrated in Figure 2.17, DISCOVER scheme brings up to 8 dB PSNR improvement for the sequence "Hall Monitor" when compared to H.263+ Intra and 3 dB improvement for H.264/AVC Intra. However for "Foreman" sequence, the RD performance of DISCOVER, illustrated in Figure 2.18, is 0.5 dB off H.264/AVC Intra while it is 3 dB improved for H.263+ Intra. Considering the results obtained, authors state that the DISCOVER WZ codec can exploit the temporal correlation in an efficient way while using a rather simple encoder and still be competitive when compared to the more complex H.264/AVC Intra encoder [11]. More simulation results can be obtained from the project's web page [28].

## 2.5 Final Remarks on DVC Solutions proposed in Literature

Emerging applications have requirements such as low-complexity and low-power consumption at the encoder and these are quite different from those targeted by MPEGx and H.26x standards. This has stimulated the evolution of a new coding paradigm, Wyner-Ziv video

coding – a particular case of distributed video coding which refers to the case where each video frame is encoded independently (intraframe coding), but decoded conditionally (i.e. interframe decoding). Several solutions have been proposed in the literature and the achieved results are promising for some applications where low encoding complexity is the major goal. Before selecting the architecture for the implementation within the context of this thesis, three most relevant and most cited solutions proposed in the literature have been investigated; i) PRISM (Section 2.2 ) and ii) Feedback-channel based Stanford Solutions ( Section 2.3) and iii) DISCOVER (Section 2.4). All codecs evolved with the requirements of emerging applications, such as low-complexity and low-power consumption at the encoder, in mind. However, due to lack of published computational complexity analysis of PRISM and Stanford Solutions, complexity comparison metrics between solutions is not well defined. Therefore, special emphasis has been given to the rate-distortion performances of each solution. Rate-distortion results of PRISM show a performance between conventional intraframe transform coding and conventional motion-compensated interframe transform coding similar to the results obtained by Stanford Solutions [29] and DISCOVER. Considering published rate-distortion performance plots, solutions proposed by Stanford Group are comparable with the DISCOVER results. Figure 2.13(b) (given in Section 2.3.2.3) and Figure 2.18 (given in 2.4.1), show the PSNR values at the same rate (*350 kbps*) for comparison. Both results have been obtained using "Foreman" QCIF sequence at 15 fps. In both cases GOP is two, i.e. odd frames of the video sequence are key frames and even ones are Wyner-Ziv frames. In Figure 2.13(b) key frames are intracoded as I frames using a standard H.263+ codec. In Figure 2.18 key frames are encoded with H.264/AVC Intra (Main profile). In both cases, the Wyner-Ziv frames are Wyner-Ziv encoded and the side information at the decoder is generated using motion-compensated interpolation of the previous and next key frame. Figure 2.13(b) shows

the Rate vs. PSNR results given in [20]. The rate-distortion performance of several solutions proposed by Stanford Group have been compared to H.263+ intraframe coding (all I frames) and H.263+ interframe coding with an I-B-I-B predictive structure. The rate and PSNR values are averaged over both key frames and Wyner-Ziv frames. In the legend "DCT-domain WZ" stands for the solution in [16], "Pixel-domain WZ residual" stands for the solution in [20] and "Pixel-domain WZ" stands for the solution in [12]. "The best performance" solution in the figure is "DCT-domain WZ" ([16]) for the bit rate interval of interest. The difference between H.263+ intraframe coding is at about 1 dB for most of the bit rates. Figure 2.18 shows the Rate vs. PSNR results for DISCOVER [11]. The DISCOVER codec is compared with standard low complexity encoders. On one hand, H.263+ Intra and H.264/AVC Intra, since they are two very well known codecs where no temporal correlation is exploited. On the other hand, H.264/AVC with no motion (I-B-I GOP structure), which has a lower encoding complexity compared to the full H.264/AVC Inter codec. Performance gains are up to 3 dB for the rate interval of interest when compared to H.263+ Intra. In a nutshell, rate-distortion performance of DISCOVER [11], which is based on Stanford DVC Solution [29], not surprisingly, is better than that of Stanford solution [16] for comparable bit rates. In addition to that, DISCOVER

- is the most recent DVC solution proposed in the literature,
- has the most detailed specifications published,
- executable binary is available,
- has the most detailed evaluation performed and
- is a well-suited testing framework with its modular architecture.

These are the main reasons for DISCOVER being selected as the architectural implementation reference within the context of this thesis.

# CHAPTER 3

# IMPLEMENTED DISTRIBUTED VIDEO CODEC

In Chapter 2, the most relevant Wyner-Ziv video coding solutions presented in the literature, e.g. PRISM [18], hash-based Wyner-Ziv video coding [23], Wyner-Ziv residual video coding [20] and DISCOVER Project [11] are described. The experimental results illustrated in Chapter 2 show that Wyner-Ziv video coding provides promising coding solutions for applications where low encoding complexity is a major goal. This chapter is focused on describing the Wyner-Ziv Video Codec implemented within this thesis which is based on the approach proposed by Artigas *et al*. in [11].

## 3.1 Basic Architecture on Implemented Codec

Figure 3.1 illustrates the architecture of the implemented distributed video codec. The general architecture of this solution is similar to the DISCOVER architecture. However, in the implemented codec there are some simplifications in some modules, which do not significantly effect overall PSNR performance, to further decrease encoder and decoder complexity, such that, Wyner-Ziv and key frame splitting is performed on constant GOP size basis, minimum rate estimation is omitted which does not effect side information quality, constant channel modeling is employed instead of adaptive channel modeling and LDPC encoder parameters have been optimized in order to improve compression efficiency of the overall system.

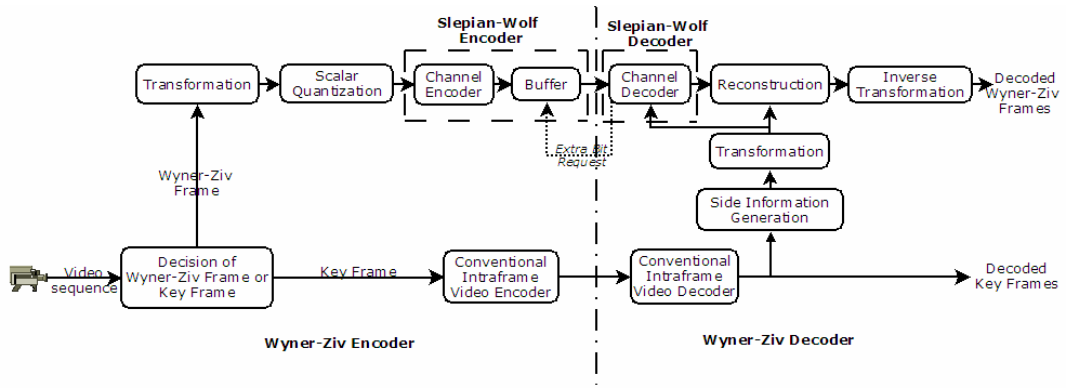


Figure 3.1: Basic Architecture of Implemented Distributed Video Codec

Briefly, the coding procedure illustrated in Figure 3.1 is as follows: the video sequence to be encoded is divided into two types of frames, namely, key frames and Wyner-Ziv frames. In this architecture, GOP Size is two, i.e. odd frames of the sequence to be encoded are key frames and even frames of the sequence are Wyner-Ziv frames. Each key frame is encoded conventionally by H.264/AVC Intra encoder and transmitted to the encoder side. Over each Wyner-Ziv frame, a 4x4 block-based discrete cosine transform (DCT) is applied. Transformed coefficients -DCT coefficients- of the entire frame are then organized in bands where every DCT coefficient band contains the coefficients associated to the same frequency in different blocks. There are $4^2=16$ different DCT coefficients band that can be formed since 4x4 block based DCT transform is used. Each obtained transformed DCT coefficients band is then uniform scalar quantized according to eight predefined sets of parameters [31]. The bits representing band-separated coefficients are ordered in bit planes and fed into a systematic channel encoder. The channel encoder, also known as the Slepian-Wolf encoder, has been selected to use the rate-compatible LDPC Accumulate (LDPCA) codes [32]. Channel encoder generates a set of syndrome bits of the encoded bit planes. Discarding the systematic bits, channel encoder stores these syndrome bits in a

buffer. The stored bits are progressively transmitted to the decoder upon reception of extra bit request from decoder. The decoder may ask for more bits via feedback channel for successful decoding operation.

The decoding operation is more computationally complex due to the motion estimation process which exploits the temporal correlation between frames for side information generation. Firstly, key frames which have been conventionally encoded with H.264/AVC Intra are decoded using the same method. Thus, decoded key frames are obtained. The decoded key frames are used for generating side information which plays an important role in Wyner-Ziv video decoding. Side information is an estimate of the current Wyner-Ziv frame to be decoded. The more correlated side information and given Wyner-Ziv frame are the higher rate-distortion performance of Wyner-Ziv decoding is. In order to generate the side information for a given Wyner-Ziv frame, motion compensated frame interpolation between two closest key frames is performed. For the case, where GOP is two, closest key frames are the adjacent frames for the given Wyner-Ziv frame. The difference between original Wyner-Ziv frame and corresponding side information can be considered as the correlation noise in a virtual channel. Virtual channel model has been selected as a Laplacian model, which yields a good approximation of the residual Wyner-Ziv and side information distribution [30]. The same 4x4 block-based discrete cosine transform (DCT) used at the encoder is applied to the interpolated frame, i.e. generated side information, in order to obtain the coefficients of the Wyner-Ziv frame. The transformed side information coefficients together with virtual channel model are used by channel decoder (LDPC decoder) for proper Wyner-Ziv decoding. Channel decoder can, iteratively; request for more parity bits using the feedback channel until a predetermined distortion criterion is satisfied. This request-and-decode iteration continues until parity bits are successfully decoded. The decoding is considered as successful if the current bitplane error probability for a DCT band is lower than or equal

to a given error probability threshold. Successfully decoded syndrome bits are used together with side information coefficients and virtual channel model in order to obtain the reconstructed coefficients. Reconstructed Wyner-Ziv coefficients are then inverse 4x4 block-based discrete cosine transformed. Thus, decoded Wyner-Ziv frames are obtained. Using decoded key frames and Wyner-Ziv frames, decoded video sequence is obtained. The detailed description of each architecture block will be given in the following sections.

## 3.2 Decision of Wyner-Ziv Frame or Key Frame

For side information generation, several techniques have been proposed in the literature such as side information generation from motion compensated frame interpolation and hash-based side information generation [20]. As a starting point, motion compensated frame interpolation techniques has been selected for the implementation using successive groups of a fixed number of pictures, i.e. using a fixed GOP (Group of Pictures). As can be seen from the results in [31], highly dependent on the input video sequence characteristics, using variable GOP size does not guarantee better rate-distortion results by better exploiting temporal correlation in the sequence. Considering results achieved by [31], using fixed GOP length, i.e. two, has been the selected approach in the implementation in order to further reduce the encoder complexity. Every odd frame of the input video sequence is selected key frame and every even frame is a Wyner-Ziv frame. The key frames are encoded with H.264/AVC Intra (Main profile). In order to have almost constant decoded video quality for the whole sequence, consisting of multiplexed key and Wyner-Ziv frames, quantization steps for the H.264/AVC Intra encoded key frames have been selected to match the average PSNR of the Wyner-Ziv frames. Quantization matrices for the Wyner-Ziv frames are indexed with $Q_i$'s and will be mentioned in Section 3.4.

The key frames are coded with constant quantization parameters, QP, and have been obtained for GOP = 2, QCIF@15 Hz as defined in Table 3.1. [31]

Table 3.1: Selected QP values for different $Q_i$'s [31]

| | Qi=1 | Qi=2 | Qi=3 | Qi=4 | Qi=5 | Qi=6 | Qi=7 | Qi=8 |
|---|---|---|---|---|---|---|---|---|
| Foreman | 40 | 39 | 38 | 34 | 34 | 32 | 29 | 25 |
| Hall Monitor | 37 | 36 | 36 | 33 | 33 | 31 | 29 | 24 |
| Coastguard | 38 | 37 | 37 | 34 | 33 | 31 | 30 | 26 |
| Soccer | 44 | 43 | 41 | 36 | 36 | 34 | 31 | 25 |

## 3.3 Transformation

The purpose of using a block-based transform in the Wyner-Ziv video coding architecture is the same with its usage purpose in traditional video coding schemes: i) to decorrelate block samples by exploiting the spatial redundancy between neighboring samples, and ii ) to compact the block energy into as few transform coefficients as possible. The selected transformation technique for the implementation is the Discrete Cosine Transformation since it is widely used by the state-of-the-art traditional video coding standards, from the H.261 to the H.264/MPEG-4 AVC standards as well as the all transform domain Wyner-Ziv solutions proposed in the literature [11], [16], [19] and [23].

Wyner-Ziv frames are first transformed using a 4×4 DCT. The $4^2$=16 correlated samples inside the 4×4 block are converted into 16 independent DCT transform coefficients or bands, $b_k$'s, in the spatial frequency domain. The DCT coefficients which are arranged in a 4×4 block called the DCT coefficients block, which is illustrated in Figure 3.2.

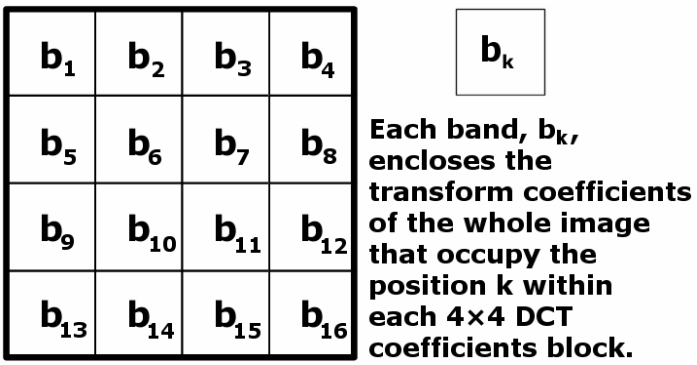


Figure 3.2: Bands, $b_k$'s, inside a 4x4 DCT coefficients block

The top-leftmost DCT coefficient, $b_1$, is called the DC coefficient or band and corresponds to the spatial frequency zero. The remaining 15 coefficients are known as AC coefficients or bands and correspond to non-zero spatial frequencies; the AC coefficient located at bottom-rightmost DCT coefficient corresponds to the highest spatial frequency. The bottom-rightmost DCT coefficient, $b_{16}$, corresponds to the highest coefficients band.

## 3.4 Quantization and Bit Ordering

Each DCT coefficients band, $b_k$, is independently quantized using a predefined number $2^{M_k}$ of levels, where $M_k$ is the number of bits needed to map a DCT coefficient band value into one of $2^{M_k}$ quantizer levels associated to that band. Depending on the target quality for the Wyner-Ziv frame, different performances can be achieved by changing $M_k$ value for the DCT band $b_k$ since each $M_k$ value has a certain rate-distortion associated to it. The DC coefficients band, $b_1$, is characterized by high amplitude positive values since each DC transform coefficient expresses the average energy of the corresponding 4×4 samples block. The quantization for the DC coefficients is performed using a uniform

scalar quantizer which is illustrated in Figure 3.3. The data range for DC band is assumed as [0, $2^8$).

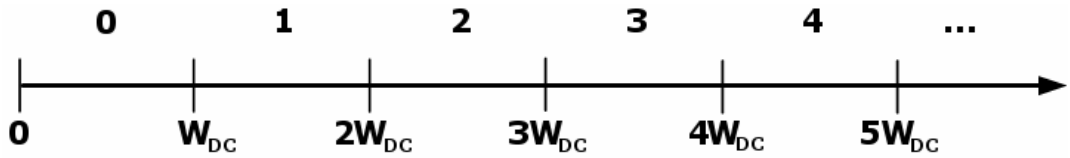


Figure 3.3: Uniform Scalar Quantizer with quantization step size $W_{DC}$

The range for the *q*-th quantization interval is $I^q{}_{DC}$

$$I^q{}_{DC} = [qW_{DC}, (q+1)W_{DC}) \qquad (3.1)$$

The DC quantization step size is $W_{DC}$

$$W_{DC} = 2^{(8-M_1)} \qquad (3.2)$$

$M_1$ is the number of bits reserved for each quantized value of the DC band (i.e. *band 1*). Here, the same approach with [11] is used with different data range. For AC bands, the quantizer input assumes both positive and negative values since the basis functions associated to the AC coefficients present zero mean values [33]. A dead-zone quantizer with doubled zero interval, which is illustrated in Figure 3.4, is used for the quantization of AC bands. To obtain the quantization step size for the AC bands $b_k$, $k$=2, 3, … , 15, 16, the maximum absolute value, i.e. maximumValueOf*b*, within each band $b_k$ is first determined.
As in [11], the data range [-maximumValueOf*b*, maximumValueOf*b*) is calculated separately for each *b*-th band, where $b > 1$ for AC bands.

