



Politechnika Poznańska
Wydział Elektryczny
Instytut Elektroniki i Telekomunikacji
Zakład Telekomunikacji Multimedialnej i
Radioelektroniki
ul. Piotrowo 3A, 60-965 Poznań



Łukasz Błaszak

Advanced Scalable Hybrid Video Coding

Doctoral Dissertation
Advisor: Prof. Marek Domański

Poznań, 2006



Politechnika Poznańska
Wydział Elektryczny
Instytut Elektroniki i Telekomunikacji
Zakład Telekomunikacji Multimedialnej i
Radioelektroniki
ul. Piotrowo 3A, 60-965 Poznań



Łukasz Błaszak

Zaawansowane Skalowalne Hybrydowe Kodowanie Sygnałów Wizyjnych

Rozprawa Doktorska
Przedłożona Radzie Wydziału Elektrycznego
Politechniki Poznańskiej

Promotor: Prof. dr hab. inż. Marek Domański

Poznań, 2006

Contents

List of symbols and abbreviations	i
Chapter 1	1-1
Introduction.....	1-1
1.1. Scope of the dissertation	1-1
1.2. Thesis and goals of the dissertation	1-2
1.3. Overview of the dissertation	1-4
Chapter 2	2-1
Advanced Video Coding	2-1
2.1. Introduction.....	2-1
2.2. Coding Tools.....	2-3
2.2.1. Logical layers.....	2-3
2.2.2. Spatial prediction	2-3
2.2.3. Temporal prediction for variable block sizes	2-4
2.2.4. Motion vectors prediction	2-5
2.2.5. Motion estimation	2-5
2.2.6. Integer transform.....	2-5
2.2.7. In-loop deblocking filter	2-6
2.2.8. Adaptive entropy coding	2-7
2.2.9. Improvements of interlaced video coding	2-8
2.2.10. Multiple resolution tool	2-8
2.2.11. Fading compensation	2-9
2.2.12. Switching frames technique.....	2-9
2.2.13. Error resilience.....	2-9
2.3. Summary	2-10
Chapter 3	3-1
Scalable Video Coding	3-1
3.1. Introduction.....	3-1
3.2. Basic groups of scalable video codecs.....	3-2
3.3. Wavelet video codecs	3-3
3.3.1. Classification	3-3
3.3.2. Development of wavelet video coding	3-3
3.3.3. Implementation of motion-compensated temporal filtering	3-7
3.3.4. Scalability in wavelet video coding	3-9
3.4. Hybrid scalable coding	3-10
3.4.1. Temporal scalability	3-10
3.4.2. Spatial scalability	3-12
3.4.3. Quality scalability	3-13

Chapter 4	4-1
Multilayer Advanced Video Coding	4-1
4.1. Introduction	4-1
4.2. Spatial Scalability	4-3
4.3. Temporal scalability scenarios for scalable codec	4-5
4.4. Interpolation and Decimation	4-7
Chapter 5	5-1
Spiral scan	5-1
5.1. Introduction	5-1
5.2. Spiral scan in video compression	5-2
5.3. Spiral Scan for Quality Scalability in AVC Codecs	5-9
5.3.1 Introduction	5-9
5.3.2 Intra-frame prediction	5-12
5.3.3 Inter-frame prediction	5-16
5.3.4 CABAC coding	5-19
5.4. Model of codec with quality scalability	5-22
5.4.1 Overview	5-22
5.4.2 Spiral Scan in AVC - Complexity	5-25
Chapter 6	6-1
Codecs implementation – parameter setting	6-1
6.1. Introduction	6-1
6.2. Determining Huffman codes for symbols representing encoding modes for scalable H.263 codec	6-2
6.3. Determining encoding modes hierarchy for scalable H.264 codec	6-9
6.4. Determining k parameter for edge-adaptive bi-cubic interpolation for scalable H.264 codec	6-12
Chapter 7	7-1
Experimental assessment of the scalable video codecs	7-1
7.1. Introduction	7-1
7.2. Assessments using objective measure	7-1
7.2.1. Experiments for H.263 codec with spatio-temporal scalability	7-2
7.2.2. Testing of H.264 codec with spatio-temporal scalability	7-5
7.2.2.1. Comparison to simulcast and non-scalable H.264	7-5
7.2.2.2. Comparison to encoder proposed by the MPEG: JSVM 2.0	7-9
7.2.3. Experiments: comparison of non-scalable H.264 with raster scan and with spiral scan	7-14
7.2.3.1. Intra-frame coding test results	7-14
7.2.3.2. Inter-frame coding test results	7-23
7.2.4. Testing the H.264 with raster scan and spatial, temporal and quality scalability	7-31
7.3. Assessment using subjective measure	7-33
7.3.1. Introduction	7-33
7.3.2. Testing the scalable model with quality scalability based on JM 7.3 reference software	7-35

7.3.3. Testing the scalable model with quality scalability based on JSVM 1.0 software	7-38
Chapter 8.....	8-1
Conclusions.....	8-1
8.1. Summary	8-1
8.2. Original achievements	8-2
References.....	I
Author's contributions	I
References used in mention in the dissertation.....	IV
Annex A.....	A-1
Intra prediction.....	A-1
A.1. Intra prediction for chroma blocks.....	A-1
A.2. Intra prediction for luma 16x16 pixel blocks.....	A-9
Annex B.....	B-1
Proposal for slice header syntax	B-1

Abstract

This dissertation treats with digital video sequence coding by use of advanced scalable video codecs. In this work advanced scalable video codec based on multilayer coding structure has been proposed. The proposed codec is able to realize spatial, temporal and fine granular scalability. In the dissertation, the innovatory adoption of modified macroblock coding order technique has been proposed as a tool for fine granularity scalability. Several coding tools have been adapted to the spiral scan of macroblocks. These modifications increase encoding efficiency when the spiral scan is used. The encoder with the spiral scan and modified tools has the same coding efficiency as the encoder with the raster scan of macroblocks. The model of advanced video codec, proposed and tested by the author, described in this dissertation, is based on verification model of non-scalable H.264/AVC codec. The proposed codec is fully compatible with its non-scalable predecessor (H.264/AVC). The objective and subjective estimates of encoding efficiency for independent coding techniques as well as for the whole scalable codec have been performed. The coding efficiency has been compared to other well-known scalable video coding techniques.

Streszczenie

Rozprawa porusza zagadnienia kompresji sekwencji wizyjnych przy pomocy zaawansowanych kodeków skalowalnych. W pracy zaproponowany został zaawansowany kodek wizyjny o strukturze wielopętlowej. Kodek ten realizuje technikę skalowalności czasowej, przestrzennej oraz technikę skalowalności drobnoziarnistej. W pracy zaproponowano nowatorskie zastosowanie techniki zmienionej kolejności kodowania makrobloków jako narzędzia do realizowania skalowalności drobnoziarnistej. Zaadaptowano różne narzędzia kodowania do spiralnego uszeregowania kodowanych makrobloków. Modyfikacje te zwiększają wydajności kodera, ze spiralnym uszeregowaniem, tak by jego efektywność była nie mniejsza niż dla kodera ze standardowym uszeregowaniem makrobloków. Model zaawansowanego kodeka skalowalnego, zbudowanego i przebadanego przez autora, opisanego w pracy, bazuje na modelu weryfikacyjnym nieskalowalnego kodeka H.264/AVC. Zaproponowany kodek cechuje pełna kompatybilność z nieskalowalnym kodekiem H.264/AVC. Przedstawiono obiektywną oraz subiektywną ocenę wydajności zarówno poszczególnych technik kodowania jak i całego kodeka skalowalnego. Wydajność ta została porównana do wydajności innych znanych współczesnych metod skalowalnej kompresji sekwencji wizyjnych.

List of symbols and abbreviations

<i>2-D</i>	- two-dimensional,
<i>3-D</i>	- three-dimensional,
<i>4CIF</i>	- progressive 4:2:0 704×576 pixels video sequence,
<i>AC</i>	- DCT coefficient, for which the frequency in one or both dimensions is non-zero,
<i>AMC-FGS</i>	- Adaptive Motion-Compensated Fine Granularity Scalability,
<i>ASO</i>	- Arbitrary Slice Order,
<i>AVC</i>	- Advanced Video Coding,
<i>bitrate</i>	- number of bits per second,
<i>B-frame</i>	- bi-directionally inter-frame encoded frame used in non-scalable coding and base layer of scalable coding,
<i>BE-frame</i>	- B-frame occurring only in enhancement layer, it is not used as a reference frame,
<i>BR-frame</i>	- B-frame occurring only in enhancement layer, it is used as a reference frame,
<i>CABAC</i>	- Context-based Adaptive Binary Arithmetic Coding,
<i>CAVLC</i>	- Context Adaptive Variable Length Coding,
<i>CIF</i>	- progressive 4:2:0 352×288 pixels video sequence,
<i>CoI</i>	- Centre of Interest,
<i>DC</i>	- DCT coefficient with zero frequency in both dimensions,
<i>DCT</i>	- Discrete Cosine Transform,
<i>DP</i>	- Data Partitioning,
<i>DPCM</i>	- Difference Pulse Code Modulation,
<i>DWT</i>	- Discrete Wavelet Transform,
<i>EZBC</i>	- Embedded image coding algorithm using ZeroBlocks of subband/wavelet coefficients and Context modeling,
<i>EZW</i>	- Embedded Zerotree Wavelet,
<i>FGS</i>	- Fine Granularity Scalability,
<i>FIR</i>	- Finite Impulse Response,
<i>FMO</i>	- Flexible Macroblock Order,
<i>fps</i>	- frames per second,

<i>GOP</i>	- Group of Pictures,
<i>HH</i>	- high-high-spatial frequency subband,
<i>HL</i>	- high-low-spatial frequency subband,
<i>IBMATF</i>	- In-Band Motion Aligned Temporal Filtering,
<i>IDR</i>	- Instantaneous Decoding Refresh,
<i>I-frame</i>	- Intra-frame encoded frame,
<i>IDCT</i>	- Inverse Discrete Cosine Transform,
<i>Intra</i>	- intra-frame,
<i>ITU</i>	- International Telecommunication Union,
<i>JM</i>	- Joint Model,
<i>JSVM</i>	- Join Scalable Video Model,
<i>JVT</i>	- Joint Video Team,
<i>Kbps</i>	- kilobits per second,
<i>LBR</i>	- Low Bit Rates,
<i>LH</i>	- low-high-spatial frequency subband,
<i>LL</i>	- low-low-spatial frequency subband,
<i>LPS</i>	- Least Probable Symbol probability,
<i>LZW</i>	- Lempel-Ziv-Welch,
<i>MBAFF</i>	- MacroBlock Adaptive Frame Field,
<i>MC-EZBC</i>	- Motion Compensated - Embedded image coding algorithm using ZeroBlocks of subband/wavelet coefficients and Context modeling,
<i>MCTF</i>	- Motion Compensated Temporal Filtering,
<i>MOS</i>	- Mean Opinion Score,
<i>MPEG</i>	- Motion Pictures Expert Group,
<i>MV</i>	- full-frame motion vectors,
<i>NAL</i>	- Network Abstraction Layer,
<i>PFGS</i>	- Progressive Fine Granularity Scalability,
<i>PicAFF</i>	- Picture Adaptive Frame Field,
<i>PSNR</i>	- Peak Signal to Noise Ratio,
<i>QCIF</i>	- progressive 4:2:0 176×144 pixels video sequence,
<i>QP</i>	- Quantization Parameter,
<i>RD</i>	- Rate Distortion,
<i>RoI</i>	- Region of Interest,
<i>RS</i>	- Redundant Slices,

<i>SDMATF</i>	- Spatial-Domain Motion Aligned Temporal Filtering,
<i>SNR</i>	- Signal to Noise Ratio,
<i>SPIHT</i>	- Set Partitioning in Hierarchical Trees,
<i>SSMM</i>	- Single Stimulus MultiMedia,
<i>SVM</i>	- Scalable Video Model,
<i>UVLC</i>	- Universal Variable Length Coding,
<i>VCEG</i>	- Video Coding Experts Group,
<i>VCL</i>	- Video Coding Layer,
<i>VLC</i>	- Variable Length Coding,

Chapter 1

Introduction

1.1. *Scope of the dissertation*

Since the early 90's, significant progress of digital video coding techniques has been made. Up to day the encoding technique efficiency has been considerably improved. Although during that time, only a few generations of video codecs have been developed, their performance and applications have been extended significantly. Among various proposals of video coding algorithms, the major technology, which has been approved by commercial market, was hybrid coding with motion-compensated prediction and block-based transform coding. Each newly developed technique of video coding was subjected to the standardization process. The three groups of standards are listed below:

- MPEG-1 [ISO93], H.261 [ISO90];
- MPEG-2 [ISO94], MPEG-4 [ISO94], H.263 [ISO96];
- H.264 [ISO-AVC], VC-1[SMPTE05].