The quantization step size is

$$W_b = \left\lceil \frac{2.maximumValueOfb}{2^{M_b}} \right\rceil \qquad (3.3)$$

where $M_b$ is the number of bits reserved for each quantized value of the $b^{th}$ band, and the quantization intervals are defined as follows:

$$I_b^q = \begin{cases} \left[(q-1)W_b, qW_b\right) & q < 0 \\ \left[-W_b, W_b\right) & q = 0 \\ \left[qW_b, (q+1)W_b\right) & q > 0 \end{cases} \qquad (3.4)$$

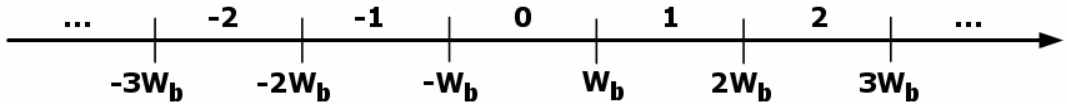


Figure 3.4: A dead-zone quantizer with doubled zero interval with step size $W_b$

As in [31], 4×4 quantization matrices for DCT coefficients, corresponding to the considered eight rate-distortion points in simulation results, will be referred as $Q_i$, i.e. *Quantization Index*, with i= 1, …, 8; the higher is $Q_i$, the higher are the bitrate and the quality.

For $Q_i$ = 1, 2, 3 and 4 respectively,

$$\begin{matrix} 16 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \quad \begin{matrix} 32 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \quad \begin{matrix} 32 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \quad \begin{matrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{matrix} \qquad (3.5)$$

For $Q_i$ = 5, 6, 7 and 8 respectively,

$$\begin{matrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \\ 4 & 4 & 0 & 0 \end{matrix} \quad \begin{matrix} 64 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \\ 4 & 4 & 0 & 0 \end{matrix} \quad \begin{matrix} 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{matrix} \quad \begin{matrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{matrix} \tag{3.6}$$

The DCT coefficients bands numbering, i.e. the order by which the DCT bands are encoded, is illustrated in Figure 3.2. Within a 4×4 quantization matrix, the value at position *k* in Figure 3.2 indicates the number of quantization levels associated to the DCT coefficients band $b_k$. The quantization matrices depicted in (3.5) and (3.6) are assumed to be known by both the encoder and decoder. In the quantization matrices, the value 0 means that no Wyner-Ziv bits are transmitted to the decoder for the corresponding bands; the decoder will replace the DCT bands to which no Wyner-Ziv bits are sent by the corresponding side information DCT coefficients bands determined at the decoder. As mentioned in Section 3.2, QP values for H.264/AVC Intra coding of key frames and quantization matrices for DCT coefficients are matched.

After quantizing the DCT coefficients band $b_k$, the quantized symbols (represented by integer values) are converted into a binary stream. The quantized symbols bits of the same significance, i.e. same band, (for instance, most significant bit) are grouped together forming the corresponding bitplane array of $M_k$ bit planes and fed to the LDPC encoder.

## 3.5 Slepian-Wolf Encoding / Decoding

Several schemes for channel encoding and decoding have been proposed in literature based on turbo Codes [25] and the rate-compatible LDPC Accumulate (LDPCA) codes [32]. Varodayan *et al* [32] stated that the LDPCA codes better approach the capacity of a variety

of communication channels, including the virtual channel in DVC, when compared to the turbo codes. Results achieved with DISCOVER Solution, illustrated in [31], Liveris *et al* [34] and [35] show that a channel coder based on LDPC codes outperforms a channel coder using turbo codes in rate distortion and complexity performance for all sequences using same sizes of GOP. The channel encoder used in this implementation, i.e. Slepian-Wolf encoder, is a rate-compatible LDPC Accumulate (LDPCA) coder proposed by [32], which is available in [36]. However, first practical implementations of distributed video codec within the context of this thesis were based on turbo coding thus, for the sake of completeness, a brief definition of a turbo coder and decoder will be given.

### 3.5.1 Turbo Coding / Decoding

A turbo coder, illustrated in Figure 3.5, is a parallel concatenation of two identical Recursive Systematic Convolutional (RSC) Encoders. A Pseudo-random interleaver separates the code.

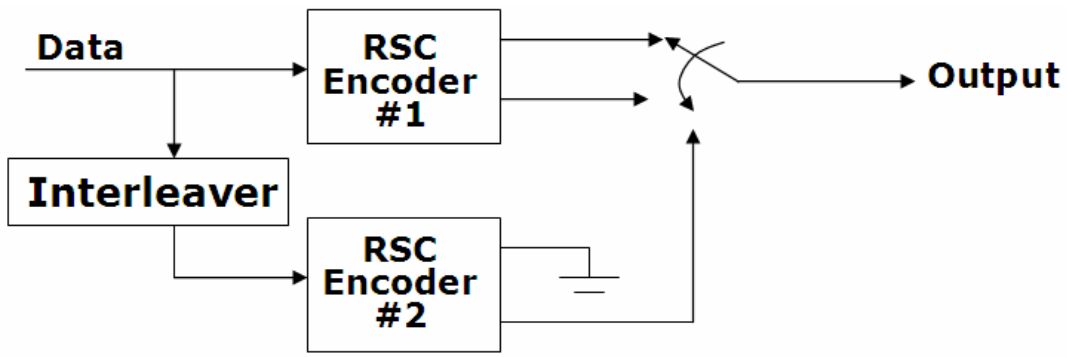


Figure 3.5: A turbo coder

A turbo decoder, illustrated in Figure 3.6 consists of two elementary Soft-Input Soft-Output (SISO) decoders that work cooperatively. Each

decoder produces a posteriori information, which is used as a priori information by the other decoder.

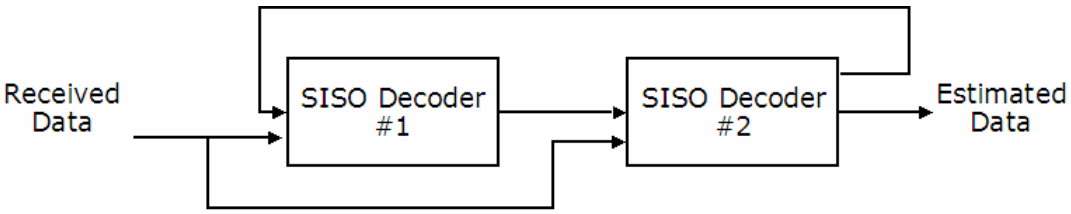


Figure 3.6: A turbo decoder

Briefly turbo coding and decoding, mentioned previously in 2.3.1.1 and 2.3.1.2 respectively, works as follows; after quantizing Wyner-Ziv frame DCT coefficients and forming bitplane array of *M* bit planes associated to the whole image quantized symbols, each bitplane array is then fed into turbo coder based Slepian Wolf Encoder starting with the most significant bitplane array. The Slepian-Wolf encoder comprises a turbo encoder and a buffer. The turbo encoder produces a sequence of parity bits for each bitplane array; the amount of parity bits produced for each bitplane depends on the turbo encoder rate, i.e. on the ratio of turbo encoder output bits per turbo encoder input bit. The parity bits produced by the turbo encoder are then stored in the buffer, punctured, according to a given puncturing pattern, and transmitted upon decoder request via the feedback channel.

The Slepian-Wolf decoding is performed using an iterative turbo decoding procedure. For each transform coefficient band, the turbo decoder starts decoding the most significant bitplane followed by the sequential decoding of the remaining bitplanes. Each transform coefficient band bitplane is decoded using the received Wyner-Ziv bits associated to that bitplane and the side information. When the received Wyner-Ziv bits together with side information are not sufficient to

provide a reliable decoding of the current bitplane, more bits are requested via the feedback channel by the decoder. After the additionally requested Wyner-Ziv bits are received, a new attempt to decode the relevant bitplane is performed. The requests and following decoding operations are executed the turbo decoding of a transform coefficient band bitplane is considered to be successful.

### 3.5.2 LDPC Encoding / Decoding

LDPC codes have been used effectively in fixed rate distributed source coding [34] and [35]. Varodayan et al [32] proposed a solution to use LDPC codes as part of a rate-adaptive scheme for distributed source coding. The LDPCA encoder, which is illustrated in Figure 3.7 consists of an LDPC syndrome-former concatenated with an accumulator.

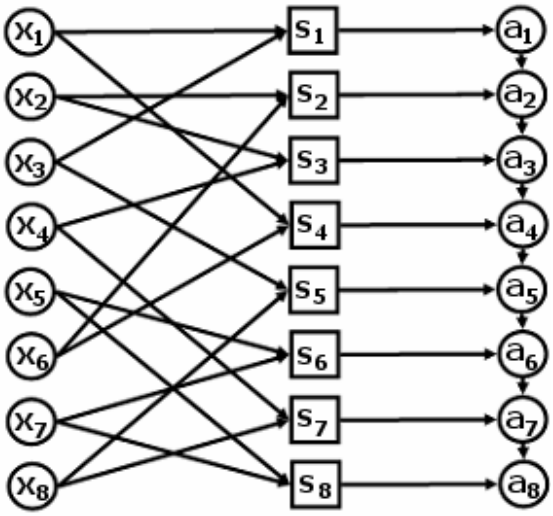


Figure 3.7: The LDPCA Encoder, based on [32]

Briefly, an LDPCA encoder works as follows; the source bits, $x_1$, $x_2$, …, $x_7$, $x_8$, are summed modulo 2 at the syndrome nodes according to the LDPC graph structure, yielding syndrome bits, $s_1$, $s_2$, …, $s_7$, $s_8$. These syndrome bits are in turn accumulated modulo 2, producing the

accumulated syndrome ($a_1$, $a_2$, …, $a_7$, $a_8$). The encoder buffers the accumulated syndrome and transmits it incrementally to the decoder. [32]

LDPCA Decoding is performed as follows, the LDPCA decoder handles rate-adaptivity by modifying its decoding graph each time it receives an additional increment of the accumulated syndrome. Assuming that all accumulated syndrome bits, $a_1$, $a_2$, …, $a_7$, $a_8$, have been received by decoder. To obtain the syndrome bits, $s_1$, $s_2$, …, $s_7$, $s_8$, the consecutive differences modulo 2 of a 's are taken. The syndrome-adjusted LDPC iterative decoding method of [34] can be applied on the same graph that was used for encoding $s_1$, $s_2$, …, $s_7$, $s_8$ from $x_1$, $x_2$, …, $x_7$, $x_8$. The decoding graph is illustrated in Figure 3.8.

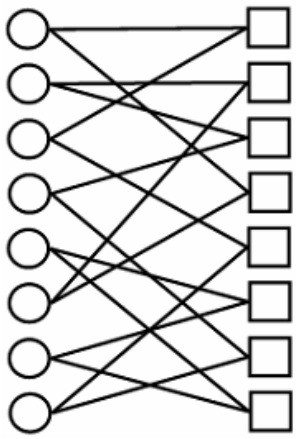

Figure 3.8: LDPC Decoding graph, assuming all accumulated syndrome bits have been received. [32]

For decoding, the source nodes are seeded with conditional probability distributions of the source bits given the side information, namely $\Pr\{X_1|Y\}$, $\Pr\{X_2|Y\}$, … , $\Pr\{X_7|Y\}$, $\Pr\{X_8|Y\}$. Then messages are passed back and forth between the source nodes and the syndrome nodes [34] until the estimates of the source bits converge. The correctness of the

recovered source values can be tested with respect to the syndrome bits. The modification of decoding graph structure manifests at higher compression ratios. For instance a compression ratio of two can be achieved by transmitting only the even-indexed subset of the accumulated syndrome, $a_2$, $a_4$, $a_6$ and $a_8$. The consecutive difference modulo 2 operation at the decoder then produces $s_1 + s_2$, $s_3 + s_4$, $s_5 + s_6$ and $s_7 + s_8$. In Figure 3.9, the graph which may encode $s_1 + s_2$, $s_3 + s_4$, $s_5 + s_6$ and $s_7 + s_8$ from $x_1$, $x_2$, …, $x_7$, $x_8$ is given.

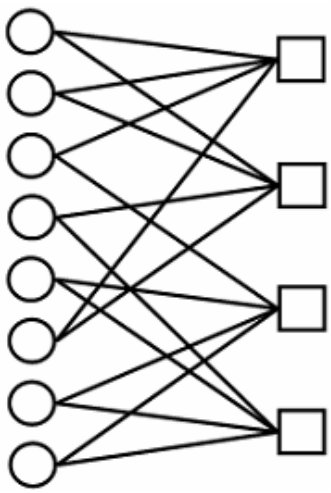

Figure 3.9: LDPC Decoding graph, assuming all even indexed syndrome bits have been received. [32]

The degree of all source nodes is maintained by Figure 3.9. Therefore, it can be used for effective iterative decoding with source bit seeding $\Pr\{X_1|Y\}$, $\Pr\{X_2|Y\}$, … , $\Pr\{X_7|Y\}$, $\Pr\{X_8|Y\}$. Upon completion of decoding, the recovered source can be tested against the syndrome to verify correctness. Varodayan et al states that the encoding and decoding complexity of these LDPCA codes is linear in the number of edges, which is invariant under the proposed construction and the

complexity of encoding and decoding is *O(n)*, where *n* is the blocklength of the code in bits. [32]

The usage of LDPCA Codes in the implementation is as follows;

In the encoding phase, for each bit plane, syndrome bits are created using the LDPC code and accumulated modulo 2 to produce the accumulated syndrome as mentioned above. The Slepian-Wolf Encoder stores these accumulated syndromes in a buffer and initially transmits only a few, in chunks. In addition to that, to aid the decoder detecting residual errors, an 8-bit CRC (Cyclic Redundancy Check) sum [37] of the encoded bit plane is also transmitted by the encoder. If the Wyner-Ziv decoder fails, it requests extra accumulated syndromes from the encoder using the feedback channel. In the decoding phase, the channel decoder, i.e. Slepian-Wolf decoder, corrects the bit errors in the side information using a belief propagation procedure on the initial number of accumulated syndromes received from the encoder buffer. A syndrome error check mechanism is used in order to establish if decoding is successful. The convergence is tested by computing the Hamming distance between the received syndrome and the one generated using the decoded bit plane, followed by a CRC [38]. If the computed Hamming distance is not zero, then the decoder proceeds to the next iteration. After a certain amount of iterations, if the Hamming distance is not zero, then the bit plane is assumed to be decoded erroneously and the LDPCA decoder requests extra accumulated syndromes using the feedback channel. If the calculated Hamming distance is zero, then the decoding operation is verified using the 8-bit CRC sum [38]. If the CRC sum of the decoded bit plane matches the CRC received from the encoder, the decoding is declared successful and the decoded bit plane is sent to the reconstruction module. Otherwise the decoder requests extra accumulated syndromes and thus a final low error probability is always guaranteed. [11]

## 3.6 Side Information Generation

The Wyner-Ziv decoder requires the side information to be available to reconstruct *X*. Considering Figure 3.1, basic architecture of the implemented codec, it is assumed that the key frames, i.e. odd frames of the sequence are available at the decoder. Several techniques that can be employed at the Wyner-Ziv decoder to generate the side information, *Y*, have been proposed in the literature. The simplest frame interpolation technique, for instance, that can be used are to make *Y* equal to the previous temporally adjacent frame, which means assuming that there is no temporal variation between previous frame and the current Wyner-Ziv frame to be reconstructed. Another technique can be performing bilinear (average) interpolation between the previous and the next key frames. However, using these techniques to generate the side information in video sequences where the similarity between two temporally adjacent frames is low, *Y* will be a rough estimate of *X*. In this case, the decoder needs to request extra accumulated syndrome bits from the encoder to decode the encoded bitplane when compared to the case where *Y* is a closer estimate of the *X* frame. Thus the Wyner-Ziv bitrate will increase for the same PSNR; the Wyner-Ziv bitrate refers to the number of Wyner-Ziv bits (parity bits) needed to decode encoded bitplane. Subjectively, these simple frame interpolation techniques will introduce in the decoded frame *X'* artifacts such as "jerkiness" and "ghosting", especially for low bitrates. Thus, sophisticated techniques based on the motion estimation of the video sequence are essential to generate high quality side information, which is a good estimate of the original Wyner-Ziv frame when the similarity between adjacent frames is low. With more complex techniques than the simple copy of the previous frame, it is possible to obtain a side information frame more similar to the original Wyner-Ziv frame and thus minimize the Wyner-Ziv bitrate for the same decoded Wyner-Ziv frame quality. The accuracy of side information generation significantly influences the rate-distortion performance of the Wyner-

Ziv codec. The motion estimation techniques, used in traditional video coding at the encoder, attempt to choose the best prediction for the current frame in the rate-distortion sense; in other words, for a given block in the current frame, the motion estimation techniques attempt to find the best match in the reference frame independently of the true motion of the block in the scene. For frame interpolation, these techniques are not well-suited since in this case the current frame is not known and it is necessary to estimate the true motion to correctly interpolate the original Wyner-Ziv Frame, usually by motion compensation between temporally adjacent frames. In Figure 3.10, the frame interpolation technique used in the implementation is illustrated.

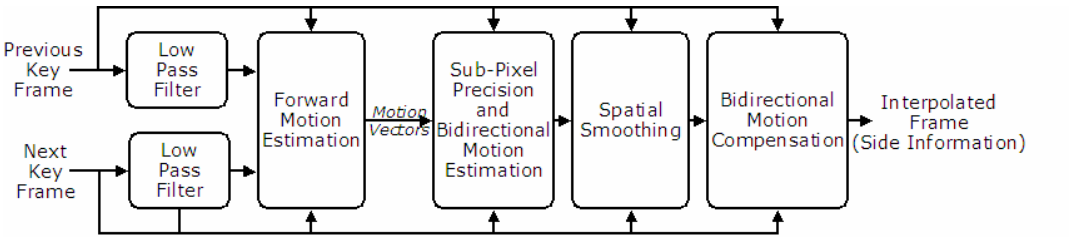


Figure 3.10: Side Information Generation based on Motion Compensated Frame Interpolation

This scheme has been proposed by Ascenso *et al* [39] for pixel domain DVC, however, within the context of this thesis, this is the implemented side information generation method. In addition to low pass filter and the motion compensation modules which are always used, there are three modules, namely Forward Motion Estimation module, Sub-Pixel Precision and Bidirectional Motion Estimation module and Spatial Smoothing module, for more accurate frame interpolation, i.e. side information generation. In a nutshell, this scheme proposes a technique based on motion compensated frame interpolation, that is, for a given Wyner-Ziv frame, motion estimation is performed between

temporally adjacent previous and next frames (key frames). Motion vectors obtained by the motion estimation are then used with both key frames for bidirectional motion compensation. Thus, side information is obtained.

#### 3.6.1.1 Low-Pass Filtering and Forward Motion Estimation

First step to obtain the interpolated frame, i.e. side information, is low-pass filtering. For a given Wyner-Ziv frame, for which the interpolated frame is to be generated, both temporally adjacent previous and next key frames are low-pass filtered in order to improve the reliability and spatial correlation, i.e. smoothness, of the motion vectors between them. Then, to estimate the motion between the next previous temporally adjacent key frames a low complexity block matching algorithm based on sum of absolute differences (SAD) is employed. This motion estimation algorithm is characterized by the window size, search range and step size parameters. The window size is the dimension of the square block used as basic unit to perform motion estimation and is fixed for the entire frame. The search range parameter defines the previous key frame area dimension in which the block most similar to the current block in the next key frame is searched for. The step size is the distance between pixels in the previous key frame a motion vector is searched for; this parameter enables i) to reduce the computational complexity of the motion estimation scheme by increasing the step size and ii) to provide only a coarse approximation of the true motion field. This block-based forward motion estimation algorithm does not capture all features of the motion field like occluded areas. If frame interpolation is performed, overlapped and uncovered areas will appear in the interpolated frame. This is due to the obtained motion vectors which do not necessarily intercept the interpolated frame at the center of each non-overlapping block in the interpolated frame. This problem is illustrated in Figure 3.11 where pixels between two neighboring blocks in the interpolated frame are not interpolated.

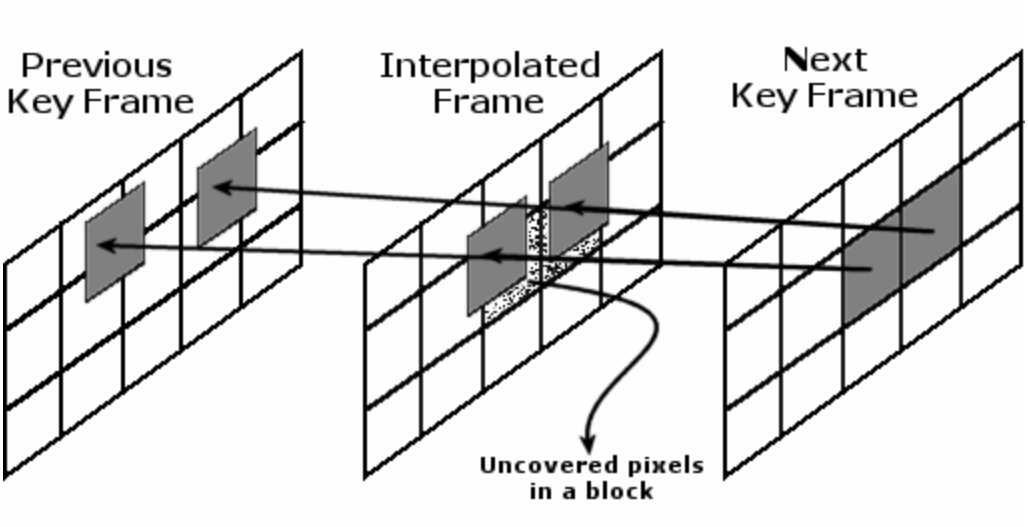


Figure 3.11: Unknown pixel values between neighbouring blocks

One possible technique to solve the problem illustrated in Figure 3.11 can be as follows; the motion vectors which are obtained by the forward motion estimation stage serve as candidates for each non-overlapping block in the interpolated frame *Y*. Figure 3.12 illustrates the motion vector selection scheme for a given block of the interpolated frame *Y*. For each block in *Y*, from all candidate motion vectors obtained by the forward motion estimation, the motion vector that intercepts the interpolated frame *Y* closer to the center of block under consideration is selected.

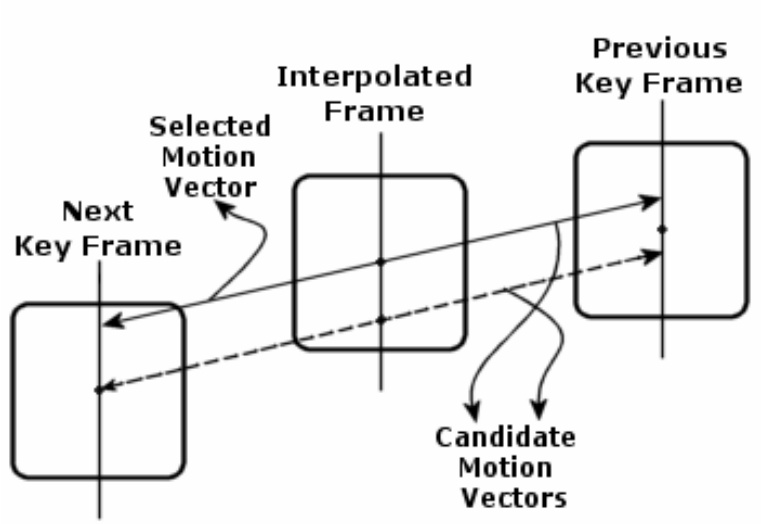


Figure 3.12: Motion vector selection among candidates obtained by forward motion estimation for a given block

Thus each block in the interpolated frame *Y* has a motion vector associated to it.

#### 3.6.1.2 Sub-pixel Precision and Bidirectional Motion Estimation

After assigning a motion vector for every block in the interpolated frame *Y*, bidirectional motion estimation is performed with sub-pixel precision. In order to estimate the sub-pixel motion linear or bicubic interpolation i.e. upsampling is performed on the previous and next key frames. A bidirectional motion estimation algorithm similar to the B-frames coding mode used in current video standards is employed to refine the motion vectors obtained in the forward motion estimation procedure. However, the pixel values in the interpolated frame *Y* are not known unlike the B-frames coding mode. The adopted bidirectional motion estimation scheme is illustrated in Figure 3.13.

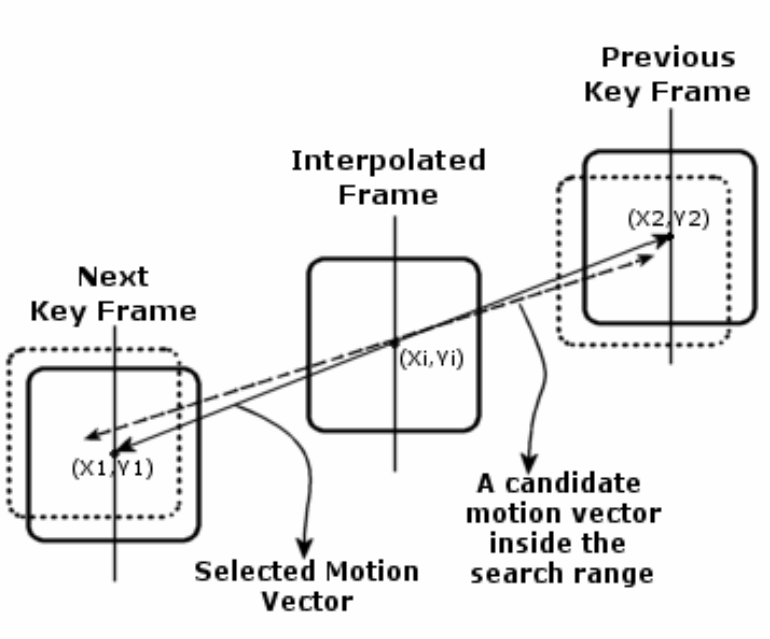


Figure 3.13: Selection of linear trajectory among candidate motion vectors

This scheme selects a linear trajectory between then next and previous key frames passing at the center of the blocks in the interpolated frame *Y*. The search range is limited to a small displacement around the initial block position and the motion vectors between the interpolated frame and previous and next key frames are symmetric, that is

$$(X1, Y1) = (Xi, Yi) + MotionVectorToNextFrame \quad (3.7)$$

$$(X2, Y2) = (Xi, Yi) - MotionVectorToNextFrame \quad (3.8)$$

Where $(X1,Y1)$ are the coordinates of the block in next key frame and $(X2,Y2)$ are the coordinates of the block in the previous key frame. Similarly, $(Xi, Yi)$ are the coordinates of the block in the interpolated frame. $MotionVectorToNextFrame$ is the half of the motion vector obtained in the forward motion estimation stage. The motion vector obtained in

the forward motion estimation stage is divided by two since the interpolated frame *Y*, is equally distant to both previous and next key frames. Considering (3.7) and (3.8), bidirectional motion estimation is performed between next and previous key frames. Motion vectors obtained by the forward motion estimation stage are refined with small symmetrical displacements on the previous and next key frames on condition that the selected linear trajectory passes through the center of a given block in the interpolated frame. In bidirectional motion estimation stage, a hierarchical approach from coarse to fine is applied. That is, in the first iteration by using 16x16 block size, motion vectors corresponding to fast movements of the frame pixels are obtained. In the second iteration finer motion vectors have been obtained by using 8x8 blocks for motion search range. The refined motion vectors represent more accurate motion trajectories and some of the errors introduced by the initial motion vectors are corrected.

#### 3.6.1.3 Spatial Smoothing

Some of the motion vectors obtained by bidirectional motion estimation may have low spatial coherence. In order to achieve higher motion field coherence, spatial smoothing techniques can be employed. The implemented spatial smoothing algorithm uses median filters [40] which are commonly applied for noise removal in multichannel images. This median filter maintains the motion field spatial coherence, for each block, by searching for candidate motion vectors from neighboring blocks.

#### 3.6.1.4 Bidirectional Motion Compensation

The interpolated frame, *Y*, can now be filled using the motion vectors obtained by spatial smoothing. By the assumption that the time interval between the previous key frame and interpolated frame is equal to the time interval between the next key frame and interpolated frame, each key frame has the weight, $\frac{1}{2}$, when motion compensation

is performed. That is, by taking the average of each block in both key frames the blocks in the interpolated frame can be filled. Thus, interpolated frame, i.e. side information is obtained.

## 3.7 Reconstruction and Inverse Transformation

After decoding the $M_k$ bitplanes associated to the DCT band, $b_k$, the bitplanes are grouped together to form the decoded quantized symbol stream associated to the $b_k$ band; this procedure is performed over all the DCT coefficients bands to which Wyner-Ziv bits are transmitted. Once the entire decoded quantized symbol streams are obtained, it is possible to reconstruct the matrix of DCT coefficients, $X'_{DCT}$. As was mentioned in Section 3.4, the DCT coefficients bands to which no Wyner-Ziv bits are sent are replaced by the corresponding DCT bands of the side information, $Y_{DCT}$. The remaining DCT bands are obtained through LDPCA decoding procedures, as was described in Section 3.5.2. Since it is assumed that the decoder knows the maximum absolute value within each AC band, which is sent by the encoder, and the number of quantization levels for each AC band, the quantization step size for the AC coefficients bands can be easily computed at the decoder, as described in Section 3.4. The highest value within the DC coefficients band is $2^8 = 256$ and is assumed to be fixed (see Section 3.4). The quantization step size computed at the decoder is therefore equal to the one at the encoder side, for each DCT coefficients band. After the quantization step size is calculated, it is possible to establish the boundaries of the quantization intervals, for a given DCT coefficients band $b_k$ and reconstruction can be performed. There are three cases for reconstruction procedure for each DCT coefficient band $b_k$,

i) The side information DCT coefficient $Y_{DCT}$ can be within the LDPC decoded quantized symbol interval $[I_{q'}^{\min}, I_{q'}^{\max}]$. The reconstructed DCT

coefficient $X'_{DCT}$ is made equal to the side information DCT coefficient. (Illustrated in Figure 3.14 )

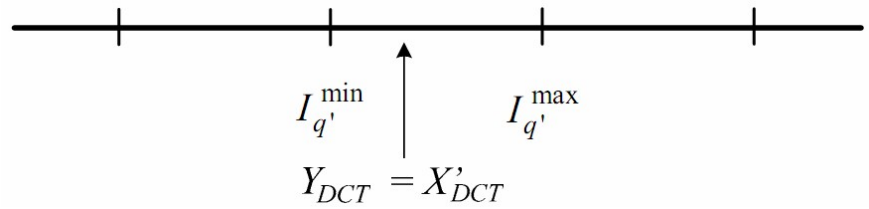


Figure 3.14: Side information is within the decoded quantized symbol interval

ii) The side information DCT coefficient $Y_{DCT}$ can belong to a quantized symbol lower in magnitude than the LDPC decoded symbol. The reconstructed DCT coefficient $X'_{DCT}$ assumes the lowest intensity value within the decoded quantized symbol, $I_{q'}^{\min}$, which is the lower bound of the quantization interval indexed by the decoded quantized symbol. (Illustrated in Figure 3.15)

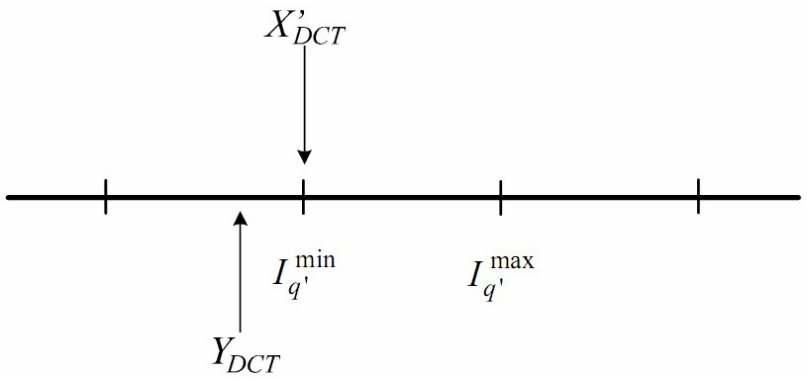


Figure 3.15: Side information belongs to a quantized symbol lower in magnitude than the LDPC decoded symbol

iii) The side information DCT coefficient $Y_{DCT}$ can belong to a quantized symbol higher in magnitude than the LDPC decoded symbol. The reconstructed DCT coefficient $X'_{DCT}$ assumes the highest intensity value within the decoded quantized symbol, $I_{q'}^{\max}$, which is the upper bound of the quantization interval indexed by the decoded quantized symbol. (Illustrated in Figure 3.16)

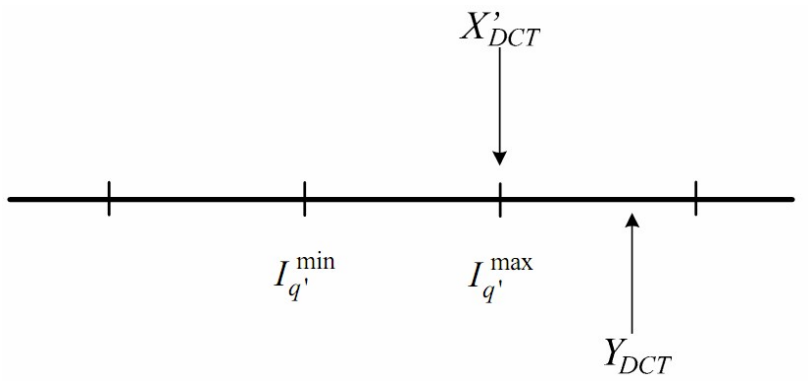


Figure 3.16: Side information belongs to a quantized symbol higher in magnitude than the LDPC decoded symbol

Reconstructed DCT coefficient is always between the boundaries of the decoded quantized symbol, thus, the error between DCT coefficients of $X_{DCT}$ and $X'_{DCT}$ is limited to the quantizer step size $W_b$. (See Figure 3.4 )
In order to reconstruct the Wyner-Ziv frames, the inverse discrete cosine transform (IDCT) must be applied over the reconstructed matrix of DCT coefficients $X'_{DCT}$. At the decoder, the side information, $Y$, is obtained with motion compensated frame interpolation using previous and next temporally adjacent frames. To obtain the DCT coefficients of the side information, $Y_{DCT}$, a 4x4 DCT is applied to the side information $Y$. The LDPCA decoder uses $Y_{DCT}$ to obtain the decoded quantized symbol stream associated to the DCT band $b_k$. $Y_{DCT}$ is also necessary in the reconstruction module, together with the quantized symbol stream

to help in the DCT coefficients matrix reconstruction task, $X'_{DCT}$. Since encoded DCT coefficients bands are transmitted to the decoder, the inverse of the DCT operation, known as inverse discrete cosine transform – IDCT, has to be performed at some stage of the Wyner-Ziv decoding procedure in order to obtain the reconstructed $X$ frame, $X'$. The IDCT operation is carried out over $X'_{DCT}$, i.e. the reconstructed matrix of DCT coefficients. The IDCT transform is performed in a similar way to the DCT transform operation.