Each consecutive standard covered techniques of significantly better encoding performance. But together with increasing performance the algorithm complexity has also grown.

The techniques, developed since that time, were mostly related to non-scalable coding, i.e. if the available throughput is smaller than the required bitrate, transmission is not possible. The appearance of new network technologies causes a problem for video coding and video transmission. Because of connecting various network technologies, channel capacity between a video transmitter and a video receiver would become time-variant or would depend on the receiver location. Thus, it became difficult to estimate the bitrate for encoded video bitstreams. And here the scalable video technology would help. The scalable video coding is coding of embedded bitstreams, each representing a

different level of quality. Thus, the decoder of scalable video bitstreams is able to decode video sequences by use of whole or only part of the bitstream received. By receiving consecutive embedded bitstreams the quality of decoded video sequence is getting better and better.

The functionality of scalability has already been present in MPEG-2 but the technique used for this standard was inefficient, and for that reason it was rarely used. But meanwhile some better techniques have been developed [Dom00c, Dom01, Li01, Rad99b].

Recently developed and also standardized technology is advanced video coding. Many new tools this technology consists of make it very efficient in comparison to earlier technologies. There are two main advanced video coding technologies developed at the same time which have been standardized: H.264 and VC-1.

The advanced video coding is a very flexible technology because of multiplicity of different coding tools it consists of but it lacks the functionality of scalability. There is a need for such a tool which would provide a codec with scalability while maintaining high encoding efficiency.

There may be different types of scalability and application scenarios. The quality may be reduced by dropping some video frames as well as by decreasing spatial resolution. There is also a so called SNR scalability where the temporal and spatial resolution remains unchanged but the number of details of video sequence is reduced. The spatial scalability should be applied when the receiver's resolutions differ, for example: one is a standard TV monitor and another is a cell phone. The temporal scalability may be used wherever the picture quality and resolution should be the highest, for example in security systems. The SNR scalability may be used for broadcast TV, internet TV where the decoded video sequence is expected to be fluent and at constant resolution.

1.2. Thesis and goals of the dissertation

Goals:

The goals of the work are the following:

- To propose tools which allow adding the functionality of scalability to advanced video coding techniques.

- To propose consistent technology of scalable video coding that should be as compliant as possible with the existing standards of advanced video coding.
- The proposed scalable video coding technology should provide high compression efficiency, close to that of modern advanced video coding.
- The proposed scalable video coding techniques should be assessed by experimental comparison to the existing advanced video coding techniques.

Requirements:

- The proposed tools for scalable coding should be as simple as it is possible in order not to increase excessively codec structure complexity.
- The proposed techniques should be suitable for systems with low encoding and decoding delays.
- The encoder with proposed tools embedded should be backward compatible with non-scalable decoders mentioned above.

Thesis:

It is possible to enhance advanced video codecs for scalable coding and achieve high compression performance by the use of limited set of new tools.

Methodology:

The proposal of new tools and techniques will be prepared on the basis of the studies of the bibliographic references as well as experience on advanced video coding and designs of scalable codecs for classic coding technology. The assessment of the proposed tools and techniques will be done by means of a set of experiments. The experiments will be performed by use of the experimental model designed and built by the author. This experimental model will be software based on existing software verification models for advanced video coding. There are two standard advanced video codecs widely used:

- AVC (ISO/IEC 14996-10 known also as H.264 [ISO-AVC]),
- SMPTE VC-1 (M421)/Windows Media 9 [SMPTE05].

The first codec has been well documented in an international standard since its very beginning. Moreover, a reference software implementation is publicly available. The specification of the other one has become public only very recently and was not available at the time of the author's work on this doctoral dissertation. Therefore, AVC/H.264 codec has been chosen as the reference for the experiments.

Two experimental software implementations have been used by the author. The first uses the verification model of scalable codec, created by the author together with other scientists from Poznań University of Technology, on the basis of the AVC/H.264 reference software ver. 7.3 (JM 7.3). In this dissertation also another scalable codec verification model was used. It was recently available JSVM [ISO-JSVM] (Join Scalable Video Model) ver. 2.0 extended with the tools proposed by the author.

1.3. Overview of the dissertation

This dissertation describes the results of the research that the author made when he was an active developer of the new international scalable advanced video coding standard. The research was constructive, i.e. a proposal of codec with a set of tools is described. This codec and coding tools have been proposed during the standardization process. The resulted codec exhibits efficiency of coding similar to the codec which has been chosen as an international standard.

This doctoral dissertation consists of eight chapters following this introduction. Chapter 2 describes in general advanced video coding techniques, new tools used for video encoding and the main leading technologies used nowadays. Chapter 3 describes scalable video coding techniques which have been developed recently including techniques which are still under development. Chapter 4 consists of the description of multilayer advanced video codec developed by the author. The description includes the generic structure of codec and author's inventions. Next, Chapter 5 describes specific order of macroblock scan with special modifications of data prediction proposed by the author for hybrid DCT-like codecs. Chapter 6 shows how the author has set the parameters for verification models. The results of several experiments, used for setting the codec parameters, are included there. In Chapter 7 all experiments concerning verification of proposed scalable video codec's performance are accumulated. Chapter 8 summarizes this doctoral dissertation.

Chapter 2

Advanced Video Coding

2.1. Introduction

The history of hybrid video codecs begins in the early 80's. At the beginning DPCM, scalar quantization and variable-length coding were used for the video coding compression. Those tools were used for defining the first international standard of digital video coding which was ITU-T (ex-CCITT) Rec. H.120. In the late 80's, the motion compensation and background prediction were used for video coding and they were also added to the second version of H.120 standard. Although some of these tools have been used so far, the H.120 standard is essentially not in use today.

Beginning from the early 90's, the video frame of video sequence has been partitioned into blocks. Each block is encoded separately by use of a motion-compensated DCT-like transform, and then frequency domain coefficients are quantized and then encoded by means of Huffman entropy coder.

Based on those techniques the first widespread practical success of digital video coding was the H.261 [ISO90] version 1 in 1990 and version 2 in 1993. Soon after, in the same year, the MPEG-1 [ISO93] was introduced, and later in 1994 the MPEG-2/H.262 [ISO94]. The standard H.263 [ISO96] was released in three versions: version 1 in 1995, version 2 in 1998 and version 3 in 2000. Meanwhile, the MPEG-4 (Part 2) [ISO99] was proposed in 1999. All these standards were designed for specific applications such as video conferencing, broadcast television, etc.

In the 90's there was a great progress in video compression. A lot of new techniques were developed and enhanced, such as motion vector prediction, spatial

prediction, filtering of block transform artifacts, application of multiple reference pictures for motion estimation. The motion estimation accuracy was enhanced by use of reference picture interpolation. At the end of the 90's the object based coding was proposed. The picture there was treated as a scene containing various objects. Each object could be encoded separately and its position could be freely changed at the decoder side. At the same time the body and face animation were proposed. Object based coding and body and face animation became part of MPEG-4 (Part 2) [ISO99] standard.

At the beginning of this century scientists decided to create a new video compression technique which would combine existing most successful tools with new tools. Most powerful tools were taken, some were modified such as block sub-partitioning, motion vector prediction, spatial prediction, deblock filtering, and some new tools were added such as fully reversible integer transform, new techniques of entropy coding (CABAC, CAVLC). On the basis of those tools two competitive video coding standards were proposed. It was ITU-T H.264/AVC [ISO-AVC] and VC-1 [SMPTTE05].

By the standardization process the application range for these codecs became quite wide. All above techniques were, at the beginning, designed to serve different application domains. Thus, the H.261 was designed to be used in video telephony for a kind of network as ISDN (Integrated Services Digital Network), H.263 and MPEG-4 intended to be used in network video communication for a kind of network as PSTN – Public Switched Telephone Network, but also in the Internet and in mobile networks. MPEG-1 was used for consumer video on CD, while MPEG-2 was used on DVD. The last one was mainly designed to be used for broadcast of standard definition or high definition TV but also, together with MPEG-4, they were used for network video communication in ATM (Asynchronous Transfer Mode) networks. During all the years, when these techniques were widely used, the borders between initial applications disappeared.

2.2. Coding Tools

The newest and most advanced coding technology uses various tools and techniques to achieve the best encoding performance. Currently, there are two codecs based on architecture of motion-compensated DCT-like transform which may be classified as advanced video codecs. These are H.264/AVC and Windows Media 9. Some of the tools and techniques they use are very similar. But by the use of different algorithms they achieve almost the same encoding efficiency. Because most of the techniques used by these codecs are similar, they will be described on the basis of one of the codec.

2.2.1. Logical layers

Advanced video codecs were designed to cover a wide range of applications. It means that such codecs have to be able to produce bitstreams that may be transmitted by almost any kind of network. Because of this, the structure of the encoded bitstream needs to be very adaptive. A good example of such a structure may be presented by the use of H.264/AVC codec.

For example the hierarchical structure [Wie03a, Sul04], for H.264/AVC, of video sequence is as follows: The sequence is composed of pictures; The picture is divided onto slices which can be of different sizes; The slices are composed of macroblocks which are block of 16×16 pixels; Each macroblock can be divided into partitions which are block of 16×8 , 8×16 or 8×8 pixels; The partitions of 8×8 may be divided into sub-partitions which are blocks of 8×4 , 4×8 or 4×4 pixels.

Thus, Advanced Video Codec H.264 defines two separate abstract layers: Video Coding Layer (VCL) which corresponds to the slice layer bits and Network Abstraction Layer (NAL) which corresponds to the higher layer level bits. The NAL was designed to easily adapt the produced bitstreams to variety of delivery frameworks (e.g. broadcast, wireless, and storage media).

2.2.2. Spatial prediction

Spatial prediction of macroblock samples is a technique used in H.264/AVC [ISO-AVC] as well as in Windows Media 9 encoder [SMPTE05].

Intra coding [VCEG-N54, VCEG-O31, VCEG-O48 and Ric03] may exploit the spatial correlation among pixels by use of the spatial prediction. There may be several algorithms of spatial prediction based on the full-macroblocks (16×16 blocks) or based on 8×8 blocks or even 4×4 blocks. The pixels may be predicted from various spatial directions. As an example see Fig. 2.1, where the directions of spatial prediction for H.264 encoder are proposed.

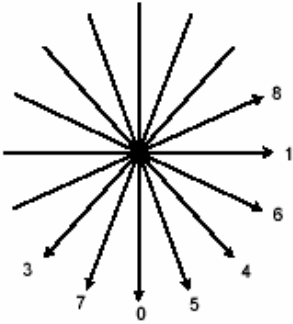


Fig. 2.1. Example of directions for spatial prediction of a 4×4 block in H.264/AVC standard.

2.2.3. Temporal prediction for variable block sizes

The temporal prediction can be made for variable block sizes. For instance, H.264/AVC [ISO-AVC] uses the following block sizes: 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 (see Fig. 2.2). For each block separate motion estimation can be made.

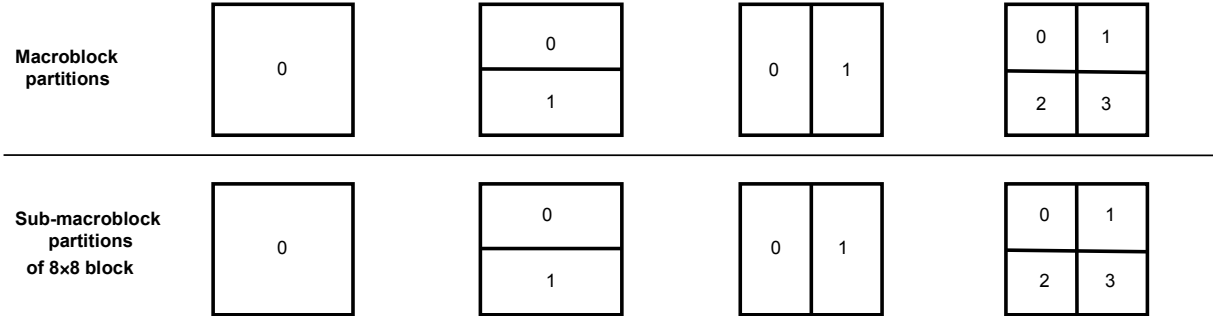


Fig. 2.2. Macroblock partitioning for motion estimation and compensation.

2.2.4. Motion vectors prediction

Motion vectors, before being sent to the decoder, are first predicted by the use of upper and left neighboring motion vectors. There are several ways of predicting motion vector depending on the availability of the neighboring block. The prediction may be a value of motion vector from one of the neighboring blocks or a median value of motion vectors taken from three neighboring blocks. For the *Direct* mode the motion vector prediction is made by means of either motion vectors from spatially neighboring blocks or by use of motion vectors from blocks from reference frames. Because of the fact that a motion vector may point to several different reference frames, the prediction of the motion vector is made only by use of other vectors which are pointing to the same reference frame.