# CHAPTER 4

# EXPERIMENTAL RESULTS

In order to evaluate the rate-distortion performance of the implemented side information generation technique and DVC codec, four well-known test sequences, namely Coastguard, Foreman, Hall Monitor and Soccer have been used. Table 4.1 provides a brief description of the main characteristics of each test sequence.

Table 4.1: Main characteristics of the video test sequences

| Test Sequence | TestSequence1 | TestSequence2 | TestSequence3 | TestSequence4 |
|---|---|---|---|---|
| First Frame |  |  |  |  |
| Total number of frames | 150 | 150 | 165 | 150 |
| Number of evaluated frames | 101 | 101 | 101 | 101 |
| Spatial Resolution | QCIF (176x144) | QCIF (176x144) | QCIF (176x144) | QCIF (176x144) |
| Temporal Resolution | 15 fps | 15 fps | 15 fps | 15 fps |
| GOP Size | 2 | 2 | 2 | 2 |
| Activity Type | Well defined, regular and low activity | Medium Activity | Low Activity | High amount of activity |

As given in Table 4.1, in all sequences GOP Size is 2, temporal resolution is 15 fps and spatial resolution is QCIF (176x144). In all

simulations first 101 frames of the sequence have been evaluated. The Coastguard sequence is characterized by a well defined motion of the objects present in the scene. From now on Coastguard sequence will be referred as *TestSequence1*. The Foreman sequence, which will be referred as *TestSequence2* from now on, is characterized by medium activity due to the movement of the head and gestures of the speaker. The Hall Monitor sequence, will be referred as *TestSequence3*, is a surveillance type video, which contains small moving objects and complex background, therefore the amount of motion is low throughout the sequence. The Soccer sequence, which will be referred as *TestSequence4*, is an example of sports content characterized by higher amount of movement. Some test sequences listed in Table 4.1 are representatives of video conference content, typically characterized by low and medium activity – e.g. amount of movement – e.g. the *TestSequence1*, *TestSequence2* and *TestSequence3*, whereas the *TestSequence4* is an example of high activity involving sequence. The higher the activity, the more difficult is to code the video content. This content variety is important to collect enough representative and meaningful results for the implemented codec performance. Only the luminance data is considered in the implemented codec rate-distortion performance evaluation. The H.264/AVC Intra encoded previous and next key frames and The Wyner-Ziv bitstream is assumed to be error-free received, i.e. no errors are introduced during the transmission. For a given Wyner-Ziv frame the dynamic range of each DCT band is assumed to be losslessly available at the decoder. The 8 quantization matrices depicted in (3.5) and (3.6) are used to obtain 8 different RD points for the implemented codec. The side information is generated using motion compensated frame interpolation algorithms at the decoder. Besides forward and bidirectional motion estimation, a spatial motion smoothing algorithm is used to eliminate motion outliers allowing significant improvements in the RD performance. A Laplacian distribution models the residual between the Wyner-Ziv frame DCT

coefficients and the corresponding DCT coefficients of the Y frame, i.e. side information. Since the Wyner-Ziv codec performance is to be evaluated, the rate-distortion plots only contain the rate and the PSNR values for the even frames, i.e. the Wyner-Ziv coded frames, of a given video sequence.

## 4.1 Effects of Side Information on the Quality

As mentioned in Section 3.6, the quality of side information, i.e. correlation between Wyner-Ziv and side information frames, directly influences the overall performance of distributed video codec. In this section, gradual improvement in the side information quality by the implemented tools will be illustrated in detail by comparing motion compensated frame interpolation technique with a simpler side information generation method, i.e. frame averaging. Figure 4.1 illustrates the change in PSNR for the side information frame passing through the stages in motion compensated interpolation scheme for *TestSequence1*. PSNR for the side information obtained by averaging previous and next key frames (*SI by Frame Averaging*) is also included in the plot. Table 4.2 shows the PSNR values used to evaluate the results of SI generation stages.

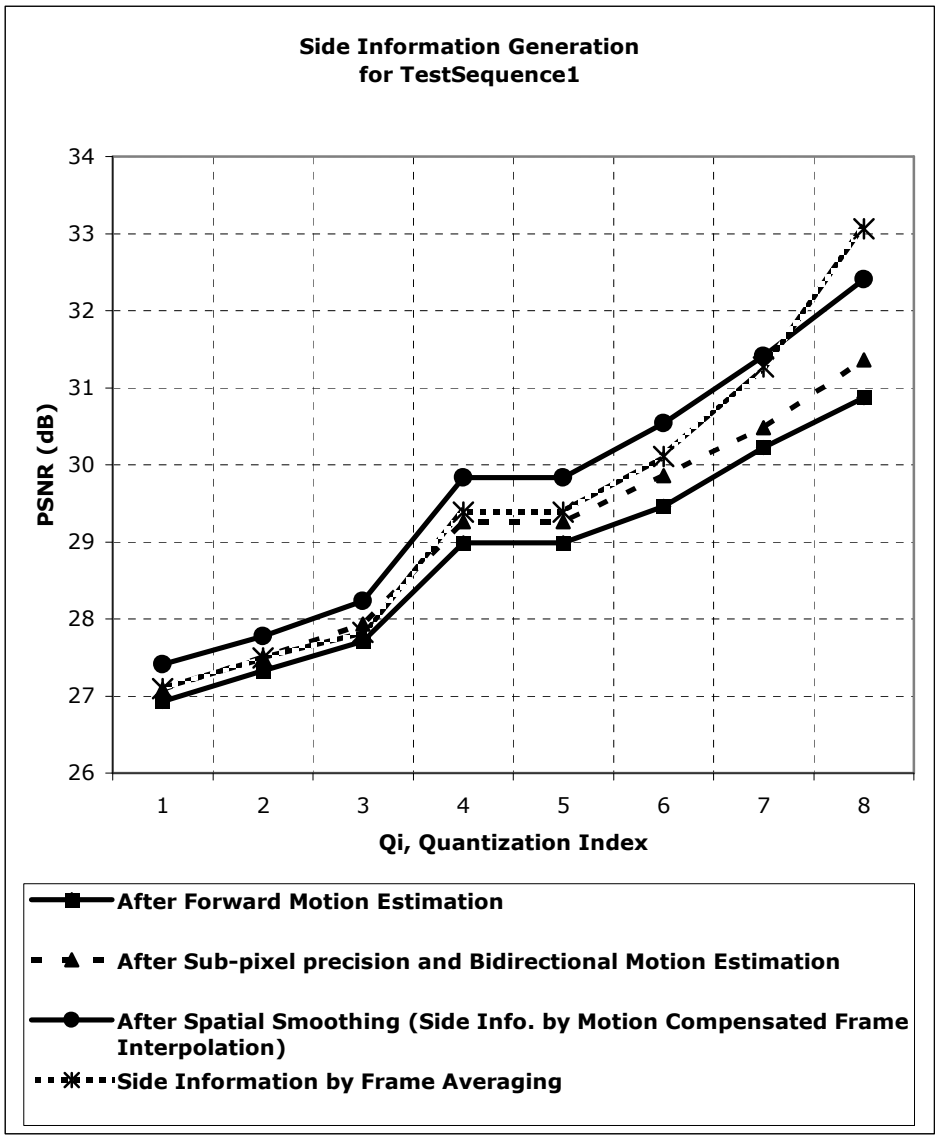


Figure 4.1: Quality Improvement of side information frame for *TestSequence1*

Table 4.2: PSNR values of *TestSequence1* SI frames for several quantization indices

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| After Forward Motion Estimation | 26,93 | 27,33 | 27,71 | 28,99 | 28,99 | 29,46 | 30,22 | 30,87 |
| After Sub-pixel precision and Bidirectional Motion Estimation | 27,09 | 27,50 | 27,93 | 29,26 | 29,26 | 29,86 | 30,48 | 31,36 |
| After Spatial Smoothing (SI From Motion Compensated Frame Interpolation) | 27,41 | 27,78 | 28,23 | 29,83 | 29,83 | 30,54 | 31,41 | 32,41 |
| SI from Frame Averaging | 27,10 | 27,50 | 27,83 | 29,38 | 29,38 | 30,11 | 31,27 | 33,06 |

As observed, the gradual increase in side information quality, introduced by the adopted tools, is higher (up to 1.6 dB) for higher bitrates, i.e. for larger $Q_i$'s. Also, it is noticed that for higher bitrates frame averaging, i.e. average frame interpolation outperforms motion compensated interpolated side information frame. Figure 4.2 illustrates the change in PSNR for the side information frame passing through the stages in motion compensated interpolation scheme for *TestSequence2*. PSNR for the side information obtained by averaging previous and next key frames (SI by Frame Averaging) is also included in the plot. Table 3.1 shows the PSNR values used to evaluate the results of SI generation stages.

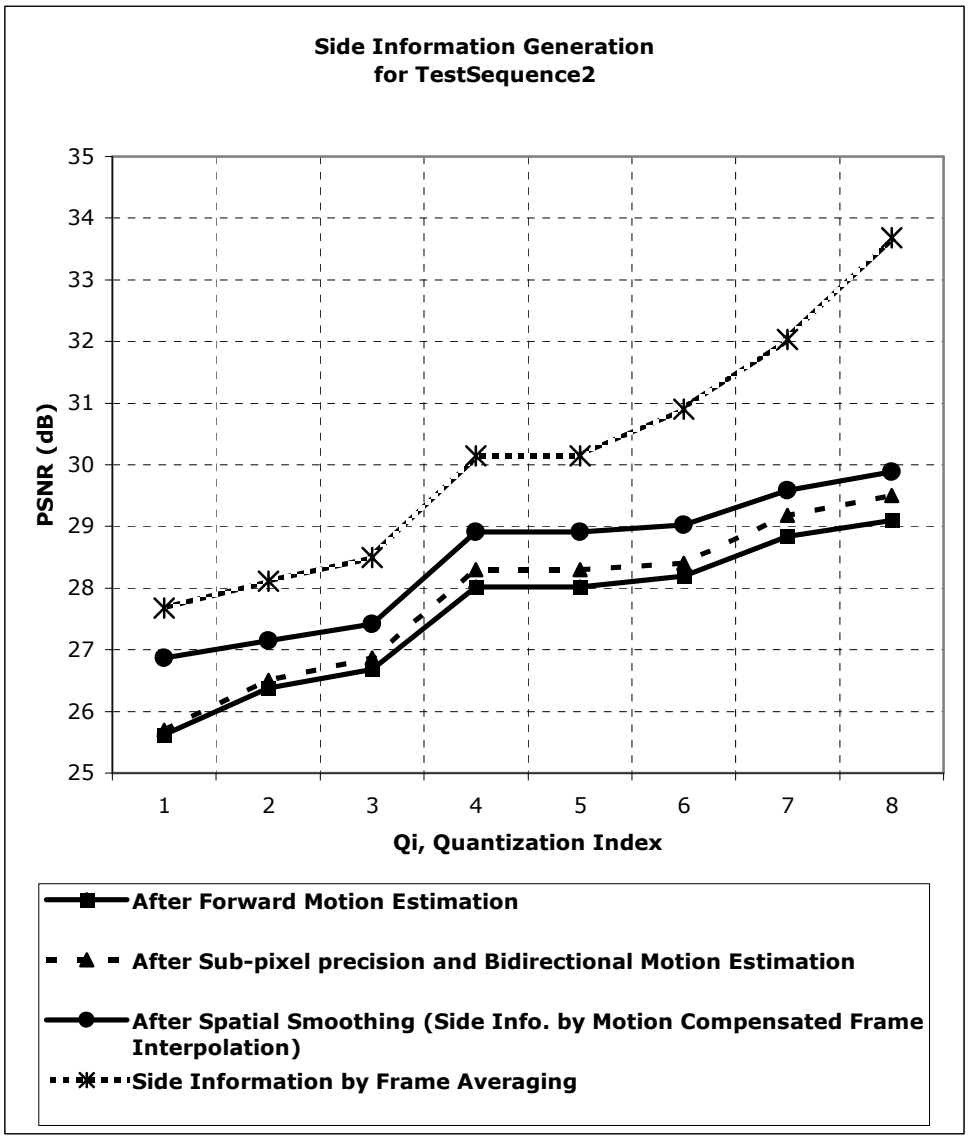


Figure 4.2: Quality Improvement of side information frame for *TestSequence2*

Table 4.3: PSNR values of *TestSequence2* SI frames for several quantization indices

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| After Forward Motion Estimation | 25,62 | 26,38 | 26,68 | 28,02 | 28,02 | 28,20 | 28,84 | 29,10 |
| After Sub-pixel precision and Bidirectional Motion Estimation | 25,69 | 26,50 | 26,87 | 28,30 | 28,30 | 28,40 | 29,18 | 29,50 |
| After Spatial Smoothing (SI From Motion Compensated Frame Interpolation) | 26,87 | 27,15 | 27,42 | 28,91 | 28,91 | 29,03 | 29,58 | 29,89 |
| SI from Frame Averaging | 27,68 | 28,11 | 28,50 | 30,15 | 30,15 | 30,90 | 32,03 | 33,68 |

In this scheme, the improvement introduced by additional tools is up to 1.2 dB for lower bitrates and drops slightly as the bitrate increases. PSNR difference between motion compensated frame interpolated side information and frame averaged side information is higher (up to 3.8 dB) for high bitrates. However, as illustrated in Figure 4.3, and Table 4.4, high PSNRs for side information frames do not always guarantee to have a better rate-distortion performance in the distributed video codec. This result will be mentioned in detail in Section 5.1. Figure 4.3 illustrates the rate-distortion performance of implemented codec with two types of side information generation techniques, i.e. motion compensated frame interpolation and frame averaging. Table 4.4 has the values used for the plot.

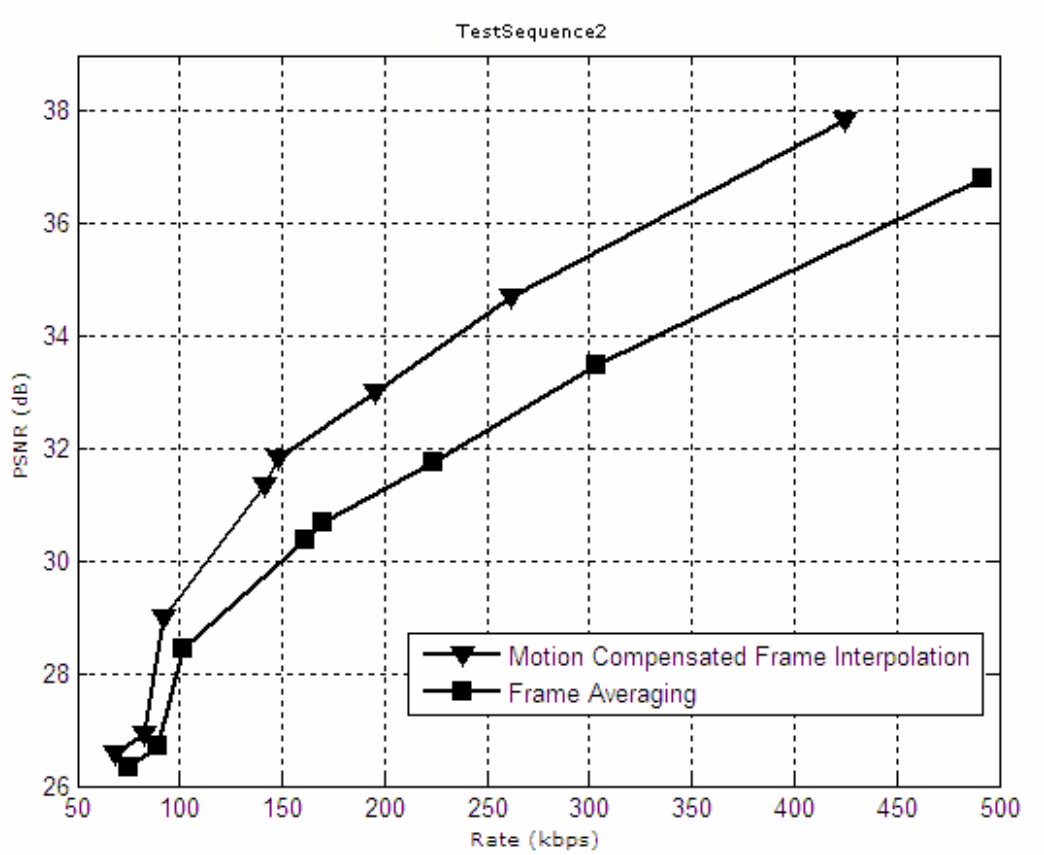


Figure 4.3: Rate-distortion performance of implemented codec with different side informations for *TestSequence2*

Table 4.4: DVC codec performance with different side informations for *TestSequence2*

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Implemented Codec with SI from Motion Compensated Frame Interpolation | | | | | | | | |
| PSNR | 26,58 | 26,9 | 28,99 | 31,35 | 31,83 | 33 | 34,7 | 37,85 |
| Rate (kbps) | 68,837 | 82,602 | 92,138 | 141,160 | 147,932 | 195,121 | 261,286 | 424,660 |
| Implemented Codec with SI from Frame Averaging | | | | | | | | |
| PSNR | 26,33 | 26,7 | 28,45 | 30,39 | 30,67 | 31,75 | 33,5 | 36,82 |
| Rate (kbps) | 74,886 | 89,401 | 101,702 | 161,427 | 170,177 | 224,097 | 303,032 | 491,644 |

As observed for *TestSequence2*, SI generated by frame averaging has higher PSNR value for a given quantization index. However, using it with the implemented Wyner-Ziv Encoder / Decoder structure yielded up to 2 dB performance decrease for the same bitrate, or, an increase in bitrate up to 100 kbps for the same PSNR. Figure 4.4 illustrates the change in PSNR for the side information frame passing through the stages in motion compensated interpolation scheme for *TestSequence3*. PSNR for the side information obtained by averaging previous and next key frames (SI by Frame Averaging) is also included in the plot. Table 4.5 shows the PSNR values used to evaluate the results of SI generation stages.

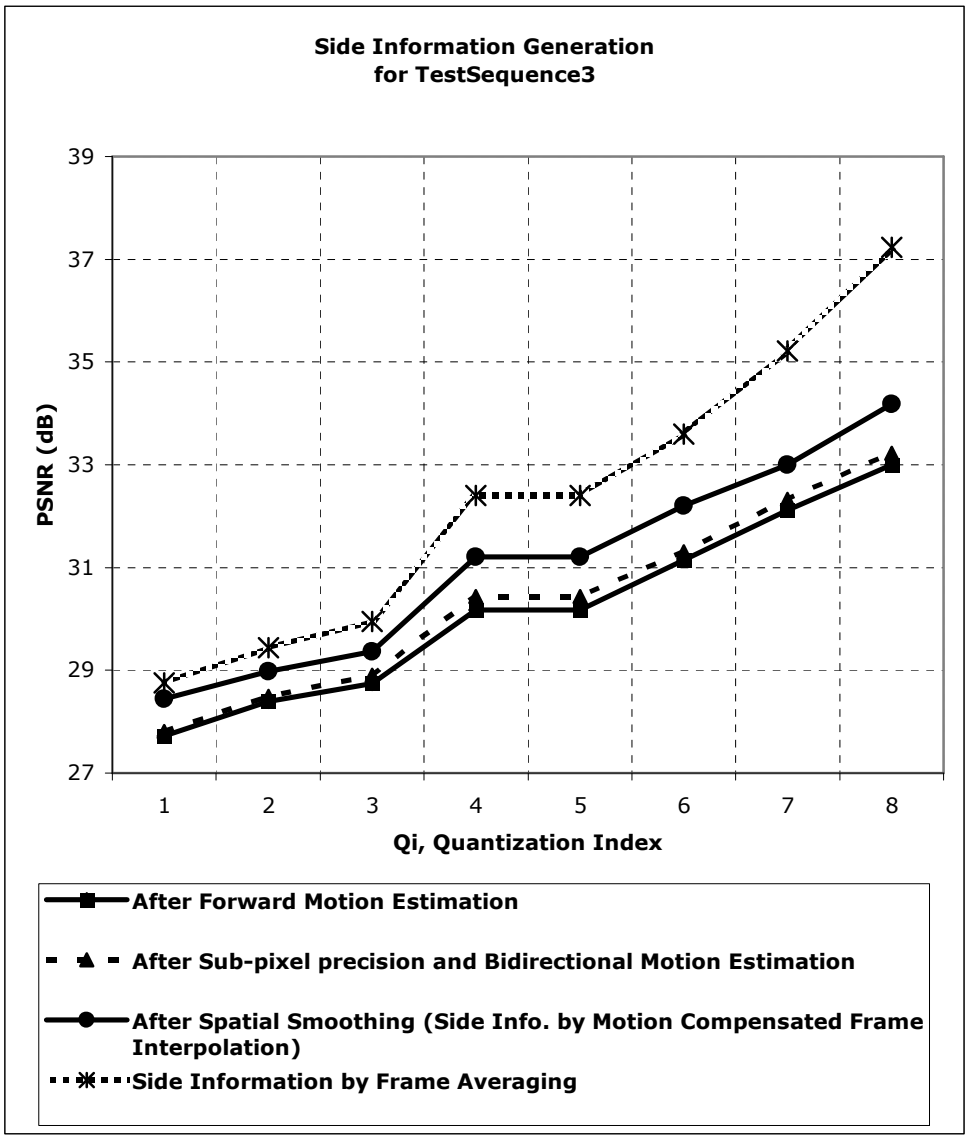


Figure 4.4: Quality Improvement of side information frame for *TestSequence3*

Table 4.5: PSNR values of *TestSequence3* SI frames for several quantization indices

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| After Forward Motion Estimation | 27,72 | 28,39 | 27,74 | 30,18 | 30,18 | 31,15 | 32,12 | 33,00 |
| After Sub-pixel precision and Bidirectional Motion Estimation | 27,81 | 28,50 | 28,91 | 30,43 | 30,43 | 31,31 | 32,33 | 33,22 |
| After Spatial Smoothing (SI From Motion Compensated Frame Interpolation) | 28,44 | 28,98 | 29,36 | 31,21 | 31,21 | 32,20 | 33,00 | 34,18 |
| SI from Frame Averaging | 28,75 | 29,44 | 29,95 | 32,40 | 32,40 | 33,60 | 35,21 | 37,23 |

For lower bitrates, the improvement by adopted tools is around 0.7 dB, however, the PSNR improvement increases as the bitrate increases. Also, frame averaging outperforms motion compensated technique (about 3.1 dB) for higher bitrates. Figure 4.5 illustrates the change in PSNR for the side information frame passing through the stages in motion compensated interpolation scheme for *TestSequence4.* PSNR for the side information obtained by averaging previous and next key frames (SI by Frame Averaging) is also included in the plot. Table 4.6 shows the PSNR values used to evaluate the results of SI generation stages.

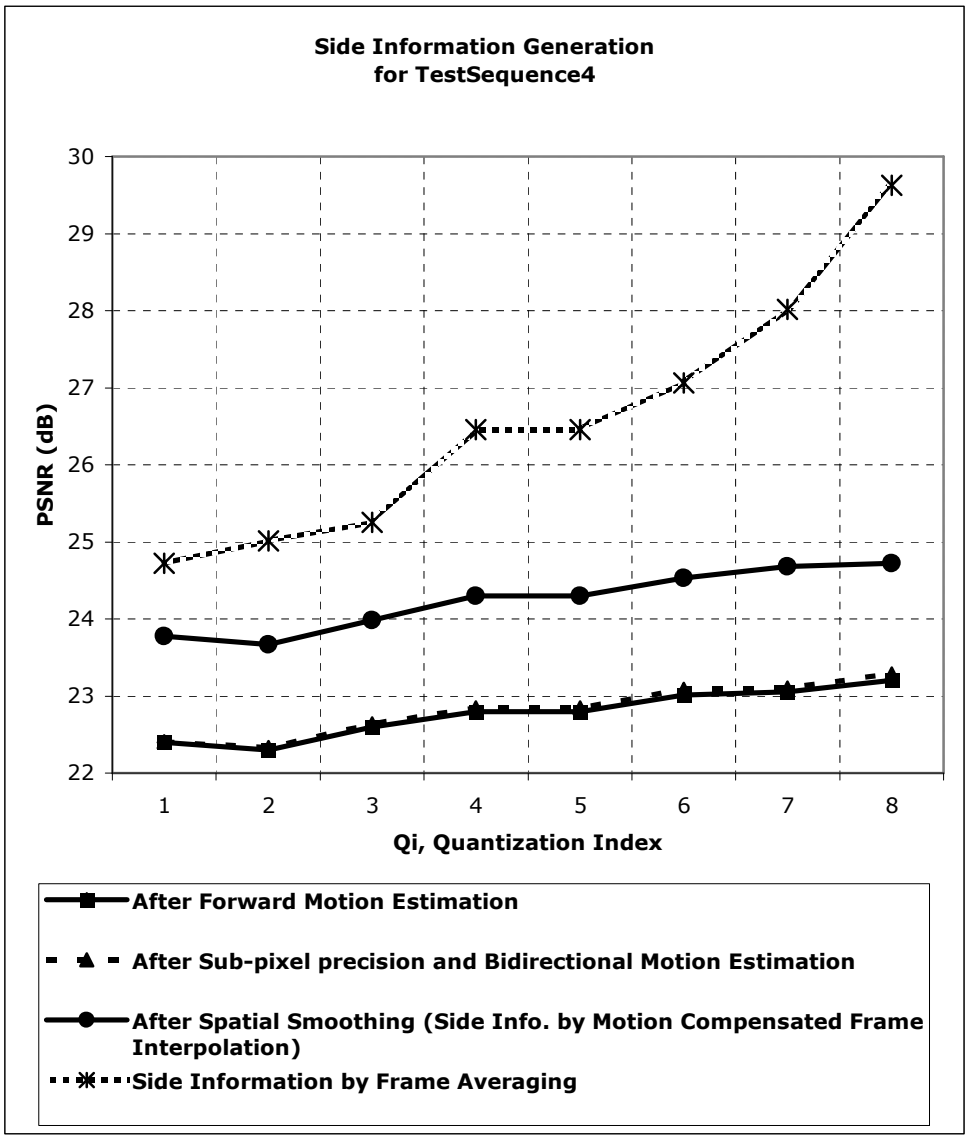


Figure 4.5: Quality Improvement of side information frame for *TestSequence4*

Table 4.6: PSNR values of *TestSequence4* SI frames for several quantization indices

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| After Forward Motion Estimation | 22,40 | 22,30 | 22,60 | 22,80 | 22,80 | 23,01 | 23,05 | 23,20 |
| After Sub-pixel precision and Bidirectional Motion Estimation | 22,41 | 22,33 | 22,64 | 22,85 | 22,85 | 23,09 | 23,10 | 23,29 |
| After Spatial Smoothing (SI From Motion Compensated Frame Interpolation) | 23,78 | 23,67 | 23,98 | 24,30 | 24,30 | 24,53 | 24,68 | 24,72 |
| SI from Frame Averaging | 24,72 | 25,01 | 25,25 | 26,46 | 26,46 | 27,06 | 28,02 | 29,63 |

*TestSequence4* is the most motion involving sequence, thus, the PSNR values for all frames are dramatically low when compared to other sequences. As observed, the highest value for PSNR difference between frame averaging and motion compensated is observed in *TestSequence4* (frame averaging outperformed up to 5 dB). This is because; the implemented motion compensating techniques are inadequate for this much of motion throughout temporally adjacent frames.

## 4.2 Distributed Video Codec Performance Evaluation

In this section, the rate-distortion performance of the implemented codec will be compared to that of some well known traditional video coding algorithms. For each test sequence, the rate-distortion performance of implemented codec is compared to H.263+ Intraframe coding (H.263+ Intra in plots), H.264/AVC Intraframe coding in Main Profile (H.264/AVC Intra in plots), H.264/AVC Interframe coding which exploits temporal redundancy in a IB…IB… structure without any motion estimation (H.264/AVC Inter No Motion in plots) and H.264/AVC Interframe coding in Baseline profile (H.264/AVC Inter in plots). In the literature, proposed DVC schemes were mostly compared to video

coding with H.263+ without exploiting temporal redundancy. It is not the best standard Intra coding available with worse rate-distortion performance but less computationally complex structure than H.264/AVC Intra, which is the most efficient Intra coding method. The most computationally expensive encoding method among the compared ones is full motion compensated Interframe coding with H.264/AVC in Baseline profile. It exploits temporal redundancy in an IP……PI structure. Considering H.264/AVC Inter No Motion mode, it achieves better performance than Intra coding because it can partly exploit temporal redundancy by using a DPCM temporal scheme, but requires far less complexity than full motion compensated Inter coding because no motion search is performed. The rate vs. PSNR values of H.263+ Intra, H.264/AVC Intra, and H.264/AVC Inter No Motion for the same test sequences with same GOP size used in the implemented codec can be found in Discover Codec Evaluation Web Site [31]. In the following subsections, the results obtained with the implemented codec for each one of the four QCIF video test sequences, listed in Table 4.1, will be presented and analyzed. In Figure 4.6, the rate-distortion performance with *TestSequence1* of the implemented distributed video codec is illustrated. A more quantitative comparison can be made using Table 4.7.

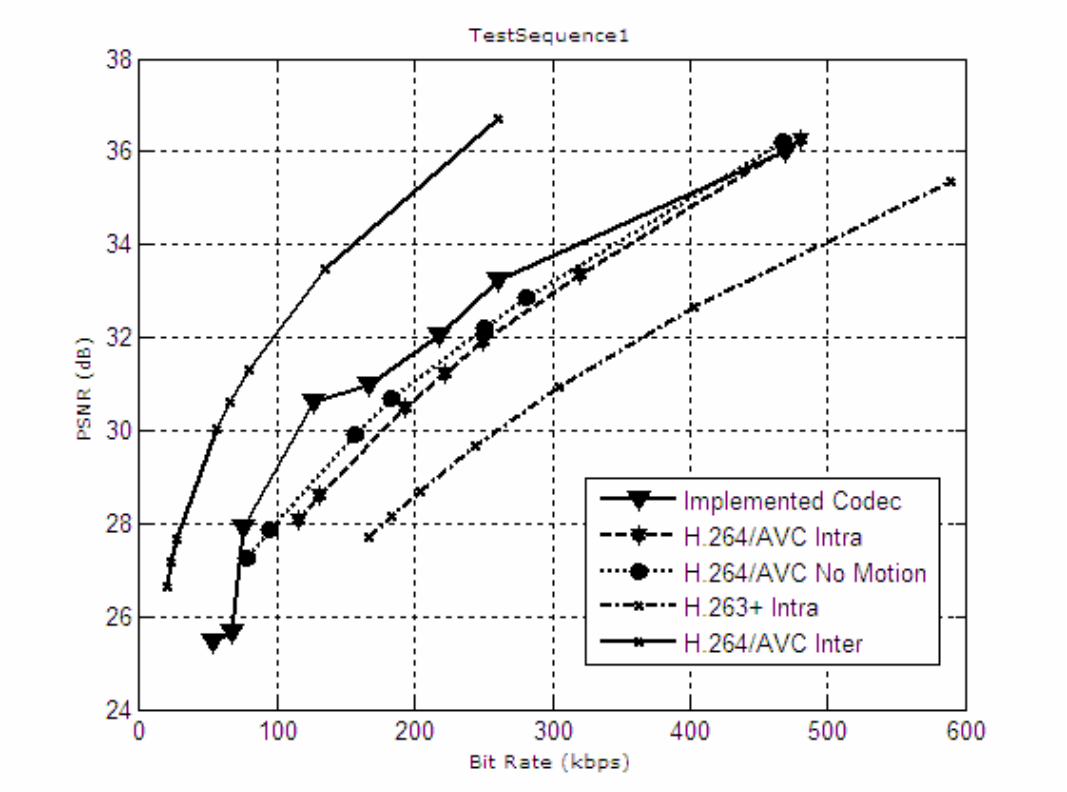


Figure 4.6: RD performance of implemented codec with *TestSequence1*

Table 4.7: PSNR and Rate values for *TestSequence1*

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 25,46 | 25,66 | 27,9 | 30,61 | 30,96 | 32,06 | 33,22 | 36 |
| Rate (kbps) | 53,642 | 67,293 | 75,437 | 126,458 | 167,234 | 217,561 | 261,161 | 469,357 |
| **H.264/AVC (Intra)** | | | | | | | | |
| PSNR | 28,09 | 28,62 | 28,62 | 30,5 | 31,23 | 31,92 | 33,35 | 36,27 |
| Rate (kbps) | 115,14 | 130,38 | 130,38 | 192,54 | 222,23 | 250,18 | 320,04 | 479,9 |
| **H.264/AVC (No motion)** | | | | | | | | |
| PSNR | 27,24 | 27,88 | 27,88 | 29,91 | 30,69 | 32,21 | 32,88 | 36,2 |
| Rate (kbps) | 78,53 | 95,22 | 95,22 | 156,46 | 183,62 | 250,4 | 281,33 | 467,42 |
| **H.263+ (Intra)** | | | | | | | | |
| PSNR | 27,71 | 28,17 | 28,7 | 29,66 | 30,94 | 32,64 | 35,36 | |
| Rate (kbps) | 167,28 | 183,43 | 203,27 | 243,98 | 304,5 | 403,11 | 589,42 | |
| **H.264/AVC (Inter)** | | | | | | | | |
| PSNR | 26,66 | 27,18 | 27,66 | 30,05 | 30,6 | 31,31 | 33,46 | 36,7 |
| Rate (kbps) | 19,73 | 23,27 | 27 | 55,82 | 65,82 | 79,345 | 135,44 | 260,56 |

As illustrated, for low and moderate bitrates, the implemented codec outperforms even H.264/AVC No Motion scheme by a PSNR difference

up to 2 dB around 140 kbps or by 50 kbps bitrate for the same PSNR level around 30 dB. This is due to the well-defined motion characteristics of the sequence, i.e. well-behaved motion throughout the whole sequence which is better estimated by the motion compensation based frame interpolation method used for side information generation. However, the performance of the implemented codec is still far less than H.264/AVC Inter. In Figure 4.7, the rate-distortion performance with *TestSequence2* of the implemented distributed video codec is illustrated. The data for the results can be found in Table 4.8.