2.2.5. Motion estimation

The motion estimation can be made in two directions forward or backward (or both ways) [Wed03, Fli03]. The block can be predicted from several reference pictures. Pixel values of the reference picture are first interpolated to achieve quarter-pixel accuracy for luminance and for chrominance. For example, in order to create half-pixel values in H.264/AVC, the following filter is used: $[1 \ -5 \ 20 \ 20 \ -5 \ 1]/32$. Filtering is done separately horizontally and vertically. To achieve quarter-pixel accuracy, the interpolation for luminance is performed by averaging two nearby values horizontally, vertically or diagonally, of half-pixel accuracy. The H.264/AVC may use also a special mode called the *Direct* mode, for which the motion vectors are not explicitly sent but they are derived by scaling the motion vectors of the co-located macroblock in another reference frame or derived by inferring motion from spatially-neighboring regions.

In the case of bi-directionally predicted frames the special weighted prediction can be applied. The encoder can use weighted average between two predictions for bi-prediction. This tool is very useful for such phenomena as “cross-fades” between different video scenes.

2.2.6. Integer transform

After the prediction of encoded samples, a transform is applied to decorrelate the data spatially. This transform is made for 4×4 block sizes, and this integer transform is

fully invertible, it does not use floating point arithmetic. The following transform was adapted in H.264 encoder but similar technique was used for Windows Media 9 encoder. The new transform [Wie03b] was proposed and its main features are the following:

- It is a fully invertible integer transform;
- For 8-bit input data precision the 16-bit arithmetic is sufficient for transform implementation;
- The transform and quantization are low-complexity [Mal03], the 4×4 block size transform can be implemented using just a few additions, subtractions, and bit shifts;

$$T_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

In addition, if while encoding process the 16×16 Intra prediction mode was used with the 4×4 block size transform, for the DC coefficients of all 4×4 luminance blocks the Hadamard transform is used.

$$H_{2 \times 2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

2.2.7. In-loop deblocking filter

Another tool that significantly improves subjective and objective image quality is deblocking filter [Lis03, VCEG-M20, JVT-B039]. It reduces the blockiness introduced in a picture. It is an adaptive filter which adjusts its strength depending on the quantization parameter, motion vector, frame/field coding decision and the values of the pixels. The higher the quantization is the stronger filtering is applied. In the case of the H.264/AVC this tool is used inside coding-decoding loop.

An alternative to deblocking filtering used in Windows Media 9 [SMPTE05] is overlap smoothing. It is a technique to achieve block edge smoothing using a simpler

operation than in the case of deblocking filtering, which is very complex because of non-linear operation. Overlap smoothing is achieved by the use of lapped transform [Tra03]. It is a transform whose input spans, besides the data elements in current block, a few adjacent elements in neighboring blocks.

2.2.8. Adaptive entropy coding

The tool, that is used recently to remove redundant information from encoded data is lossless coding technique – entropy coding. There are many algorithms for lossless entropy coding; the most basic ones are Huffman coding [Huf52], arithmetic coding [Wit87] and dictionary technique LZW (*Lempel-Ziv-Welch*) [Ziv77, Ziv78]. The entropy coding is used to replace the data elements with coded representation, at the same time reducing the remaining correlation between data elements, thus reducing the data size. Data to which the entropy coding can be applied are for example residuals of motion vectors prediction, transform coefficients, etc. Two entropy codecs have been used [VCEG-L13, VCEG-M59] for advance video coding for H.264/AVC [Mar03] and one for Windows Media 9 briefly described in [Sri04]. Here, the author focuses on H.264 entropy coding algorithms.

The first entropy codec used for advanced video coding is based on VLC (*Variable Length Coding*) technique. Here, the exponential Golomb Code, also called UVLC (*Universal Variable Length Coding*), has been used. Codes of UVLC have generic form of following bits: [ZEROS][1][DATA], where DATA is a binary representation of an unsigned integer and ZEROS are number of zeros equal to length of DATA. The given input data elements can be mapped to any other data depending on the statistical probability of encoded data occurring frequency. There may be defined several code tables and the selection of currently used table may be based on some context. Thus, the technique may become the CAVLC (*Context Adaptive Variable Length Coding*), giving a tradeoff between speed of execution and performance.

The other one, entropy codec CABAC (*Context-based Adaptive Binary Arithmetic Coding*) [Mar03] increases compression efficiency by roughly 10% relative to the CAVLC, but at the cost of additional complexity. CABAC uses as a base an arithmetic coding technique which permits assigning a non-integer number of bits per symbol. Generally, the CABAC entropy coding scheme consists of the following steps (see Fig. 2.3): a suitable model is chosen according to a set of past observations of relevant

elements (each element such as motion vector or transform coefficient has its own model), then if a given symbol is non-binary valued, it will be mapped onto a sequence of binary decisions, so-called *bins* and each bin is encoded with an adaptive binary arithmetic coding engine.

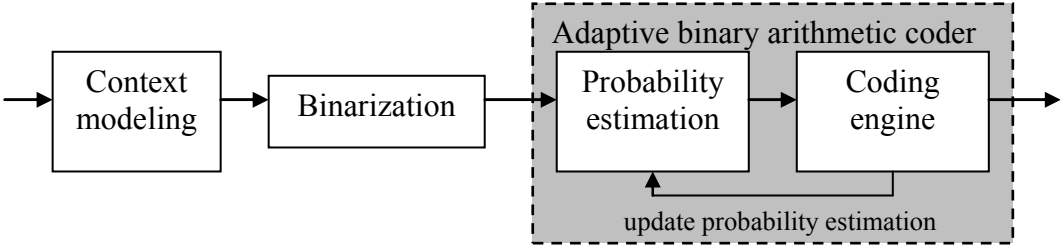


Fig. 2.3. Generic block diagram of the CABAC entropy coding scheme.

2.2.9. Improvements of interlaced video coding

The advanced video coding technique supports also coding of the interlaced video sequences. The H.264/AVC and Windows Media 9 may encode a video sequence in two modes: field picture coding mode and frame picture coding mode. The first one encodes each field separately and the second one encodes both fields jointly. In the case of H.264/AVC there were designed, additionally, special adaptive techniques for coding interlaced video sequences: Picture Adaptive Frame Field (PicAFF) which allows for switching coding technique between frame and field coding mode for each picture; and MacroBlock Adaptive Frame Field (MBAFF) which allows for switching coding technique between frame and field mode for each pair of macroblocks. These techniques significantly improve coding efficiency of interlaced video input.

2.2.10. Multiple resolution tool

There have also been proposed a tool for low bit rates (LBR) scenarios [Sri04]. The tool enables to encode frames at multiple resolutions. It is obtained by scaling down one or two dimensions of each coded frame. At the decoder site the frame is up-scaled by the factor received from the encoder. This technique has been applied into Windows Media 9 encoder [SMPTE05].

2.2.11. *Fading compensation*

Another technique proposed was fading compensation technique [Sri04] which can significantly reduce the number of bits needed for global illumination changes. The effects that can be compensated are for example fade-to-black or fade-from-black. Standard motion-compensation technique is ineffective for frames with such effect.

2.2.12. *Switching frames technique*

One of the innovations included into H.264/AVC are new picture coding types. These are switching frames [Kar03], both intra and inter coded. Their main feature is ability to reconstruct specific exact sample values, even when using different reference pictures or a different number of reference pictures in prediction process. Such frames permit bitstream switching, random access, and fast forward, reverse, and stream splicing.

2.2.13. *Error resilience*

Tools which have specific application for network transmission are Error Resilience Tools [Wen03, Sto03]. These tools have been used in H.264/AVC codec. There are several tools which have to protect video bitstream from network transmission problems. These tools are: Flexible Macroblock Order (FMO), Arbitrary Slice Order (ASO), Redundant Slices (RS), or Data Partitioning (DP). By the use of FMO the macroblock may be sent randomly, and when a segment of data is lost, errors are distributed randomly over the video frame. When ASO is used, the slices may be received in any order. RS increase bitstream protecting by repeat sending the slices representing the same part of a picture. By the use of DP it is possible to separate the coded slice data into separately-decodable sections and protect each one with the protection level according to how important these data are.

2.3. Summary

Advanced video coding is a set of several advanced tools and techniques that improve encoding efficiency and error resiliency, but at the same time they increase encoding complexity. These tools combined together became a potentially very powerful coding technique, which can be much more efficient than widely used existing techniques. But, even though, advanced video coding has a great potential, there are no tools for scalable coding. While this doctoral dissertation was written, several groups of experts, including the author of this dissertation, were working upon the scalable tools for advanced video coding. Several solutions have been proposed, whose general description may be found in the next chapter. And, the author's proposal is described in detail later in this work.

Chapter 3

Scalable Video Coding

3.1. Introduction

The scalable video coding may be defined as the ability of the encoder to produce the bitstream which consists of layers, each representing different spatial resolution, different temporal resolution or signal-to-noise ratio (SNR). Scalable encoded video bitstream consists of embedded bitstreams each representing a single layer. The more embedded bitstreams are received, the higher quality the decoded video sequence is. Thus, there are three types of scalability.

The first one, called spatial scalability, enables encoding one video sequence represented by several spatial resolutions into one bitstream. Depending on the channel capacity the decoder may receive and decode only part of the bitstream or the whole bitstream, achieving lower spatial resolution video sequence or full spatial resolution video sequence respectively.

Another one is temporal scalability which enables to produce a bitstream in such a way that the decoder may decode from part of the bitstream a video sequence with reduced frame-rate. It means that, for example, every second frame was dropped and remaining frames have full spatial resolution and image quality.

The last one is quality (SNR scalability) scalability which enables to generate an encoded bitstream from the input video sequence which contains layers, each representing a different image quality. The output video sequence quality depends on the number of layers the decoder is capable to receive. The decoder always decodes full spatial and temporal resolution video sequence but the quality may be reduced. The

quality scalability may be obtained by Fine-Granularity-Scalability (FGS) which is functionality for precise tuning of a bitstream. For this technique a single packet of data may be very small, thus, the bitstream may be cut near the value of maximum channel capacity.

There are two major groups of approaches to achieve scalability [Ohm01]: wavelet based techniques and extensions of the hybrid transform codecs. The wavelet codecs, such as 3-D wavelet codecs, naturally enable scalable encoding [e.g.: Ohm02, Woo02, Hsi01]. The hybrid coder structures based on motion-compensated prediction and transform block coding were designed for non scalable coding of video sequences.

Both techniques are still in competition, trying to achieve better results in both scalable and non scalable coding.

3.2. Basic groups of scalable video codecs

Motion pictures codecs may be divided into two groups: hybrid-codecs with DCT-like transform and wavelet codecs. The main difference between them is how they deal with scalability.

This first group has been mainly designed for non-scalable purposes. The most efficient techniques of this group have been described in the previous chapter of this doctoral dissertation. Scalability has been developed due to the requirements concerning new applications. Joint work of scientists and industrial experts has led to achieving a very powerful hybrid-scalable-codec with DCT-like transform [ISO-JSVM].

The other group of codecs which is based on wavelet transform was designed from the beginning as a scalable coding technique. From their beginnings, wavelet coding techniques have provided spatial scalability and quality scalability. Those codecs were called 2D codecs. This kind of coding was presented in [Sha93, Tau94, and Sai96]. Next, the input video signal wavelet analysis may be extended onto time domain and owing to this, the three-dimensional wavelet-based video coding appears. Those codecs were given the name of 3D codecs. Various 3D wavelet codecs may be found in [Kar88, Kro90, Ohm93, Ohm94, and Cho99].

Most recent proposals of the structure of scalable hybrid codecs include a technique of temporal analysis taken from wavelet codecs. Because of this fact, the name of 3D codecs was also extended to hybrid scalable codecs.

3.3. Wavelet video codecs

3.3.1. Classification

Scalable video codecs which use motion compensated temporal filtering may be divided into three categories: T+2D, 2D+T and multi-T+2D. Such a classification was discussed in [ISO04d, Xio05a].

The T+2D category, also called the spatial-domain motion aligned temporal filtering (SDMATF) scheme, applies temporal filtering at the encoder side to input video frames directly before spatial decomposition. At the decoder site temporal filtering is done at the resolution of target output video. Here, in this scheme, when the target resolution at the decoder side is lower than input video frame resolution, the artifacts appear in the regions with complex motion. This scheme was used in [Xio04, Xio05b].

The 2D+T category, also called in-band motion aligned temporal filtering (IBMATF) scheme, applies temporal filtering after spatial decomposition. Filtering is done for each subband. Because of the fact that MCTF is done after spatial decomposition, after decimation procedure, critical-sampled wavelet transform is only periodically shift-invariant and has certain aliasing effects. This scheme was used in [And04, Li04, Par00, and Ye03].

The last scheme is a category multi-T+2D. This category was presented in [ISO04e, ISO04f]. Here, the input video frame is first down-sampled and so various resolutions are generated. In this way the redundant pyramid representation of the original frame is produced. For this scheme, motion compensated temporal filtering is performed on each of these spatial resolution layers. Here no mismatch between the encoder and the decoder is present. Additionally, the motion compensation in the image domain before spatial transform generally has a higher accuracy and efficiency than when applied in the subband domain of lower resolution. But the disadvantage is high redundancy in the overlapped spatial-temporal subbands across various spatial resolution layers. Therefore, this redundancy has to be reduced somehow.

3.3.2. Development of wavelet video coding

The wavelet decomposition of video sequence, applied to spatial domain and temporal domain, is used by wavelet codecs to generate embedded bitstreams, each

representing different spatial and/or temporal resolution. Each embedded bitstream contains encoded data of a subband. The difference between various wavelet encoders is how the data from subbands are encoded.

Here in this Section, several techniques are described beginning from two dimensional coding, later enhanced to three dimensional coding and later extended by, for example, motion compensation.

One of the first successful codecs based on wavelet transform, was EZBC (*Embedded image coding algorithm using ZeroBlocks of subband/wavelet coefficients and Context modeling*), belonging to the 2D codecs group. This is a technique of coding based on other successful techniques: embedded zero-tree/-block coding (EZW) [Sha93] and context modeling of the subband/wavelet coefficients [Tau94]. In zero-tree/-block encoder two facts were taken into account. One is that most energy of the signal in frequency domain is accumulated near low frequencies. Another is that there is a strong correlation between data in subbands of wavelet decomposition. An EZBC was proposed in [Hsi00]. The authors adopted the adaptive quadtree splitting method [And97] to separate significant coefficients and then encode every block of zero pixels into one symbol. In the EZBC the context models were designed for coding quadtree nodes at different tree levels and subbands. This encoder is inherently applicable to resolution scalable applications. A few years later Said and Pearlman proposed in [Sai96] a set partitioning in hierarchical trees (SPIHT) image codec which was a 2D codec. This codec consists of three basic concepts:

- coding and transmission of important information first based on bit-plane representation of pixels,
- ordered bit-plane refinement,
- coding along preferred path/trees called *spatial orientation trees*.

Later in [Pea98] 3D SPIHT coding was proposed. This technique also consists of three parts:

- partial ordering by magnitude of the 3D wavelet transformed video with a 3D set partitioning algorithm,
- ordered bit-plane transmission of refinement bits,
- exploitation of self-similarity across spatio-temporal orientation trees.

However, the first time the three-dimensional coding was proposed by Karlsson and Vetterli in [Kar88], where a simple 2-tap Haar filter was used for temporal filtering.

Later on, based on the work of Kronander [Kro90], where motion compensated temporal filtering within the 3D subband coding framework was presented, Ohm introduced an idea for a perfect reconstruction filter with block-matching algorithm [Ohm93, Ohm94]. Similar work has been done by Choi and Woods in [Cho99]. The structure of temporal decomposition is shown in Fig. 3.1, and the 3D subband structure in GOP is shown in Fig. 3.2.

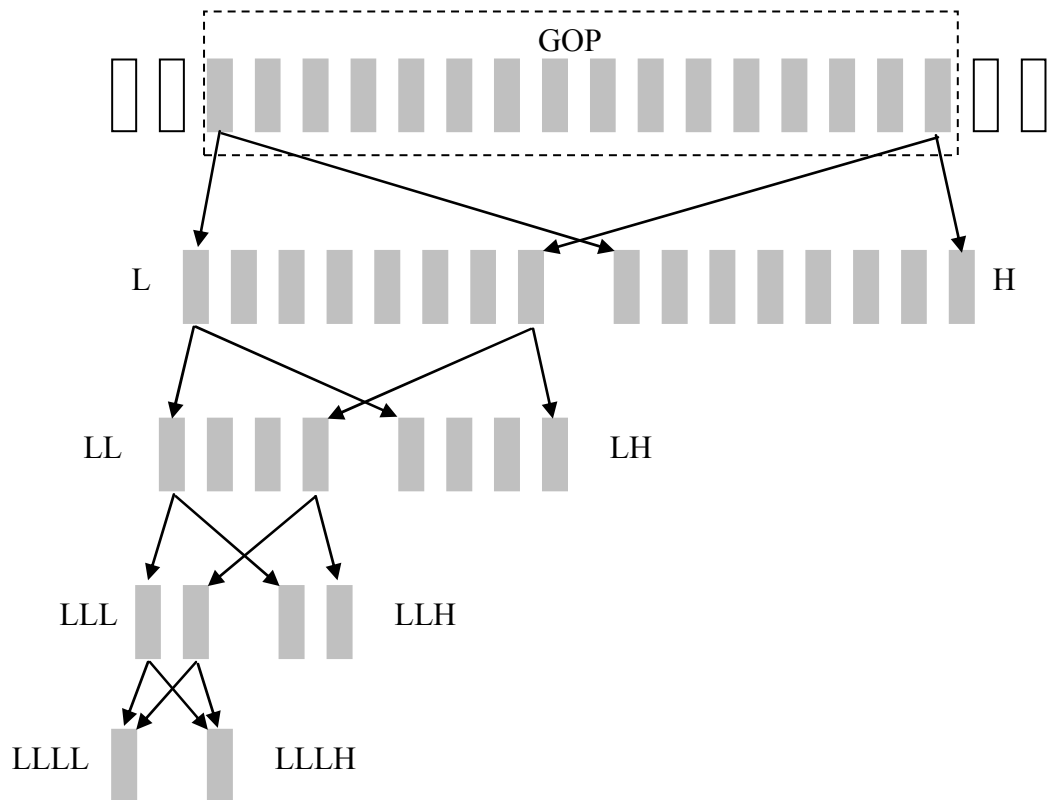


Fig. 3.1. Octave based five-band temporal decomposition.

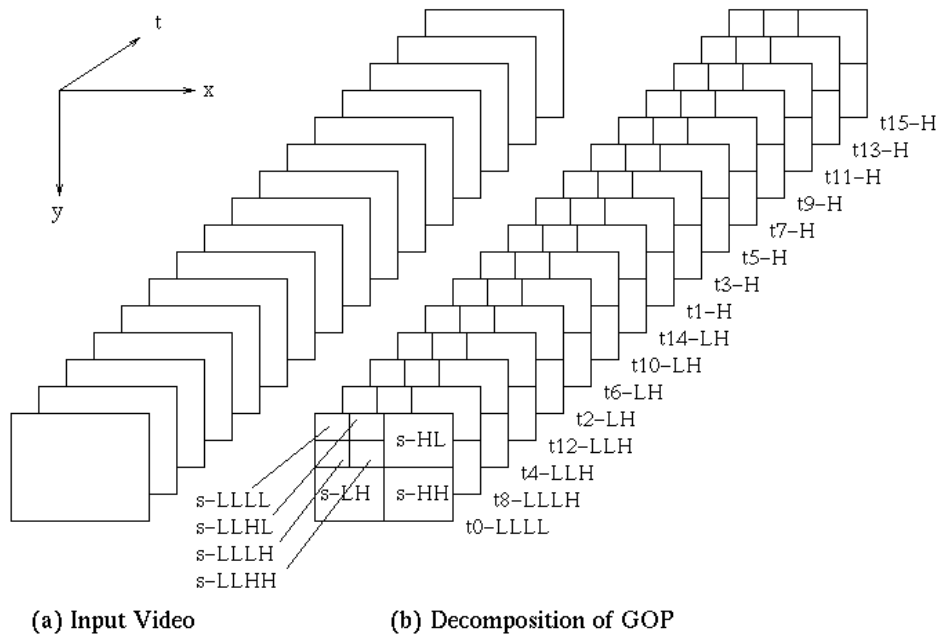


Fig. 3.2. 3-D subband structure in a GOP.

Later Hsiang and Woods proposed an enhancement of EZBC technique known as the MC-EZBC (Motion Compensated - Embedded image coding algorithm using ZeroBlocks of subband/wavelet coefficients and Context modeling), that belongs to the group of 3D wavelet codecs. MC-EZBC [Hsi01] is a video coding technique using 3-D subband/wavelet transform along the motion trajectory. It exploits temporal correlation and is fully embedded in quality/bitrate, spatial resolution and frame rate. The basic structure of the coder is shown in Fig. 3.3.

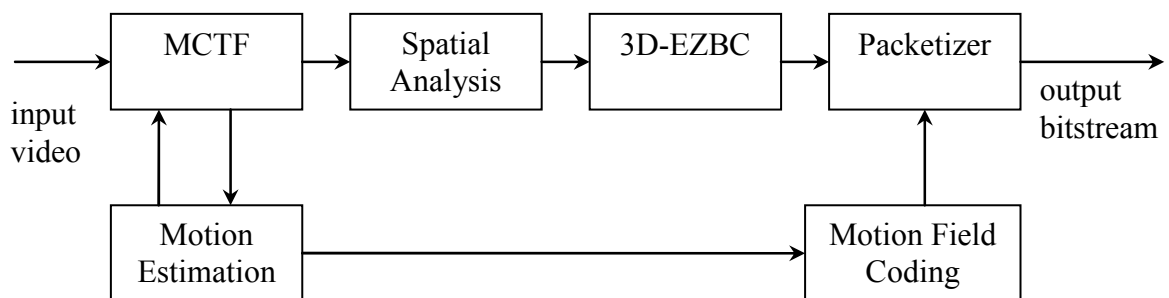


Fig. 3.3. Basic structure of MC-EZBC [Hsi01].

The coder exploits motion compensated temporal filtering (MCTF) and EZBC spatial coder. In this coder the MCTF is used to reduce the aliasing effect when the video sequence resolution in time domain is decreased. Moreover, this system does not

suffer from the drift problem which is presented in hybrid coders that have feedback loops.

Motion compensated temporal filtering is a very important part of motion compensated 3-D subband/wavelet coding. MCTF is used for subband/wavelet decomposition of video sequence in time domain. One of the ways of implementing this temporal filtering technique is using lifting scheme [Cal98] and another approach was proposed by J.-R. Ohm [Ohm94] and extended by Choi and Woods [Cho99].

3.3.3. Implementation of motion-compensated temporal filtering

The basic idea of motion compensated temporal filtering is to perform temporal filtering along the motion trajectory. But, there is a problem of how to deal with connected/unconnected pixels. There were two proposals to solve this problem. One was Ohm's method [Ohm94] where after motion compensation for the unconnected (see Fig. 3.4) pixels the original pixel values of current frame were taken as values for low temporal subband. And the scaled displace frame difference were taken as a value for high temporal subband. In [Cho96], Choi proposed that for unconnected pixels for the temporal low-subband the original value from the reference frame should be taken because unconnected pixels are more likely to be uncovered ones. The motion compensated filtering methods, both Ohm's and Choi's, are shown at Fig. 3.4.

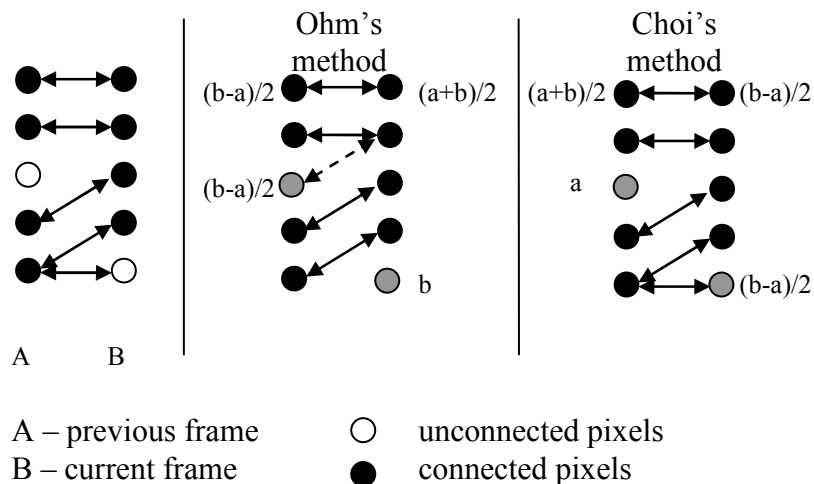


Fig. 3.4. Motion compensated filtering for connected/unconnected pixels.

Another approach of motion compensated temporal filtering is a lifting scheme which consists of three steps:

- polyphase decomposition,
- prediction step,
- update step.

Lifting implementation of analysis and synthesis filter banks is shown in Fig. 3.5.