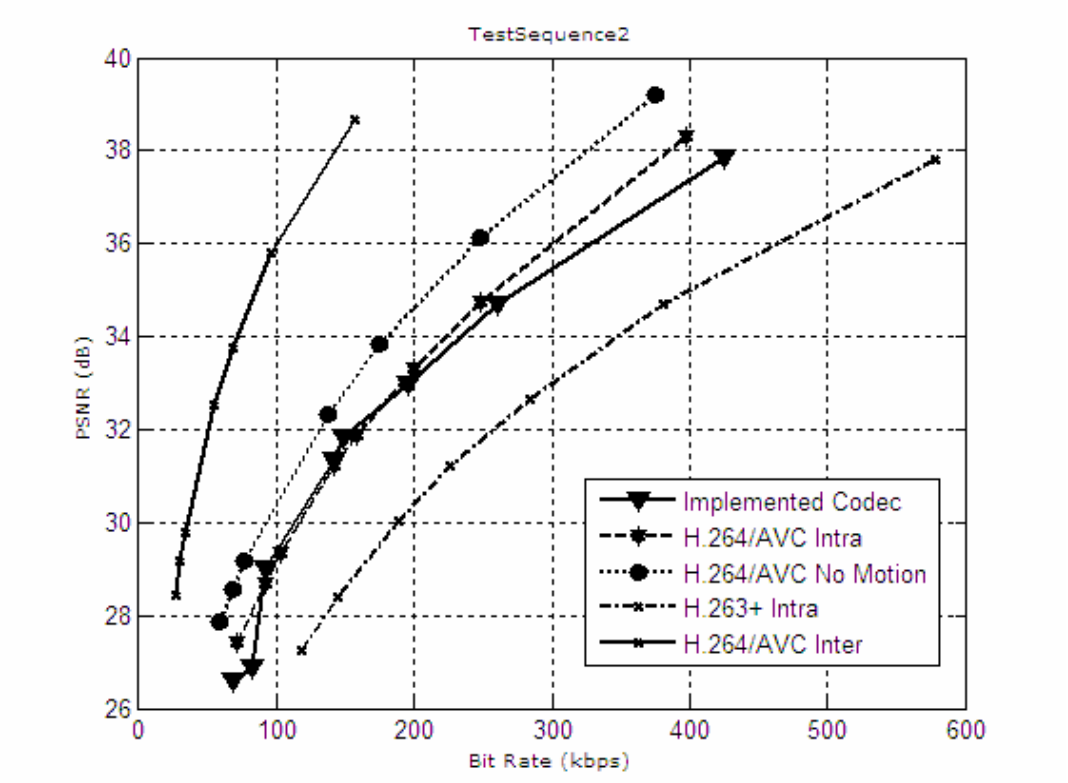


Figure 4.7: RD performance of implemented codec with *TestSequence2*

Table 4.8: PSNR and Rate values for *TestSequence2*

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 26,58 | 26,9 | 28,99 | 31,35 | 31,83 | 33 | 34,7 | 37,85 |
| Rate (kbps) | 68,837 | 82,602 | 92,138 | 141,160 | 147,932 | 195,121 | 261,286 | 424,660 |
| **H.264/AVC (Intra)** | | | | | | | | |
| PSNR | 27,43 | 28,67 | 29,32 | 31,21 | 31,86 | 33,31 | 34,75 | 38,31 |
| Rate (kbps) | 71,64 | 91,28 | 102,4 | 142,31 | 158,64 | 199,29 | 248,82 | 397,28 |
| **H.264/AVC (No motion)** | | | | | | | | |
| PSNR | 27,85 | 28,57 | 29,19 | 32,32 | 32,32 | 33,85 | 36,14 | 39,2 |
| Rate (kbps) | 58,67 | 68,07 | 77,16 | 137,03 | 137,03 | 175,02 | 247,72 | 375,05 |
| **H.263+ (Intra)** | | | | | | | | |
| PSNR | 27,25 | 28,41 | 30,05 | 31,22 | 32,66 | 34,71 | 37,79 | |
| Rate (kbps) | 118,37 | 145,03 | 188,81 | 226,6 | 283,78 | 380,53 | 578,16 | |
| **H.264/AVC (Inter)** | | | | | | | | |
| PSNR | 28,44 | 29,19 | 29,77 | 32,53 | 32,53 | 33,77 | 35,79 | 38,65 |
| Rate (kbps) | 26,37 | 30,26 | 33,46 | 54,32 | 54,32 | 67,89 | 95,825 | 157,25 |

As illustrated, the implemented codec is outperformed by H.264/AVC No Motion for all bitrates, by up to 2 dB for bitrates around 350 kbps. This is due to non-well-behaved motion characteristics of *TestSequence2* for which the implemented motion estimation scheme is not the best one. The performance of implemented codec is very comparable to that of H.264/AVC Intra. In Figure 4.8, the rate-distortion performance with *TestSequence3* of the implemented distributed video codec is illustrated. A more quantitative comparison can be made using Table 4.9.

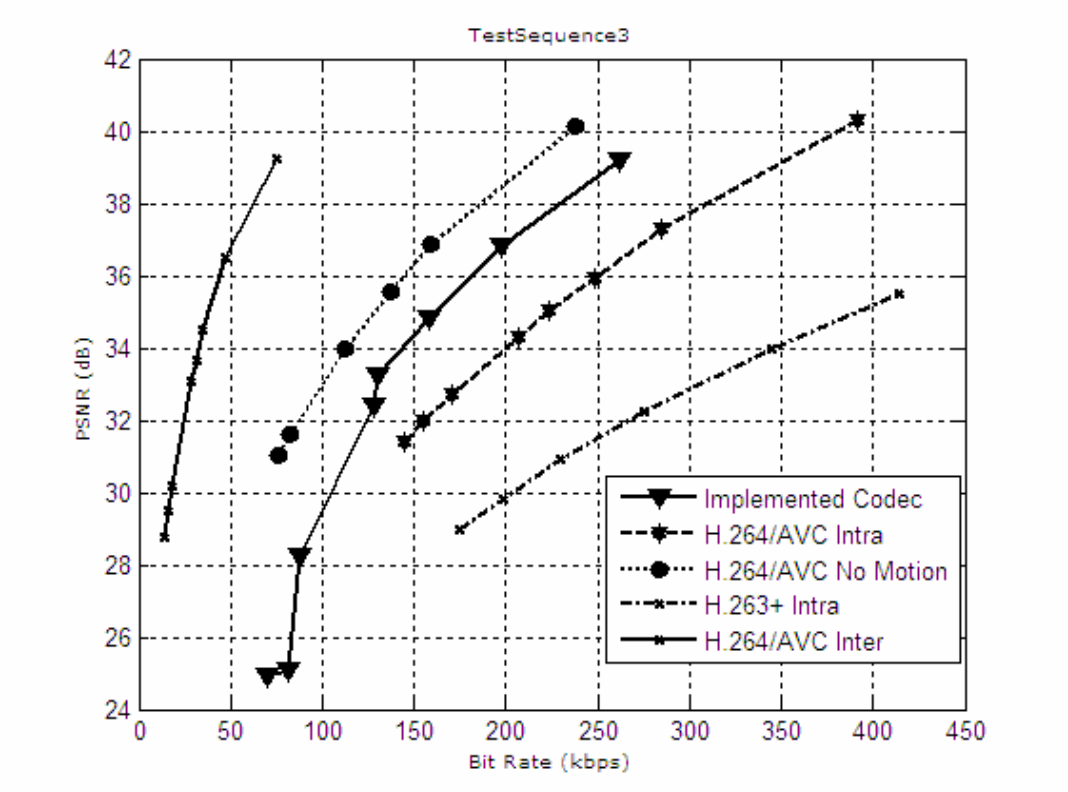


Figure 4.8: RD performance of implemented codec with *TestSequence3*

Table 4.9: PSNR and Rate values for *TestSequence3*

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 24,93 | 25,09 | 28,24 | 32,42 | 33,23 | 34,81 | 36,84 | 39,18 |
| Rate (kbps) | 70,453 | 80,930 | 88,082 | 127,991 | 130,322 | 157,709 | 197,513 | 261,480 |
| **H.264/AVC (Intra)** | | | | | | | | |
| PSNR | 31,4 | 31,96 | 32,69 | 34,3 | 35,05 | 35,92 | 37,27 | 40,28 |
| Rate (kbps) | 144,35 | 154,83 | 170,91 | 206,52 | 223,74 | 248,2 | 285 | 391,55 |
| **H.264/AVC (No motion)** | | | | | | | | |
| PSNR | 31,04 | 31,61 | 31,61 | 33,98 | 33,98 | 35,57 | 36,89 | 40,15 |
| Rate (kbps) | 76,68 | 82,89 | 82,89 | 112,8 | 112,8 | 136,95 | 158,83 | 238,02 |
| **H.263+ (Intra)** | | | | | | | | |
| PSNR | 28,95 | 29,84 | 30,91 | 32,24 | 33,97 | 35,52 | | |
| Rate (kbps) | 174,29 | 198,39 | 229,83 | 274,94 | 344,7 | 414,51 | | |
| **H.264/AVC (Inter)** | | | | | | | | |
| PSNR | 28,74 | 29,51 | 30,19 | 33,06 | 33,66 | 34,48 | 36,52 | 39,25 |
| Rate (kbps) | 14,37 | 16,15 | 17,91 | 28,05 | 31,54 | 34,26 | 46,7 | 75,17 |

Outperforming H.263+ Intra and H.264/Avc Intra, the performance of implemented codec is 2 dB off H.264/AVC No Motion for bitrates around

150 kbps. The difference in bitrate is nearly 40 kbps for PSNR values around 37 dB. In Figure 4.9, the rate-distortion performance with *TestSequence4* of the implemented distributed video codec is illustrated. The data for the results can be found in Table 4.10.

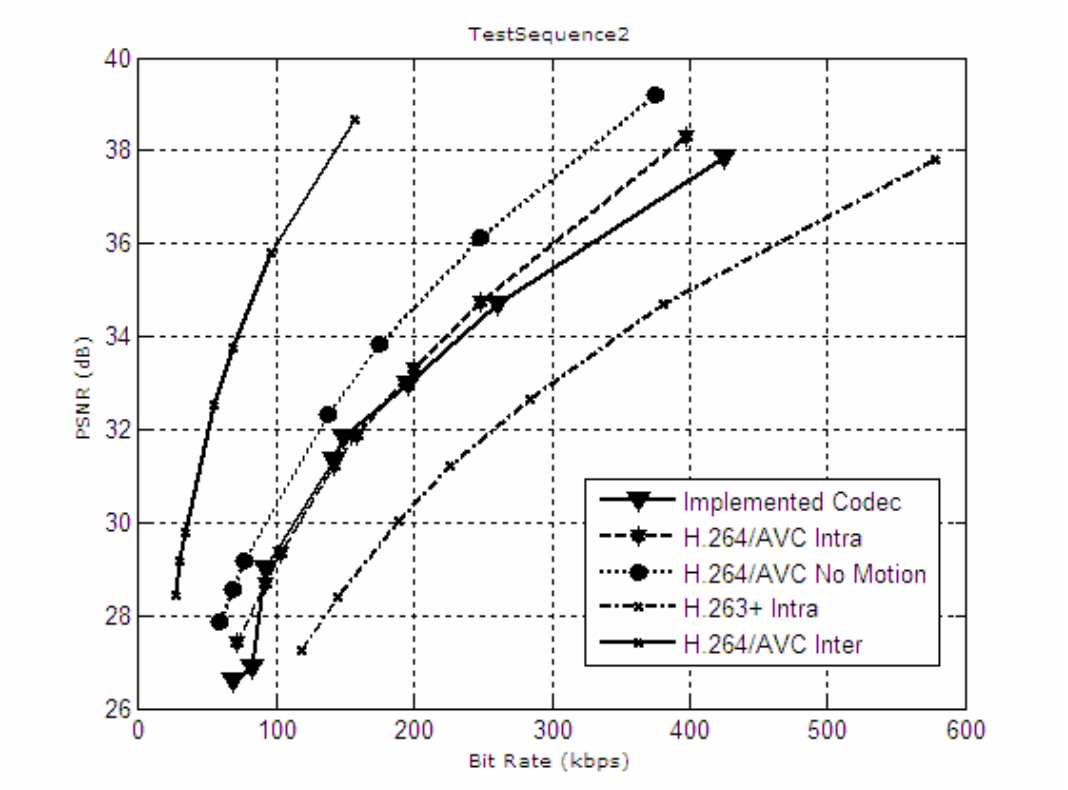


Figure 4.9: RD performance of implemented codec with *TestSequence4*

Table 4.10: PSNR and Rate values for *TestSequence4*

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 26,58 | 26,9 | 28,99 | 31,35 | 31,83 | 33 | 34,7 | 37,85 |
| Rate (kbps) | 68,837 | 82,602 | 92,138 | 141,160 | 147,932 | 195,121 | 261,286 | 424,660 |
| **H.264/AVC (Intra)** | | | | | | | | |
| PSNR | 27,43 | 28,67 | 29,32 | 31,21 | 31,86 | 33,31 | 34,75 | 38,31 |
| Rate (kbps) | 71,64 | 91,28 | 102,4 | 142,31 | 158,64 | 199,29 | 248,82 | 397,28 |
| **H.264/AVC (No motion)** | | | | | | | | |
| PSNR | 27,85 | 28,57 | 29,19 | 32,32 | 32,32 | 33,85 | 36,14 | 39,2 |
| Rate (kbps) | 58,67 | 68,07 | 77,16 | 137,03 | 137,03 | 175,02 | 247,72 | 375,05 |
| **H.263+ (Intra)** | | | | | | | | |
| PSNR | 27,25 | 28,41 | 30,05 | 31,22 | 32,66 | 34,71 | 37,79 | |
| Rate (kbps) | 118,37 | 145,03 | 188,81 | 226,6 | 283,78 | 380,53 | 578,16 | |
| **H.264/AVC (Inter)** | | | | | | | | |
| PSNR | 28,44 | 29,19 | 29,77 | 32,53 | 32,53 | 33,77 | 35,79 | 38,65 |
| Rate (kbps) | 26,37 | 30,26 | 33,46 | 54,32 | 54,32 | 67,89 | 95,825 | 157,25 |

As mentioned, *TestSequence4* is an example of sports content characterized by higher amount of movement. The higher the activity, the more difficult is to code the video content, which is observed as a dramatic drop in rate-distortion values comparison. The implemented codec performed the worst by a performance difference up to 2 dB around 200 kbps even with H.263+ Intra. Performance difference gets larger as the bitrate increases. The implemented motion estimation scheme is inadequate for such amount of motion throughout a sequence.