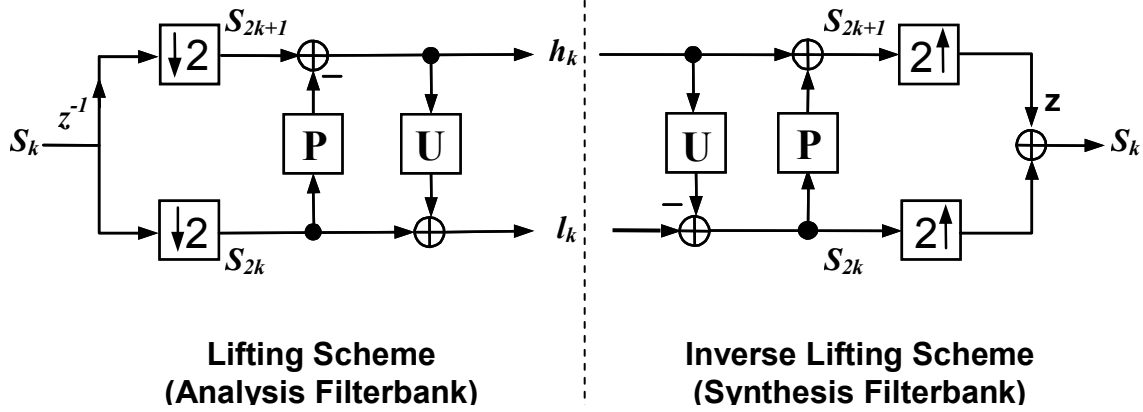


Fig. 3.5. Lifting representation of analysis and synthesis filter bank.

At the analysis side the input signal is divided into odd samples s_{2k+1} and even samples s_{2k} . The odd samples are predicted from even samples by the use of prediction operator $\mathbf{P}(s_{2k+1})$. The output h_k signal (high pass) is the prediction residuals. The l_k signal (low pass) is a sum of the signal obtained by the use of update operator $\mathbf{U}(s_{2k})$ on h_k signal and even samples s_{2k} . The p_i and u_i are prediction step and update step coefficients.

$$h_k = s_{2k+1} - \mathbf{P}(s_{2k+1}) \quad \text{with } \mathbf{P}(s_{2k+1}) = \sum_i p_i s_{2k+2i} \quad (3.1)$$

$$l_k = s_{2k} + \mathbf{U}(s_{2k}) \quad \text{with } \mathbf{U}(s_{2k}) = \sum_i u_i h_{k+i} \quad (3.2)$$

This is a perfect reconstruction filter bank because prediction and update steps are fully invertible. At the synthesis side the prediction and update operators are used in inverse order with the inversed signs of summation processes.

For uni-directional motion-compensated prediction the prediction and update step operators are as follows:

$$P_{uni}(s_{x,2k+1}) = s_{x+m_{p_0},2k-2r_{p_0}} \quad (3.3)$$

$$U_{uni}(s_{x,2k}) = \frac{1}{2} h_{x+m_{u_0},k+r_{u_0}} \quad (3.4)$$

where $s_{x,k}$ is a video signal with the spatial coordinate $x=(x,y)^T$ and the temporal coordinate k , m_{Pz} and m_{Uz} are motion vectors for prediction and update step from z frame ($z=1..2$), r_{Pz} and r_{Uz} are reference frames for prediction and update step.

For bi-directional motion-compensated prediction the operators are as follows:

$$P_{bi}(s_{x,2k+1}) = \frac{1}{2}(s_{x+m_{P0},2k-2r_{P0}} + s_{x+m_{P1},2k+2+2r_{P1}}), \quad (3.5)$$

$$U_{bi}(s_{x,2k}) = \frac{1}{4}(h_{x+m_{U0},k+r_{U0}} + h_{x+m_{U1},k-1-r_{U1}}). \quad (3.6)$$

The idea of bi-directional MCTF was introduced by Ohm [Ohm94]. The bi-directional motion-compensated prediction increases motion-vector rate in the bitstream but considerably decreases the energy of the prediction residuals.

In [ISO04a] the authors have proposed a tree-band motion-compensated filter bank. The proposed tool consists of triadic motion-compensated temporal filter bank with bi-directional predict and update operators.

3.3.4. Scalability in wavelet video coding

Wavelet coding technique decomposes input data into subbands, where the input may be as well still picture as video sequence. Each consecutive subband is encoded and may be sent as an enhancement layer to decoder. Decoder after decoding such a layer increases the quality of output picture or output video sequence, providing that way functionality of scalability. Wavelet coding takes advantage of facts that:

- signal energy after wavelet decomposition is accumulated near low frequencies,
- there is strong correlation between subband data.

The number of subbands may be increased and that way providing more layers enhancing quality of base layer as well as in spatial as in temporal domain.

If input data is three-dimensional, i.e. video sequence, various techniques of three-dimensional wavelet decomposition may be used. Depending on the technique various modification have been provided into wavelet coding such as MCTF, motion vectors coding, context-based arithmetic coding of residual data, etc. But all those modifications do not influence on basic concept of wavelet video coding.

3.4. Hybrid scalable coding

Scalable video coding for hybrid codecs with DCT-like transform was already provided and accepted for general use in MPEG-2 [ISO94] and later in MPEG-4 [ISO99]. But here, coding efficiency when scalability was used was not satisfactory. The structure used there consists of layers for each spatial resolution video output. Each layer may also be partitioned into sub-layers, each representing different output video quality at given spatial resolution. The structure is shown in Fig. 3.7.

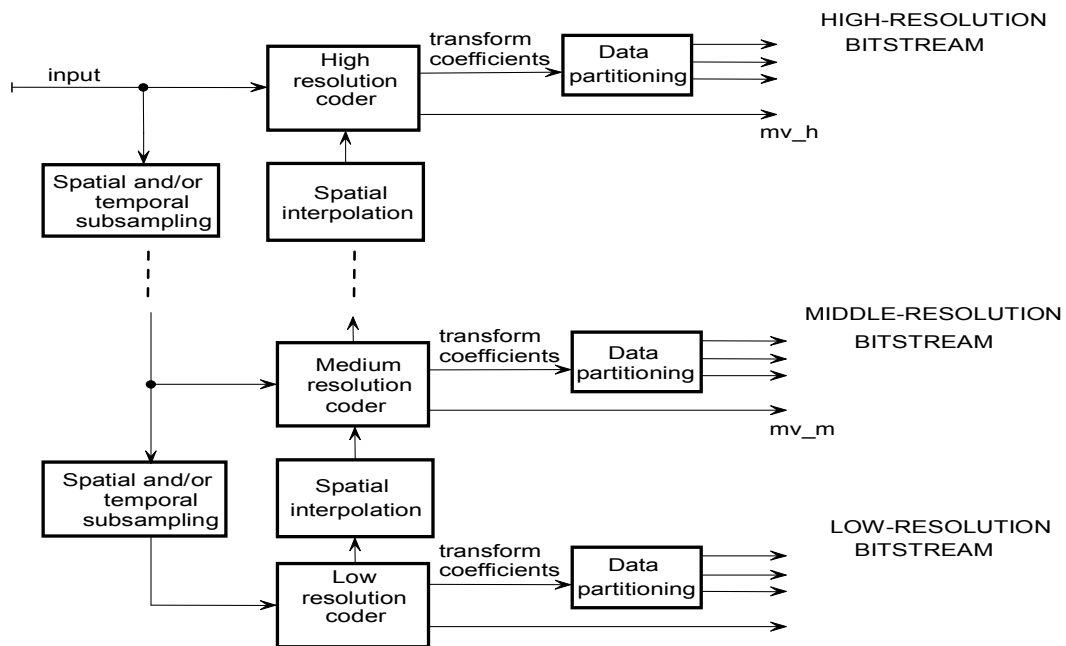


Fig. 3.7. Multi-layer scalable encoder structure.

This structure was also used in scalable techniques proposed in [Dom00a, Dom01, Mac02, Dom02e, Dom02f, and ISO04c].

3.4.1. Temporal scalability

For such a structure the number of layers of reduced spatial resolution and number of layers of reduced temporal resolution is determined at the encoder side. In the case of hybrid codecs there may be distinguished two methods for obtaining temporal scalability.

The first one which is used for most of hybrid codecs divides encoded frames into the following types:

- frame which can be used as reference for other frames,

- frame which cannot be used as reference for other frames.

If a frame cannot be a reference frame it can be dropped in the communication channel if there is not enough bandwidth to deliver the bitstream to the decoder or at the decoder side if such a decoder has no computational power to decode this frame. Both types of frames may be encoded: as access point frames, it is a frame which is encoded only by the use of spatial prediction modes (this frame does not use other frames for prediction); as a frame using one directional temporal prediction as well as two-directional temporal prediction (those frames use other frames for prediction). This way of achieving temporal scalability was used in [ISO94, ISO99, ISO-AVC, Dom03, Bla03c, and Bla05d].

Another technique used to achieve temporal scalability is the technique which uses motion compensated temporal filtering that has already been described in Section 3.3. Later in [ISO04c], Schwarz, Marpe and Wiegand proposed a scalable extension of H.264/AVC where the MCTF was applied. The connection of MCTF, taken from wavelet technology, and highly efficient hybrid codec results in a very efficient and promising scalable hybrid encoder which became a base for developing new advanced scalable encoder. An example of temporal decomposition made by use of MCTF is shown at Fig. 3.8.

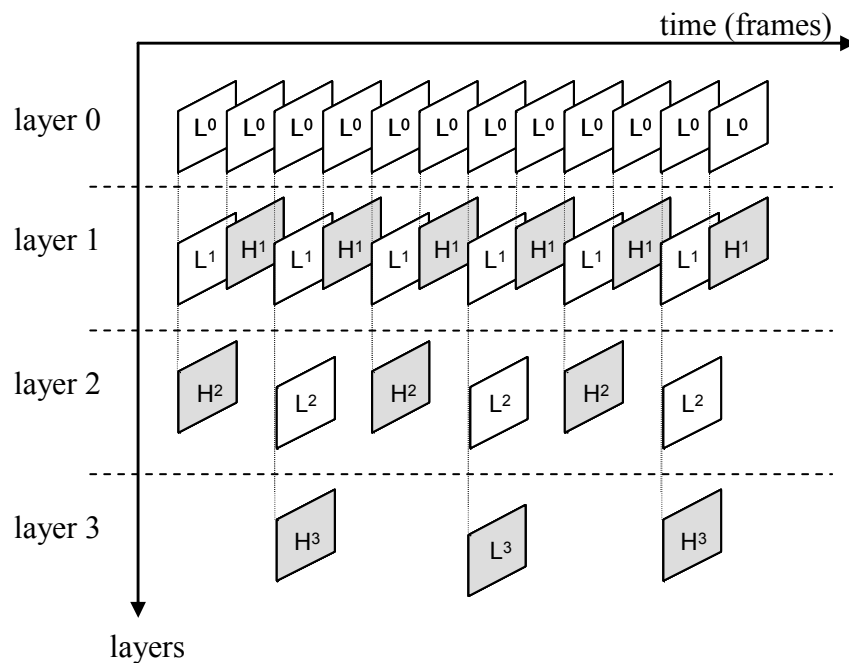


Fig. 3.8. Temporal decomposition of a group of 12 pictures providing 3 levels of temporal scalability with temporal resolution ratios of 1/2, 1/4, and 1/12.

3.4.2. Spatial scalability

Spatial scalability is obtained by encoding each spatial resolution layer separately, by the use of motion compensated encoder which can even use different coding techniques. The data from lower layers are up-sampled and used in prediction loop of higher layer encoders. Multilayer scalability was already presented in [ISO94, ISO99, Hor99, Dom00a – Dom00d, Mac02], also the author has analyzed several techniques of scalable encoding based on layered structure in [Dom02a – Dom02f, Bla03a, Dom03, Bla03a – Bla03e, Bla04a, Bla04c, Dom04a, Dom04b]. Coding efficiency of codecs with spatial scalability depends on decimation and interpolation process as it was described in [Dom03, Bla03e, Lan04] and efficient exploitation of reference and interpolated pictures as described in [Dom02e, Bla03b, Dom03].

For spatial scalability in scalable AVC as presented in [ISO04c] spatial scalability concept which was already introduced in video coding standards H.262/MPEG-2 Visual, H.263 and MPEG-4 Visual was adapted. The base layer with reduced spatial resolution is encoded as a low pass signal. Then the reconstructed pictures L_0 are spatially up-sampled. Those up-sampled pictures which are the same spatial resolution as an enhancement layer can be used as prediction for macroblock in the enhancement layer. Fig. 3.9. presents concepts of spatial scalability.

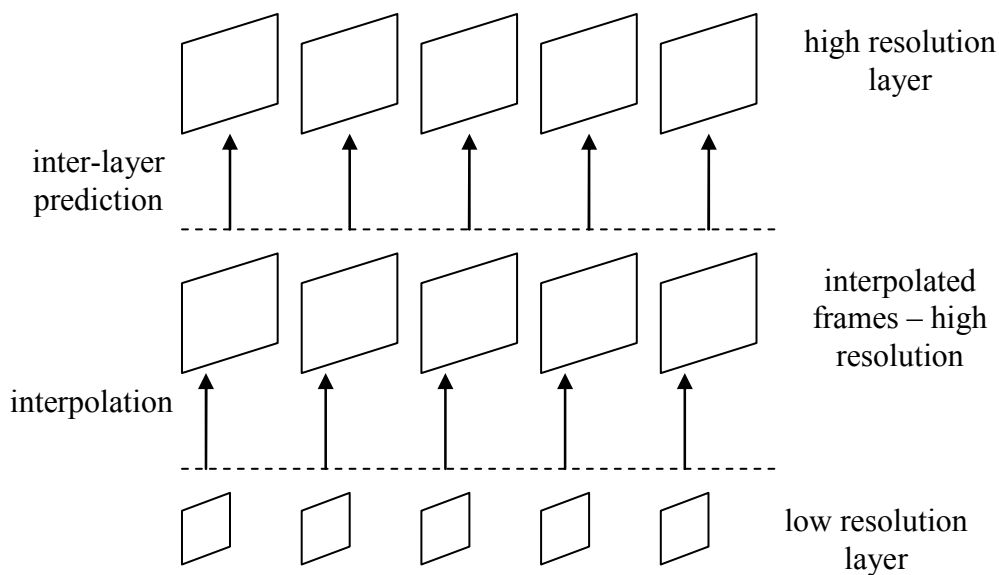


Fig. 3.9. Implementation of spatial scalability.

Each spatial resolution layer is encoded by the use of separate motion estimation and compensation process.

3.4.3. Quality scalability

The third type of scalability used in hybrid coding is quality scalability. It may be obtained by several techniques. One is a multilayer SNR scalability technique. There are two layers at the same frame rate and the same spatial resolution, but using different quantization parameters. Fig. 3.10. shows the decoder structure with SNR scalability defined in MPEG-2 [ISO94]. The enhancement layer consists of variable-length-coded DCT coefficients of residuals. Here, the enhancement layer is used in the motion-prediction loop. If the enhancement layer is not received by decoder, then drift happens and coding efficiency may be low.

This technique may be extended to multiple layers. Each layer has the same spatial resolution, but the quantization parameters differ. The number of layers depends on the quantization parameter step between corresponding layers. Moreover, each layer may use its own motion compensated prediction. Such quality scalability for advanced video coding hybrid technique was proposed in [Sch03, Sch04]. The structure of coder with this functionality is shown in Fig. 3.11.

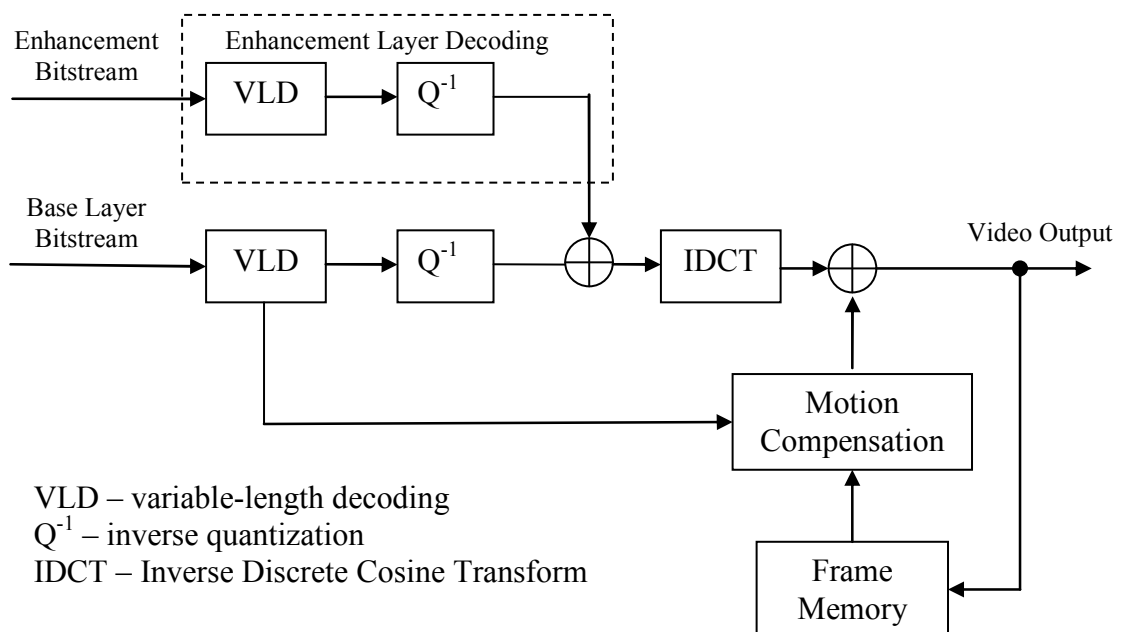


Fig. 3.10. Decoder structure with SNR scalability defined in MPEG-2.

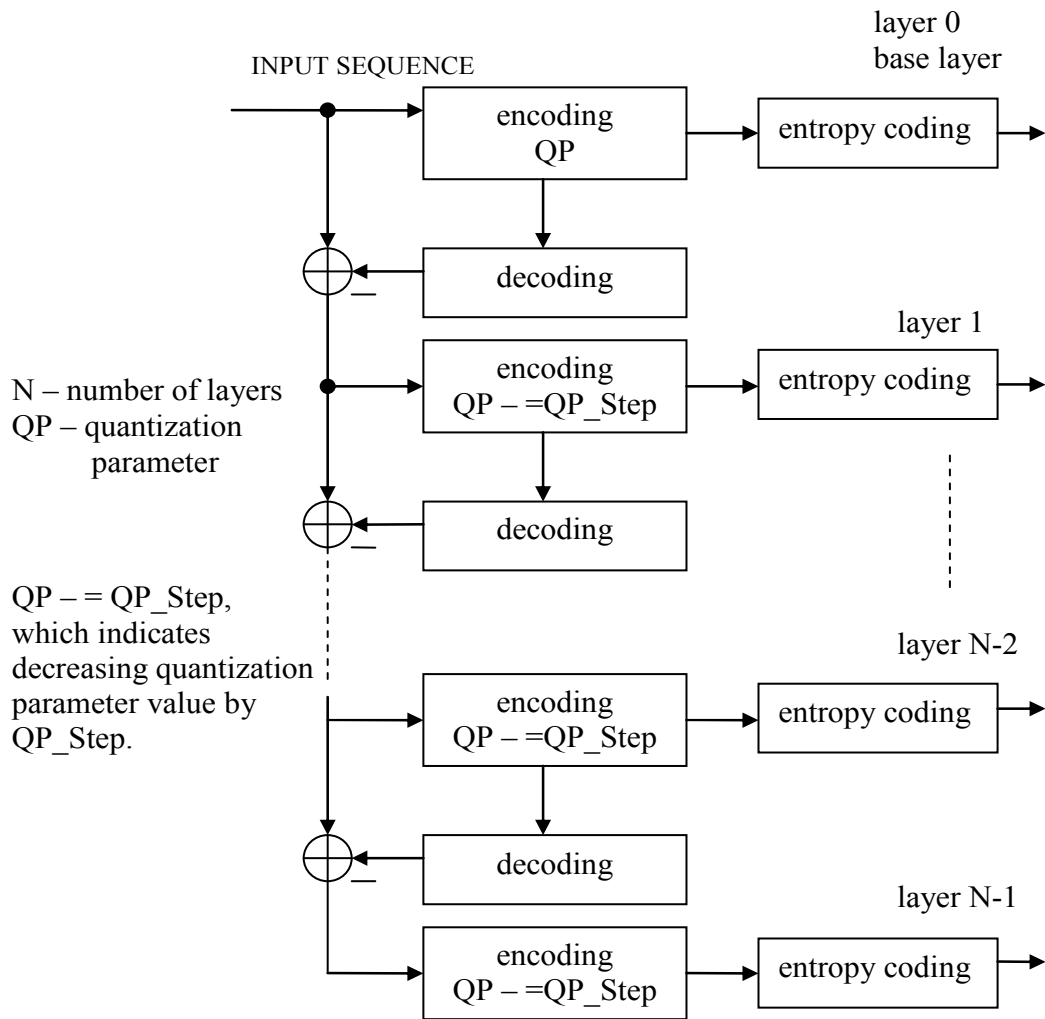


Fig. 3.11. Encoder structure, with quality scalability, based on multiple layers.

SNR scalability may also be obtained by a technique of bit-plane coding which is an extension for standard coding of transform coefficients at the encoder side. The conventional method treats transform coefficients as two-dimensional matrix of integer values, while the bitplane coding technique treats these coefficients as several two dimensional matrixes. Each matrix is composed of one-bit values. These one bit values are consecutive bits of binary representation of each coefficient. Thus, for example, for 8×8 DCT block a bit-plane of the block is defined as an array of 64 bits, taken one from each absolute value of DCT coefficient at the same significant position [Li01]. Bit-plane coding and matching pursuit coding of image residue [Ben98, Che98, Li98, Schu98] are techniques for obtaining Fine Grain Scalability (FGS), which is used for producing the bitstream which may be cut, at the decoder side, at almost any position. Those techniques have been developed for years. There were several improvements proposed such as hybrid temporal-SNR FGS proposed in [Sch01], where the temporal scalable

layer and SNR scalable layer are considered to be one scalable layer; macroblock-based progressive fine granularity scalable coding (PFGS) as described in [Wu00, Wu01, Sun04], where the high quality reference frames are used in enhancement layer coding (see Fig. 3.12), which provides higher coding efficiency, but introduces also higher possible drifting errors at the decoder side; adaptive motion-compensation fine granularity scalability (AMC-FGS) where the adaptive switching between two-loop motion compensated FGS and single-loop motion compensated FGS is done [Sch02] in order to achieve optimal streaming performance over the network. The system chooses the most suitable FGS structure based on the bandwidth variations or device capabilities; quality scalability based on bit-plane coding of matching pursuit atoms as described in [Lin05]. The FGS technique based on bit-plane coding were also developed and analyzed by Maćkowiak and Domański in [Mac02, Dom01] and also in [Dom02c, Dom02d, Bla03e].

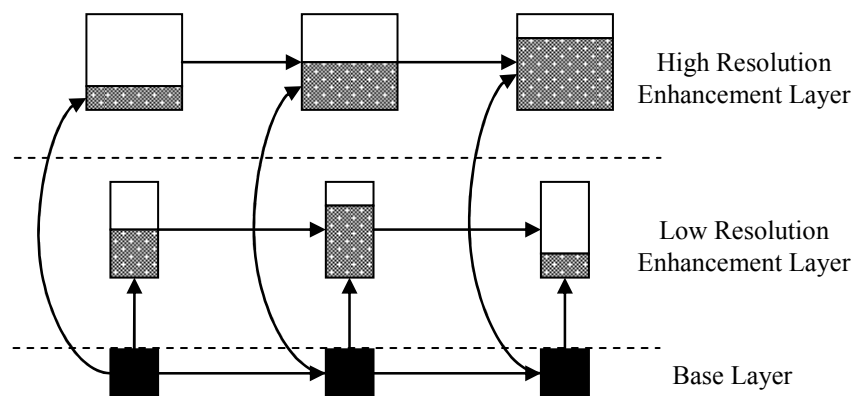


Fig. 3.12. PFGS coding scheme.

FGS scalability can be also achieved by macroblock reordering. The reordering is done in order to encode into a bitstream first the most important macroblocks and later less important ones. The most important macroblocks represent part of the image which is subjectively more important for human observer. So, this technique of FGS tries not to lose, after the bitstream cut, subjectively the most important parts of the encoded picture. This technique was proposed in [Par02, Bla04d, and Bla04e]. This technique is described later in details in this doctoral dissertation.

Chapter 4

Multilayer Advanced Video Coding

4.1. Introduction

This chapter deals with the problem of adopting and implementing the multilayer scalable video coding to advanced video technology. A generic structure of a multi-loop scalable encoder shown in Fig. 4.1 was also used in scalable MPEG-2 by S. Maćkowiak in his doctoral dissertation [Mac02]. Here, in this work, the author considers the ability of modifying and adopting this structure for use in the newly developed advanced video coding techniques. The multi-layer scalable advanced video codec has been described in previous Chapter in Section 3.4 and in [Dom02b, Dom02e, Dom02f, Dom03, Bla03a, Bla03c, Bla03d, Bla03e, Bla04a, Bla04c, and Lan04]. Also SNR scalability has been added by the author, into advanced video coding technology in [Dom02c, Dom02d, Bla04d, Bla04e, Bla05a, Bla05d, and Bla05e].