## 4.3 DISCOVER and Implemented Codec Performance Evaluation

As mentioned, the implemented codec within the context of this thesis is based on the architecture proposed by Artigas *et al* [11]. In order to obtain comparable results, the sequences obtained from [31] have been used. Both codecs have very similar rate-distortion performance when compared to conventional video coding solutions. For all four sequences, both codecs' performance remain in the same line in comparison, that is, implemented codec outperforms conventional solutions when DISCOVER did. In this section, both codecs'

performance will be given in the same plot highlighting the similar behavior. However, the lack of detail in DISCOVER definition about some of the tools and parameters used, forced the author of this thesis to use similar approaches to some of the important architectural modules such as the frame interpolation module and reconstruction module which may explain the difference between two curves. Figure 4.10 shows the rate-distortion performance comparisons for *TestSequence1.* A more quantitative comparison can be made using Table 4.11.

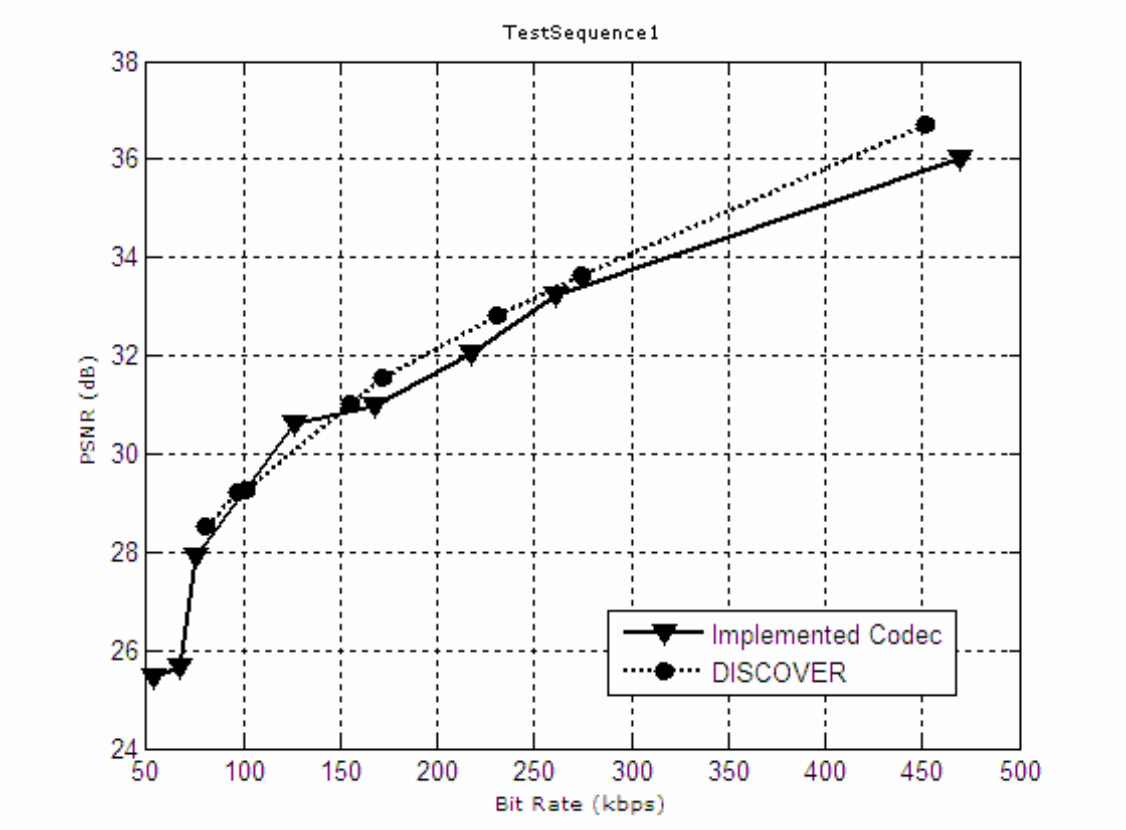


Figure 4.10: Rate-distortion performance for *TestSequence1*

Table 4.11: Rate-distortion data for detailed performance comparison

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 25,46 | 25,66 | 27,9 | 30,61 | 30,96 | 32,06 | 33,22 | 36 |
| Rate (kbps) | 53,642 | 67,293 | 75,437 | 126,458 | 167,234 | 217,561 | 261,161 | 469,357 |
| **DISCOVER** | | | | | | | | |
| PSNR | 28,51 | 29,20 | 29,25 | 31,02 | 31,55 | 32,81 | 33,65 | 36,70 |
| Rate (kbps) | 80,16 | 97,51 | 101,77 | 154,94 | 172,15 | 230,49 | 274,02 | 451,53 |

Figure 4.11 shows the rate-distortion performance comparisons for *TestSequence2.* A more quantitative comparison can be made using Table 4.12.

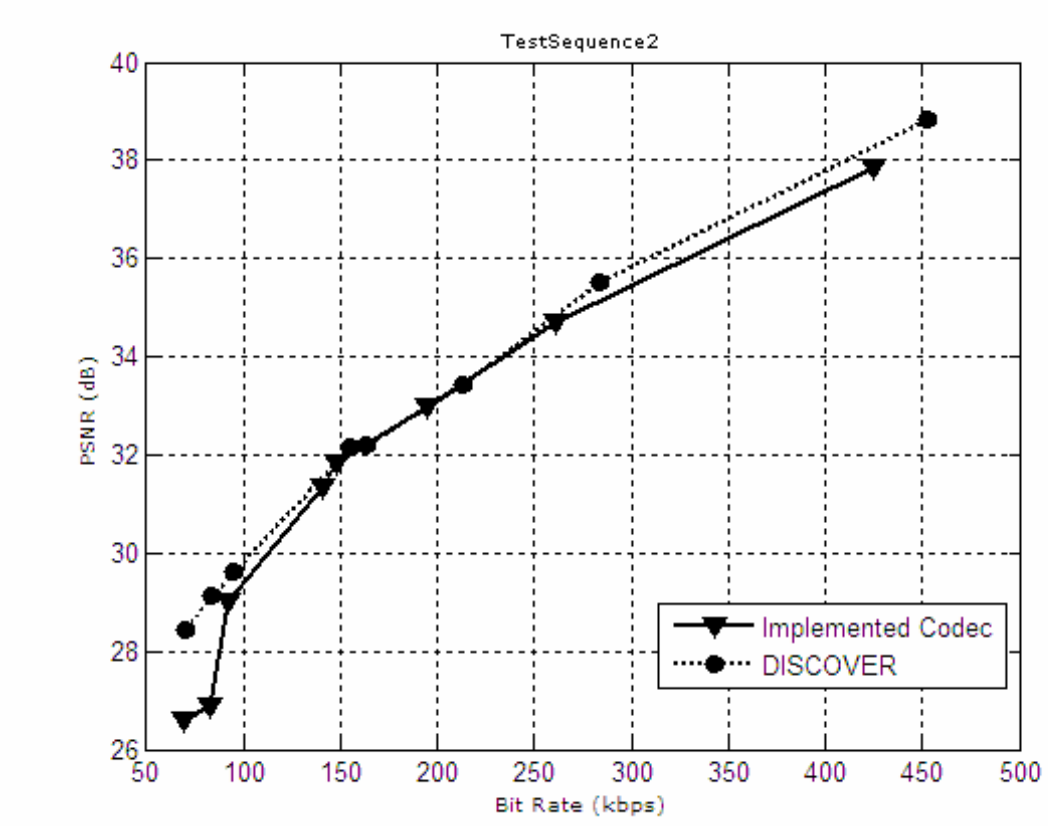


Figure 4.11: Rate-distortion performance for *TestSequence2*

Table 4.12: Rate-distortion data for detailed performance comparison

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 26,58 | 26,9 | 28,99 | 31,35 | 31,83 | 33 | 34,7 | 37,85 |
| Rate (kbps) | 68,837 | 82,602 | 92,138 | 141,160 | 147,932 | 195,121 | 261,286 | 424,660 |
| **DISCOVER** | | | | | | | | |
| PSNR | 28,44 | 29,14 | 29,64 | 32,15 | 32,2 | 33,44 | 35,52 | 38,85 |
| Rate (kbps) | 70,21 | 83,81 | 94,79 | 155,75 | 163,52 | 213,1 | 284,01 | 452,52 |

Figure 4.12 shows the rate-distortion performance comparisons for *TestSequence3.* A more quantitative comparison can be made using Table 4.13.

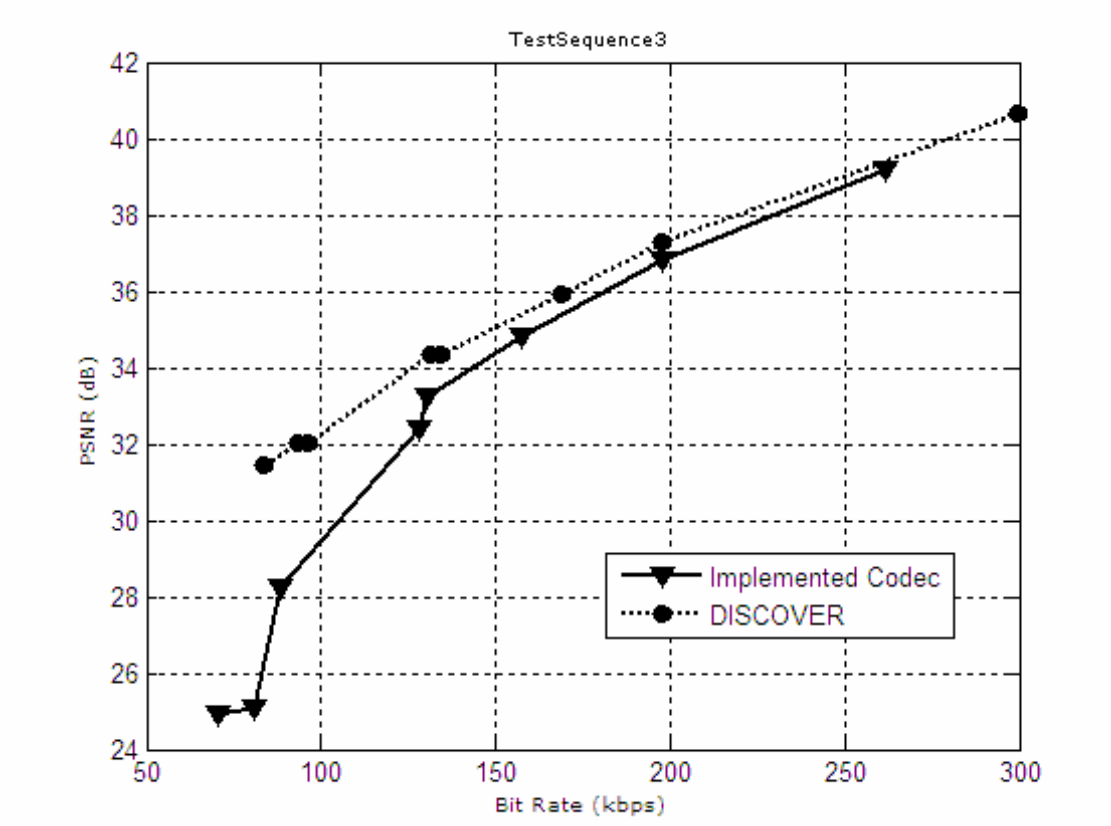


Figure 4.12: Rate-distortion performance for *TestSequence3*

Table 4.13: Rate-distortion data for detailed performance comparison

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Implemented Codec** | | | | | | | | |
| PSNR | 24,93 | 25,09 | 28,24 | 32,42 | 33,23 | 34,81 | 36,84 | 39,18 |
| Rate (kbps) | 70,453 | 80,930 | 88,082 | 127,991 | 130,322 | 157,709 | 197,513 | 261,480 |
| **DISCOVER** | | | | | | | | |
| PSNR | 31,44 | 32,01 | 32,05 | 34,33 | 34,34 | 35,91 | 37,31 | 40,65 |
| Rate (kbps) | 83,83 | 93,37 | 96,44 | 131,64 | 134,64 | 168,88 | 197,83 | 299,48 |

Figure 4.13 shows the rate-distortion performance comparisons for *TestSequence4.* A more quantitative comparison can be made using Table 4.14.

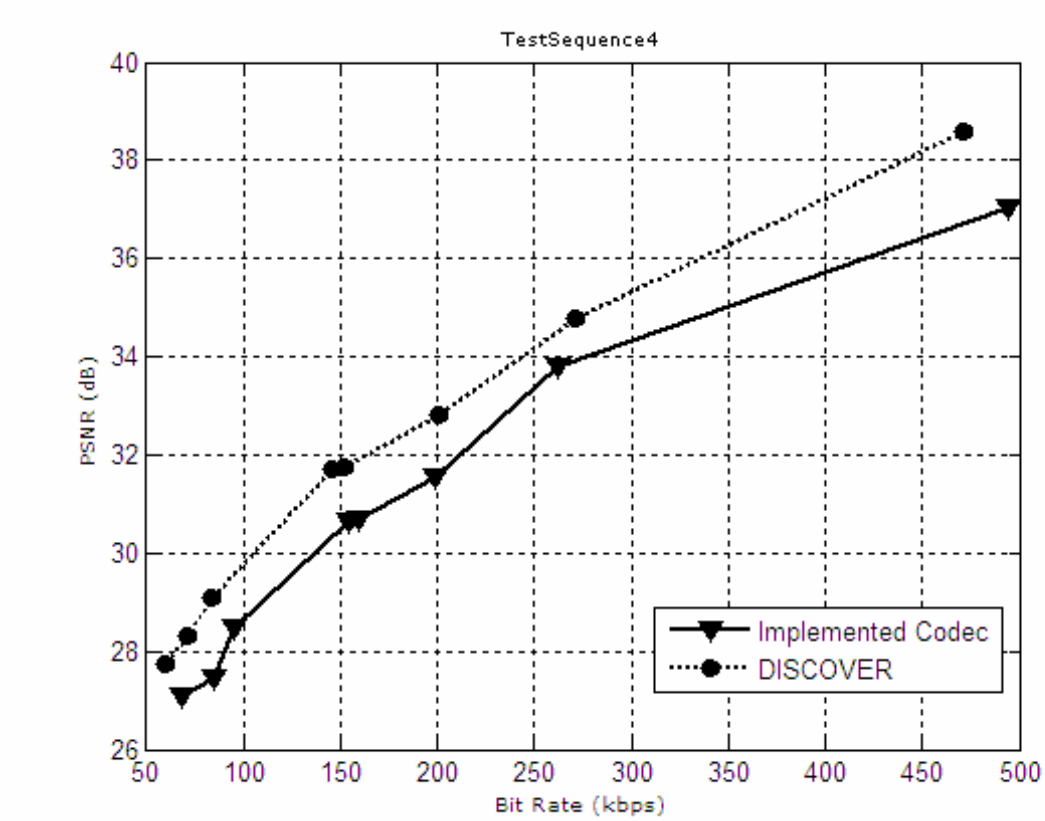


Figure 4.13: Rate-distortion performance for *TestSequence4*

Table 4.14: Rate-distortion data for detailed performance comparison

| DCT Quantization Index (Qi) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Implemented Codec | | | | | | | | |
| PSNR | 27,07 | 27,46 | 28,49 | 30,63 | 30,7 | 31,54 | 33,78 | 37,04 |
| Rate (kbps) | 68,392 | 84,243 | 94,886 | 154,110 | 159,636 | 198,435 | 262,415 | 494,333 |
| DISCOVER | | | | | | | | |
| PSNR | 27,73 | 28,33 | 29,1 | 31,7 | 31,75 | 32,82 | 34,8 | 38,58 |
| Rate (kbps) | 59,39 | 71,75 | 83,96 | 145,68 | 152,31 | 200,68 | 270,88 | 471,24 |

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

## 5.1 Conclusions

In this thesis we have implemented an end to end distributed video codec based on DISCOVER project. Using the developed architecture we show the effect of side information on the overall quality.

The quality of side information directly influences the overall performance of distributed video codec. The increase in side information quality, introduced by the adopted tools, is higher for higher bitrates. Also, it is noticed that for higher bitrates and for sequences with high motion of activity, frame averaging, i.e. average frame interpolation, outperforms motion compensated interpolated side information frame in PSNR measurement. For example, for *TestSequence4*, which is the most motion involving sequence, frame averaging outperformed motion compensated frame averaging up to *5 dB* in PSNR. This is because; the implemented motion compensating techniques are inadequate for this much of motion throughout temporally adjacent frames. Considering *TestSequence2*, SI generated by frame averaging has higher PSNR values for all quantization indices. However, using it with the implemented codec, up to *2 dB* performance decrease for the same bitrate, or, an increase in bitrate up to *100 kbps* for the same PSNR is observed. While comparing several techniques for side information generation, the technique which results in higher side information PSNR values is thought to be a proper selection. However, this thesis shows that higher PSNR values for side information do not

always guarantee higher PSNR performance for overall codec. Considering the case in Figure 4.3 and Table 4.4, SI generated by frame averaging outperforms SI generated by more complex motion compensated methods in PSNR. The decrease in PSNR for SI generated by motion compensated frame interpolation is because of the distortion in small local regions possibly with higher amount of motion activity or inadequate motion search range employed in the technique. However, SI generated by frame averaging, despite having larger PSNR values, has a global distortion which does not significantly decrease overall PSNR performance of the frame. Since the distorted regions in the motion compensated SI are small compared to the whole frame, the bits needed by LDPC decoder to correct SI is relatively less compared to the frame averaged SI case where the distortion is distributed throughout the whole SI frame. To sum up, higher PSNR values for SI frame does not yield higher PSNR performance for overall codec. What is important in SI generation is not to achieve higher PSNR values, instead finding the proper SI generation technique that results in higher overall system performance according to the motion characteristics of the sequence to be encoded. For low and moderate bitrates, the implemented codec outperforms even H.264/AVC No Motion scheme. This is due to the well-defined motion characteristics of the sequence which is better estimated by the motion compensation based frame interpolation method used for side information generation. For sequences with moderate motion activity, the performance of the implemented codec is less than H.264/AVC No Motion scheme for all bitrates but comparable to (if not greater than) H.264/AVC Intracoding scheme. For sequences with higher motion activity the performance of implemented codec is comparable to H.263+ Intra. As mentioned, sequences of this type require more complex motion compensating techniques.

For all four test sequences, Implemented Distributed Video Codec and DISCOVER performed very similar in terms of rate-distortion. Our

implemented codec, despite having PSNR performance up to 0.8 dB less than DISCOVER, shows better compression efficiency for a given quantization index $Q_i$ with its optimized LDPC Encoder parameters. The lack of detail in DISCOVER definition about some of the tools and parameters used, we used similar approaches to some of the important architectural modules such as the frame interpolation module and reconstruction module which may explain the difference between two curves.

The implemented architecture is well-suited for a testing framework, therefore can be extended to any distributed coding based architecture.

## 5.2 Future Work

In the context of this thesis, the tools and algorithms implemented and evaluated were just the first effort by the author to develop a framework on distributed video coding field. The algorithms implemented allow reducing the performance gap of Wyner-Ziv video coding when compared to the traditional video coding systems; however considerably work still needs to be done in order to achieve the compression efficiency of the state-of-the-art traditional video coding standards like the ITU-T H.264 and MPEG-4 AVC. In this context, some possible future directions, extending the work described in this thesis, can be presented in the following:

*a) Lossy key frames:* The solutions implemented in this thesis are based on the assumption that the side information is generated using key frames perfectly available at the decoder, i.e. lossless key frame coding is assumed. In fact, depending on the target quality, a huge bitrate would be needed to provide such perfect key frame reconstruction at the decoder; abrupt variations on the decoded video quality would also be noticed (which is not visually pleasant for the user) since each Wyner-Ziv frame would very likely be encoded with a lower quality when compared with the key frames quality. In this context, the next step shall be assuming lossy key frame realistic

scenario. With lossy key frame coding, it will be necessary to determine the new correlation model between the side information (generated using lossy coded key frames) and the original frame. It will also be necessary to study the impact of the generated side information in the decoded video quality, within this more realistic scenario. Some work in this direction has already been developed [41] and [42].

*b) More accurate motion interpolation and extrapolation techniques at the decoder:* In a scenario characterized by a very lightweight encoder, the time consuming motion estimation/compensation task needs to be shifted to the decoder. Several frame interpolation techniques can be employed at the Wyner-Ziv decoder to generate the side information. The choice of the technique used can significantly influence the Wyner-Ziv codec rate-distortion performance; more accurate side information through frame interpolation means that the side information *Y* is more similar to the original frame *X* and therefore the decoder needs less bits from the encoder and thus the bitrate is reduced for the same quality. In order to obtain the same performance than the traditional video coding schemes, more powerful motion estimation and compensation techniques are necessary, preferably as efficient as the powerful tools included in the latest H.264/MPEG-4 AVC standard, where multi-frame prediction, variable block size motion compensation account for most of the performance gains. However, the traditional motion estimation and compensation techniques used at the encoder for hybrid video coding are not fully adequate to perform frame interpolation since they attempt to choose the best prediction for the current, i.e. known frame in the rate-distortion sense. For frame interpolation, it is essential to find an estimate (or a guess) of the current frame which makes the problem significantly different (the current frame is not available) and therefore new and improved techniques are needed. Since frame interpolation is performed based on past and future frames, this implies that the decoding order is not equal to the presentation order and introduces an extra delay; the extra amount of delay depends on how

far the future available reference is. For some types of applications, this is not acceptable (e.g. videoconferencing) and motion extrapolation techniques have to be used this means no future frames are used. Motion extrapolation is a more challenging task, since it relies only on past decoded frames and is not possible to obtain a motion trajectory between future and past frames (and thus more precise) as occurs with motion interpolation techniques.

*c) Channel codes:* The channel codes are a very important tool in DVC in order to correct the errors (that change over time) in the side information. Thus it is important to design channel codes adequate to the distributed video coding scenario (source coding). Some of the properties looked for are: i) coding of integer-valued sources with a high dynamic range, e.g. transform coefficients, ii) to work well under high compression ratios, i.e. highly punctured, iii) rate adaptation with minimal complexity when the source correlations change, and iv) performance close to the Information Theory bound, i.e. the Shannon limit.

*d) Rate control at the encoder:* Another challenge is to perform rate control at the encoder while still maintaining a low encoding complexity. This will avoid the use of the feedback channel in the current architecture and it will open the possibilities for new applications (e.g. broadcasting) where this channel is not physically available. This is however a challenging task, since the encoder does not know the quality of the side information (obtained at the decoder) and therefore can only estimate the bitrate needed to achieve a certain decoded image quality. Since the rate-distortion curve that helps to make this decision it is not known by the encoder, techniques to model the side information quality and perform rate decision are necessary in a practical coding scheme without feedback channel. Also regarding this topic, other encoder control decisions such as mode decision are also needed, e.g. the encoder must decide which frames or blocks are intra encoded, i.e. like in traditional video coding schemes, and which

are encoded in a distributed way. If no intra coding mode decision is used when the side information and the frame to be encoded have a weak correlation, e.g. scene cuts, uncovered areas, the encoder needs to send a high amount of bits in order to decode the bitplanes sent. When low temporal correlation exists, intra coding provides better performance since exploiting low temporal correlation will not bring any coding efficiency. The mode decision burden should be minimal in order to not compromise the encoder complexity.

*e) Adaptive GOP Size Selection:* The decision of setting a frame a key frame or Wyner-Ziv Frame is made according to GOP Size. Using longer GOPs are well-suited when the amount of motion is low throughout the whole sequence or the motion is predictable (*more temporal redundancy*), and using shorter GOPs are well-suited when the amount of motion throughout the sequence is high or the motion is unpredictable (*less temporal redundancy*). Therefore an adaptive GOP size selection approach can be employed at the encoder. In the literature GOP size selection is based on two concepts, Histogram based activity measurement along the video sequence and GOP length decision based on frame correlation.

*f) Minimum rate estimation:* The variance of the correlation noise between key frames and the Wyner-Ziv frames, which is a parameter of the virtual channel model, can be estimated at the decoder side and sent back to the encoder via the feedback-channel. Using this variance, the encoder can estimate a minimum number of accumulated syndromes to be sent per bitplane and per band to reduce the number of requests to be made by the decoder. Initial number of bits to be transmitted to decoder for each bitplane can be defined by minimum rate estimation mechanism. After sending minimal number of bits determined by minimum rate estimator, the buffer may iteratively transmit parity bits upon reception of request from the decoder via feedback channel. Therefore, it is important to implement a minimum rate estimator i) to minimize the number of bits initially transmitted, ii)

to reduce the number of syndrome requests which increase computational complexity and delay.

g) *Virtual Channel Modeling and Soft Input Calculation based on generated side information:* Taking into statistical modeling of the virtual channel, soft inputs can be calculated for the information bits. The implemented codec within the context of this thesis uses a constant Laplacian distribution [30] to model the correlation noise between corresponding DCT bands of side information and original Wyner-Ziv frame. This virtual channel can be modeled by generated side information. The virtual channel model can then be used to convert the side information DCT coefficients into soft-input information to the LDPC decoder. The conditional probability between Wyner-Ziv Frames (WZ) and Side Information (SI), namely *P(WZ|SI)*, obtained for each DCT coefficient will then be converted into conditional bit probabilities by considering the previously decoded bitplanes and the value of the side information.

Concluding, distributed video coding adopts a completely different coding paradigm by giving the decoder the task to exploit - partly or entirely - the source statistics to achieve efficient compression of the video signal. This new paradigm moves the bulk of the complexity from the encoder to the decoder, allowing the provision of efficient compression solutions with simple encoders and complex decoders. Therefore, it is a strong candidate for some emerging applications, e.g. multiview acquisition, wireless video, distributed mesh coding, sensor networks, disposable cameras, etc. In this thesis the major contributions were investigated to bring the coding efficiency of distributed video coding schemes nearer to hybrid video coding schemes thus paving the way for a breakthrough regarding the next video coding generation.

# REFERENCES

**[1]** T. Wiegand, G. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, July 2003.

**[2]** ISO/IEC International Standard 14496-10:2003, "Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding".

**[3]** D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. on Inform. Theory, vol. IT-19, pp. 471–480, July 1973.

**[4]** D.Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," IEEE Trans. on Inform. Theory, vol. IT-22, pp. 1–10, Jan. 1976.

**[5]** K. Romer and F. Mattern, "The Design Space of Wireless Sensor Networks", IEEE Wireless Communications, pp. 54-61, December 2004.

**[6]** S.Hadim, N. Mohamed, "Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks," IEEE Distributed Systems Online, vol. 7, no. 3, Mar., 2006

**[7]** I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," Ad Hoc Networks (Elsevier), vol. 3, no. 3, pp. 257-279, March 2005

**[8]** G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro, K. Ramchandran, "Image Compression in a Multi-camera System based on a Distributed Source Coding Approach", European Signal Processing Conference, Antalya, September 2005

**[9]** Advanced Multimedia Processing Lab, Carnegie Mellon, The Self-Reconfigurable Camera Array, January 29th 2008, http://amp.ece.cmu.edu/projects/MobileCamArray

**[10]** Computer Vision & Image Media Lab, Univ. of Tsukuba, "Free Viewpoint Browsing of Live Soccer Games", January 29th 2008, http://www.kameda-lab.org/research/project_3dvideo/soccer_3dvideo.jpg

**[11]** X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The DISCOVER codec: Architecture, Techniques and Evaluation", Picture Coding Symposium 2007, Lisbon, Portugal.

**[12]** A. Aaron, R. Zhang and B. Girod, "Wyner-Ziv Coding of Motion Video", Proceeding of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, USA, November 2002.

**[13]** A. Aaron, S. Rane, R. Zhang and B. Girod, "Wyner-Ziv Coding for Video: Applications to Compression and Error Resilience", Proceedings of IEEE Data Compression Conference, pp. 93-102, Snowbird, Utah, USA, March 2003.

**[14]** A. Aaron, E. Setton and B. Girod, "Towards Practical Wyner-Ziv Coding of Video", Proceedings of IEEE International Conference on Image Processing, Vol. 3, pp. 869-872, Barcelona, Spain, September 2003.

**[15]** S. Rane, A. Aaron and B. Girod, "Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting", Proceedings of SPIE Visual Communications and Image Processing Conference, San Jose, California, USA, January 2004.

**[16]** A. Aaron, S. Rane, E. Setton and B. Girod, "Transform-Domain Wyner-Ziv Codec for Video", Proceedings of SPIE Visual Communications and Image Processing , San Jose, California, USA, January 2004.

**[17]** R. Puri and K. Ramchandran, "PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles," Proc. Allerton Conf., October 2002.

**[18]** R. Puri and K. Ramchandran, "PRISM: A Video Coding Architecture Based on Distributed Compression Principles", ERL Technical Report, University of California, Berkeley, USA, March 2003.

**[19]** A. Aaron, S. Rane and B. Girod, "Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver", Proceedings of IEEE International Conference on Image Processing, Vol. 5, pp. 3097-3100, Singapore, October 2004.

**[20]** A. Aaron, D. Varodayan and B. Girod, "Wyner-Ziv residual coding of video," Proc. Picture Coding Symposium, PCS-2006, Beijing, China, April 2006.

**[21]** A. Aaron, S. Rane, D. Rebollo-Monedero and B. Girod, "Systematic Lossy Forward Error Protection for Video Waveforms", Proceedings of IEEE International Conference on Image Processing, Vol. 1, pp. 609-612, Barcelona, Spain, September 2003.

**[22]** A. Sehgal, A. Jagmohan, N. Ahuja, "A State-Free Causal Video Encoding Paradigm", Proceedings of IEEE International Conference on Image Processing, Vol. 1, pp. 605-608, Barcelona, Spain, September 2003.

**[23]** A. Aaron and B. Girod, "Wyner-Ziv video coding with low-encoder complexity," Proc. Picture Coding Symposium, PCS-2004, San Francisco, CA, December 2004. Invited paper.

**[24]** Image, Video, and Multimedia Systems Group, January 29th 2008, http://www.stanford.edu/~bgirod/Research.html

**[25]** D. Rowitch and L. Milstein, "On the Performance of Hybrid FEC/ARQ Systems using Rate Compatible Punctured Turbo Codes", IEEE Transactions on Communications, Vol. 48, No. 6, pp. 948–959, June 2000.

**[26]** S. Rane, A. Aaron and B. Girod, "Systematic lossy forward error protection for error-resilient digital video broadcasting - A Wyner-Ziv coding approach," Proc. IEEE International Conference on Image Processing, ICIP-2004, Singapore, Oct. 2004.

**[27]** D. Varodayan, A. Aaron, B. Girod: "Rate-adaptive distributed source coding using Low-Density Parity-Check codes" In: Proc. Asilomar Conference on Signals and Systems, Pacific Grove, CA, 2005.

**[28]** Luis Torres, Department of Signal Theory and Communications, Technical University of Catalonia, "DISCOVER - Distributed Coding for Video Services", January 29th 2008, http://www.discoverdvc.org

**[29]** B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed Video Coding", Proceedings of the IEEE, vol. 93, no. 1, January 2005.

**[30]** C. Brites, J. Ascenso, F. Pereira, "Studying Temporal Correlation Noise Modeling for Pixel Based Wyner-Ziv Video Coding", IEEE International Conference on Image Processing, Atlanta, USA, October 2006.

**[31]** Image Group - Instituto de Telecomunicações, Instituto Superior Técnico, "DISCOVER Codec Evaluation", January 29th 2008, http://www.img.lx.it.pt/~discover/home.html

**[32]** D. Varodayan, A. Aaron, B. Girod, "Rate-Adaptive Codes for Distributed Source Coding", EURASIP Signal Processing Journal, Special Section on Distributed Source Coding, vol. 86, no. 11, November 2006.

**[33]** R. Clarke, "Transform Coding of Images", Academic Press, San Diego, USA, 1990

**[34]** A. Liveris, Z. Xiong and C. Georghiades, "Compression of Binary Sources with Side Information at the Decoder Using LDPC Codes", IEEE Communications Letters, Vol. 6, No. 10, pp. 440-442, October 2002.

**[35]** A. Liveris, Z. Xiong and C. Georghiades, "Compression of Binary Sources with Side Information Using Low-Density Parity-Check Codes", Proceedings of IEEE Global Telecommunications Conference, Vol. 2, pp. 1300-1304, Taipei, Taiwan, November 2002.

**[36]** Rate-Adaptive LDPC Accumulate Codes for Distributed Source Coding, January 29th 2008, http://www.stanford.edu/~divad/ldpca.html

**[37]** W. E. Ryan, "An Introduction to LDPC Codes," in CRC Handbook for Coding and Signal Processing for Recording Systems (B. Vasic, ed.) CRC Press, 2004.

**[38]** D. Kubasov, K. Lajnef and C. Guillemot, "A Hybrid Encoder/Decoder Rate Control for Wyner-Ziv Video Coding with a Feedback Channel", Int. Workshop on Multimedia Signal Processing, Crete, Greece, October 2007.

**[39]** J. Ascenso, C. Brites and F. Pereira, "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding", 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, July 2005.

**[40]** L. Alparone, M. Barni, F. Bartolini and V. Cappellini, "Adaptively Weighted Vector-Median Filters for Motion-Fields Smoothing", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2267-2270, Georgia, USA, May 1996.

**[41]** A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Embedding a Block-based Intra Mode in Frame-based Pixel Domain Wyner-Ziv Video Coding", International Workshop on Very Low Bitrate Video - VLBV, Sardinia, Italy, September 2005.

**[42]** A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Improved Correlation Noise Statistics Modeling in Frame-based Pixel Domain Wyner-Ziv Video Coding", International Workshop on Very Low Bitrate Video - VLBV, Sardinia, Italy, September 2005.