As proposed, a scalable coder may, in general, consist of several sub-coders, at least two. Each of sub-coders has its own prediction loop with its own motion estimation and compensation. As it was proved in [Mać02], the bitrate needed for additional data correlated with multiple motion estimation are well compensated by the decrease in the number of bits needed for prediction error coding [Dom00, Łuc00]. Thus, such a structure may be well adapted to advanced video coding as well as earlier techniques of video coding.

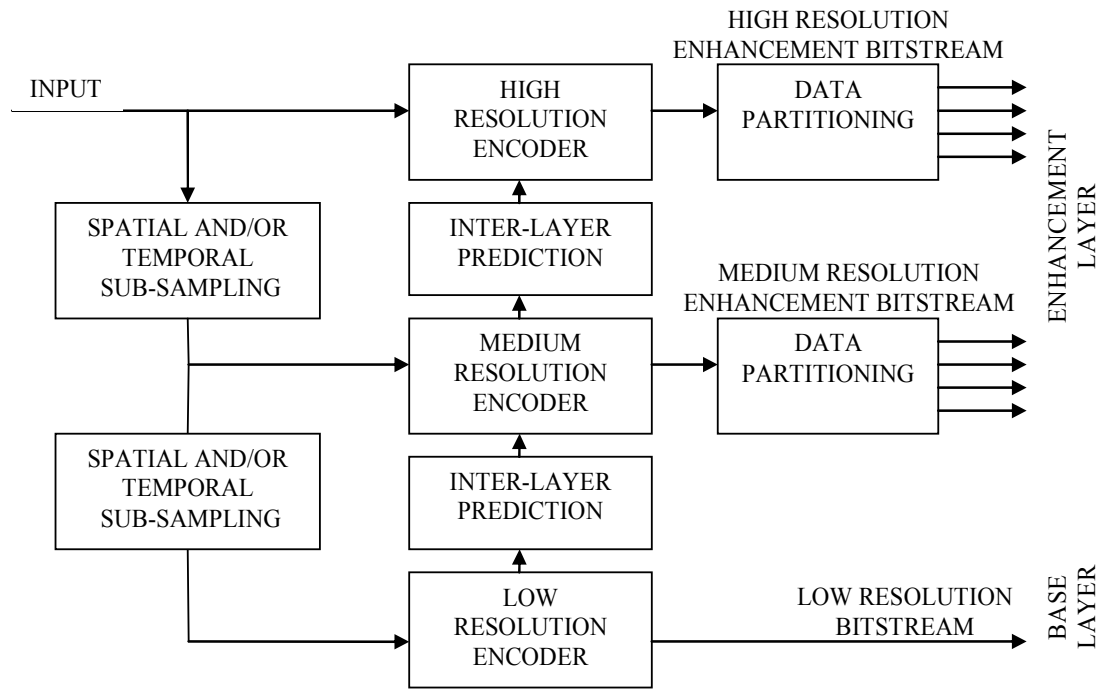


Fig. 4.1. A generic block diagram of the multi-loop scalable encoder.

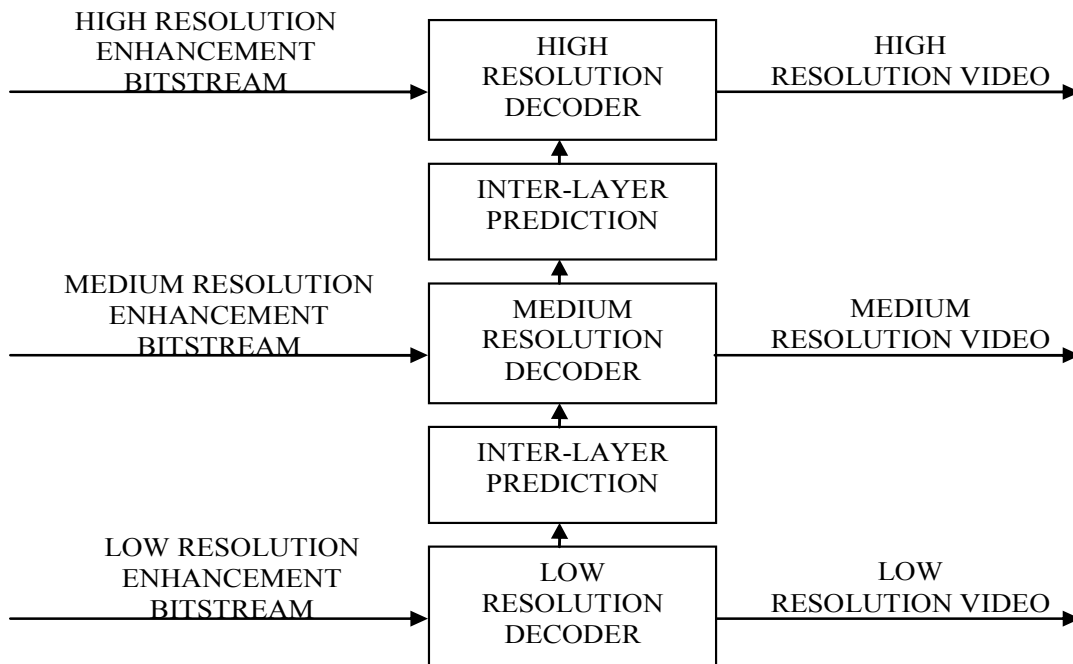


Fig. 4.2. A generic block diagram of the multi-loop scalable decoder.

This general structure represents the scalable video codec which may take advantage of three types of scalability: spatial, temporal and quality scalability. Each sub-encoder may encode video signal of different spatial and temporal resolution.

Moreover, each layer of the structure, beside the lower one, may partition the encoded data in order to obtain quality scalability. The lowest layer does not partition data because of backward compatibility with non-scalable base codec.

Here, the author shows that such a structure may be successfully used for designing a model of scalable codec based on earlier video coding techniques such as H.263 [ISO96], as well as for designing model of the encoder based on the advanced video coding techniques such as H.264/AVC [ISO-AVC]. Multilayer structure of codec may provide backward compatibility with non-scalable codecs. The lowest layer may be treated as non-scalable base layer and may have the same syntax as non-scalable. Here, in this doctoral dissertation the author presents both models of scalable encoders.

4.2. Spatial Scalability

Spatial scalability is obtained by encoding sub-bitstreams for each spatial resolution separately by use of additional information from decoded lower layer. For the spatial scalability the enhancement layer encoder is an extended version of base layer encoder. In the enhancement layers there are the following types of frame: *IE-frames*, where spatial prediction and prediction from lower layer are allowed, *PE-frames*, where spatial, temporal (one direction) and from lower layer predictions are allowed and *BR-frames*, where spatial, temporal (two directions) and from lower layer predictions are allowed. All three types of frames may be used as a reference for temporal prediction. The common feature of all these types of frames is that they use in prediction process an interpolated frame, at the time of displaying, from lower layer (it may be another enhancement layer). The author's idea was to take advantage of one of advanced video coding tools which is the multi-reference prediction. The codec can use more than one reference picture in prediction process. The idea was to take an interpolated frame from lower layer and put it as additional frame into the list of reference frames. In this way, the syntax of bitstream remains unchanged, while it still enables enhancement layer encoder to better encoding efficiency. It has to be noticed here that the currently encoded frame and the interpolated frame represent the picture at the same given time of displaying. So, it can be assumed that the best prediction of currently encoded

macroblock by the use of interpolated frame is in the same spatial coordinates. In consequence, there is no need to send motion information to the decoder. Thus, for this mode of prediction the motion vectors are not transmitted.

IE-frames in the advanced video coding technique may use spatial prediction from the same frame or from lower layer frame. *PE-frames* and *BR-frames* may additionally use a reference frame which comes into being by averaging interpolated lower layer frame and reference frame from the same layer. It is so called averaging mode. In [Mać02] the proposed averaging mode took the best temporal prediction and then averaged it with interpolated block from lower layer. The author of this doctoral dissertation has adopted this method to advanced video coding technique. This method has been implemented and included into AVC. It had to be modified to act properly with multiple prediction modes and multiple reference frames which exist in this new advanced technology.

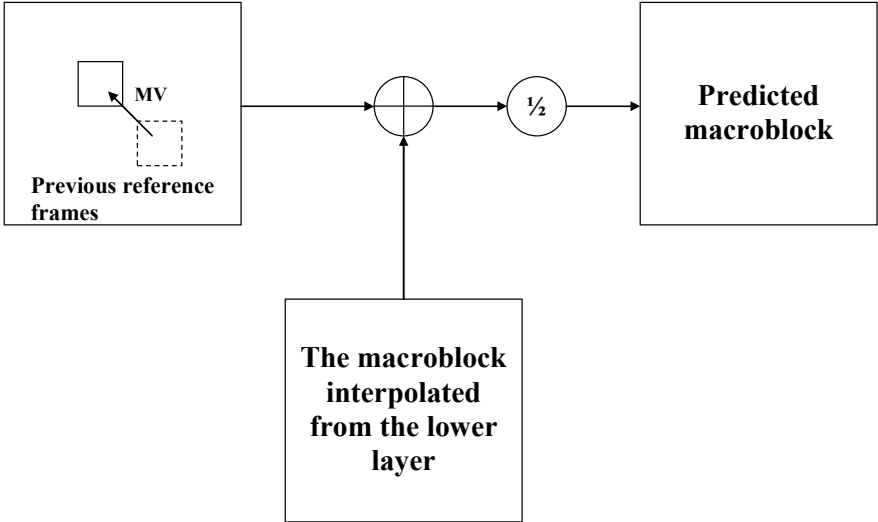


Fig. 4.5. Averaging mode from [Mać02]

Here, the author’s proposal is an extension of the averaging mode proposed by Maćkowiak. The idea is to find the block from previous reference frames which after counting the average with the interpolated block from lower layer gives the best estimation of currently coded block. Thus, the proposal in [Mać02] becomes the special case of new technique. The difference is that the new technique looks for the average which gives the best prediction and previous method looks for the best prediction from previous frame and average predicted block with interpolated block. The following figure shows the idea:

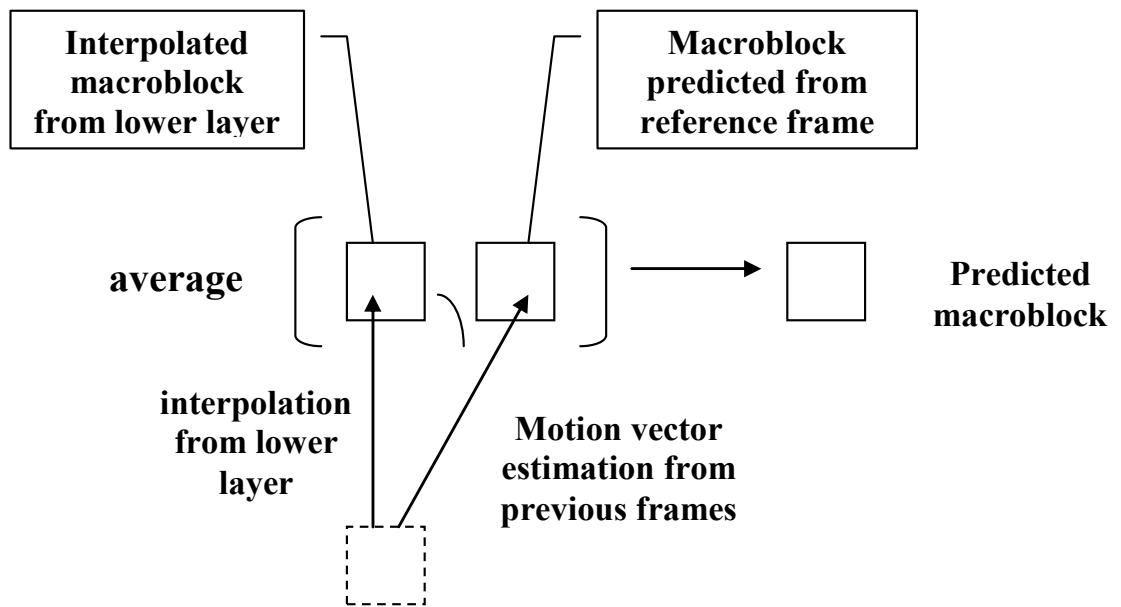


Fig. 4.6. New averaging mode proposed by the author.

This new averaging mode is modification of earlier algorithm. Modification is made in order to minimize the cost of providing additional coding mode in bitstream syntax for scalable advanced video codec. This algorithm is computationally more complex than the previous method because for each block matching in motion search algorithm the averaging with interpolated block has to be done. Thus, for block match there are additional summation and shifting for each pixel value.

IE-frames proposed in this work differ from the frame type proposed in [Mać02]. Here, this frame has a structure similar to the *P-frame*. Thus, all intra modes (spatial predictions) are allowed and one prediction mode (block size 16×16) which is based on temporal prediction coding structure, but no motion vectors are used. The reference frame list contains only one frame, i.e. interpolated frame from lower layer.

4.3. Temporal scalability scenarios for scalable codec

Temporal scalability discussed here is widely used in hybrid DCT-based codecs. The idea was not to use some video frames for predictive coding of other frames. Such frames may not be decoded at the decoder side and the remaining frames are still decodable. Moreover, those frames could be encoded by the use of prediction from

neighboring frames from two directions: forward and backward. When such frames are used, the coding efficiency increases. In considered layered structure all frames may also use a spatial prediction, and some of bi-directionally predicted frames may also use other bi-directionally predicted frames. Some of possible scenarios for two layers structure, already presented in [Mac02 pp. 102-103] are shown in Fig. 4.3 and Fig. 4.4.

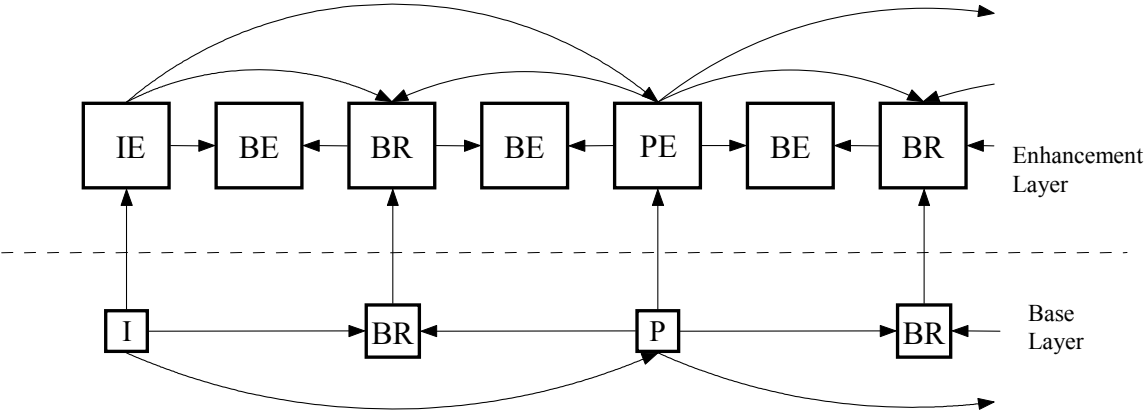


Fig. 4.3. Exemplary structures of low-resolution and high-resolution video sequences with temporal sub-sampling by factor 2.

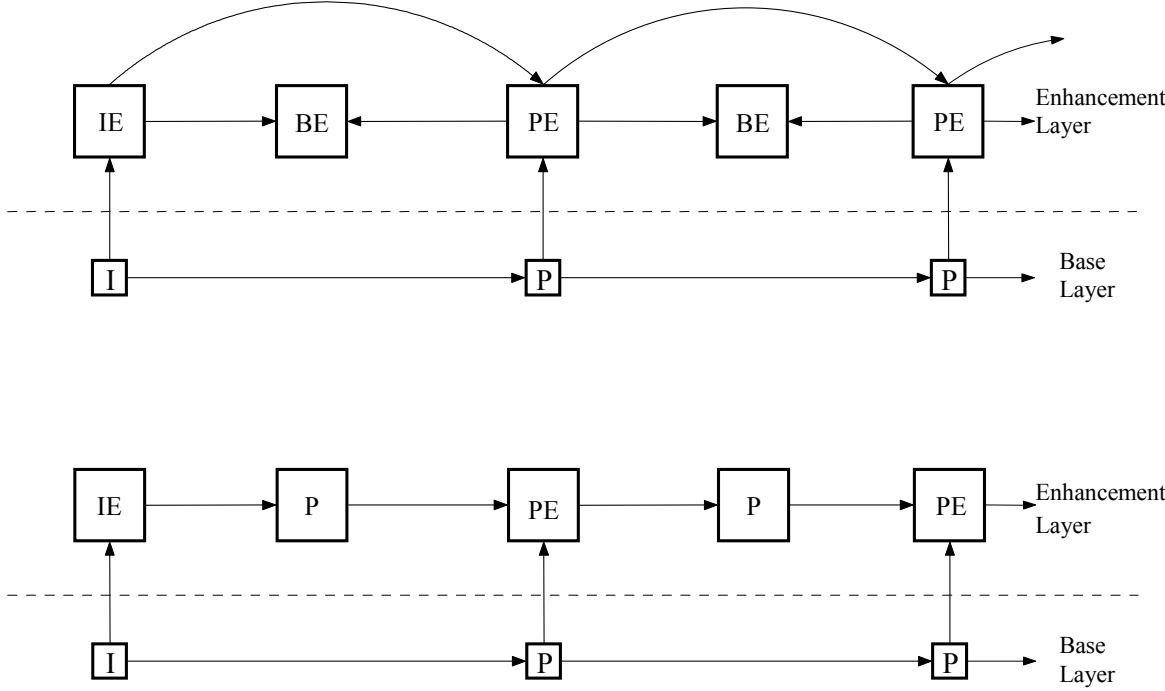


Fig. 4.3 cont. Exemplary structures of low-resolution and high-resolution video sequences with temporal sub-sampling by factor 2.

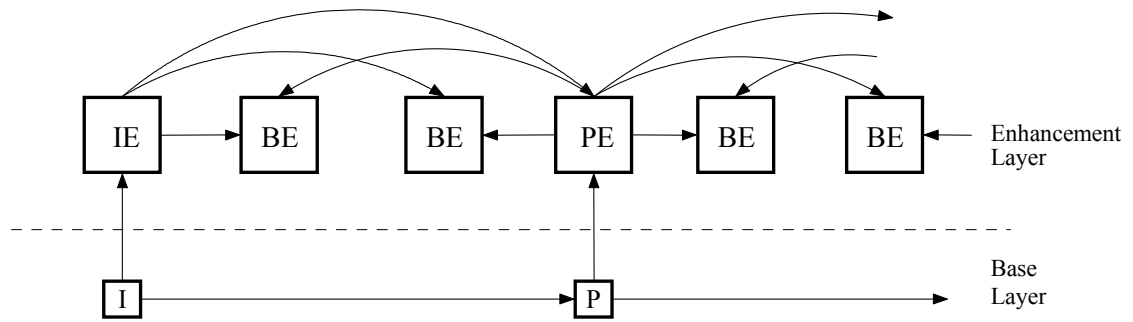


Fig. 4.4. Exemplary structure of low-resolution and high-resolution video sequences with temporal sub-sampling by factor 3.

There are two types of bi-directionally predicted (*B*) frames: *BR-frames* and *BE-frames*. The first one is a frame which may also use frame from lower layer for prediction and it may be a reference frame for other *B* frames, but cannot be used as reference for *P* frames. *BE* frames are bi-directionally predicted frames which may use as reference frame *P* or *BR-frames*. *BR-frames* and *BE-frames* belong to two different categories described in Chapter 3, i.e. frames which can or cannot be reference for other frames (see Chapter 3 Section 3.4.1).

4.4. Interpolation and Decimation

The input video signal for the scalable video coder has to be spatially and/or temporally decimated. The decimation process has to be performed as many times as many enhancement layer encoders the scalable coder consists of. Before the video signal is down-sampled it has to be filtered by the use of a low pass filter. The filtration has to be done in order to avoid the aliasing effect. In the case of scalable coding, the low pass filtration may also be used as a tool for distribution of input signal energy between encoded spatial layers. For the opposite process the up-sampling and estimation of missed pixels is done. Interpolated picture is used for prediction of the enhancement layer picture.

For verification model of scalable multilayer advanced video encoder the decimation filter of following design conditions has been used:

- passband attenuation below 1 dB,
- passband cutoff frequency of about 0.4 of the Nyquist frequency,
- stopband attenuation over 50 dB.

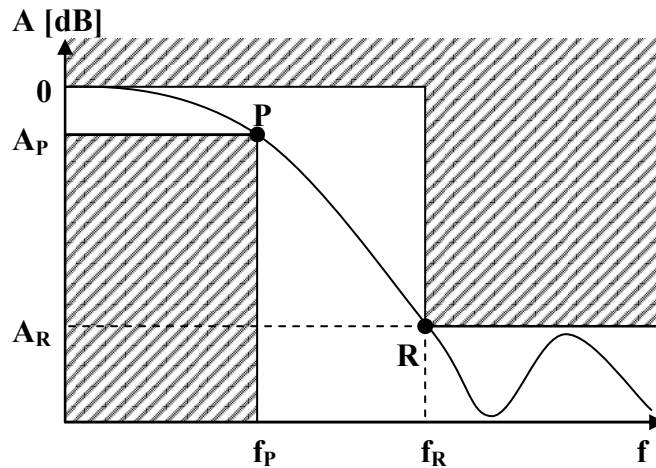


Fig. 4.7. Decimation filter design.

Two filters have been designed by the use of the least-square error technique in Matlab environment, which granted the above conditions. One 12th order FIR filter and one 24th order FIR filter have been designed. The Figs. 4.8 and 4.9 show the magnitude responses for both filters. The filter coefficients are the following:

- $h(n) = [0.015259, -0.009986, -0.066826, -0.062964, 0.083263, 0.303814, 0.411660, 0.303814, 0.083263, -0.062964, -0.066826, -0.009986, 0.015259]$;
- $h(n) = [0.005622, 0.016861, 0.013501, -0.012365, -0.028729, -0.000163, 0.043615, 0.026059, -0.056480, -0.081174, 0.065250, 0.309576, 0.431635, 0.309576, 0.065250, -0.081174, -0.056480, 0.026059, 0.043615, -0.000163, -0.028729, -0.012365, 0.013501, 0.016861, 0.005622]$.

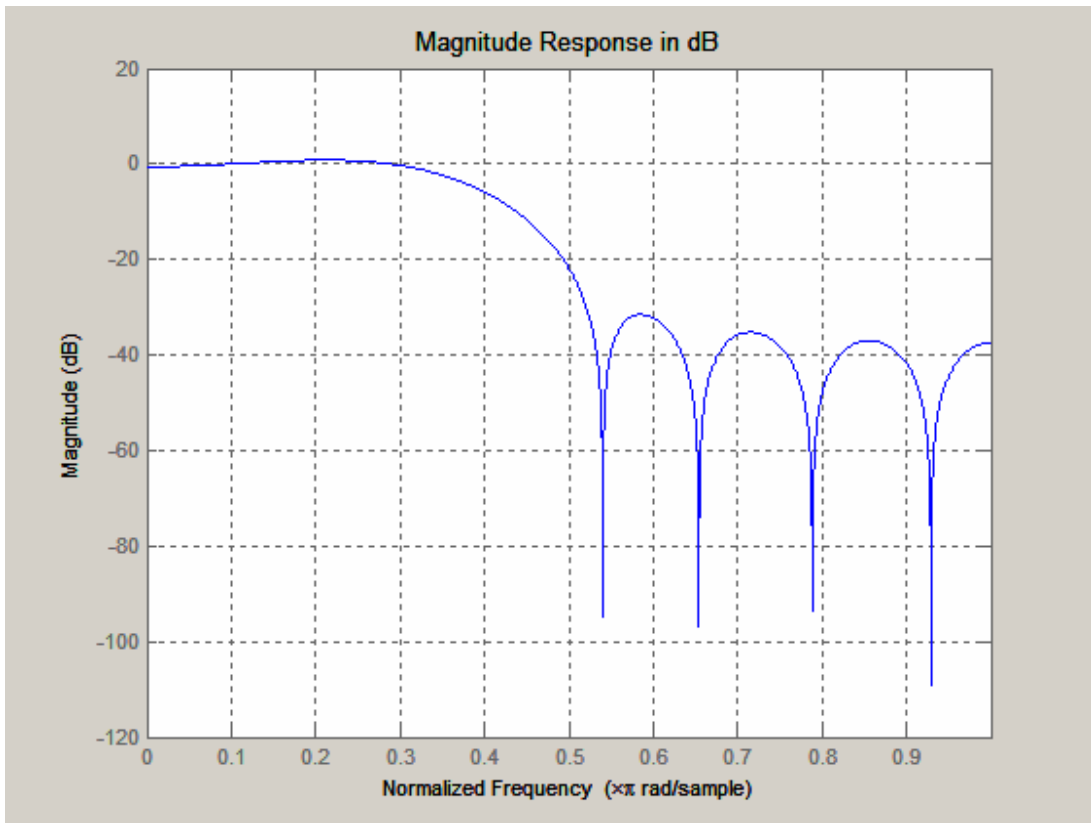


Fig. 4.8. Magnitude response of 12th order filter.

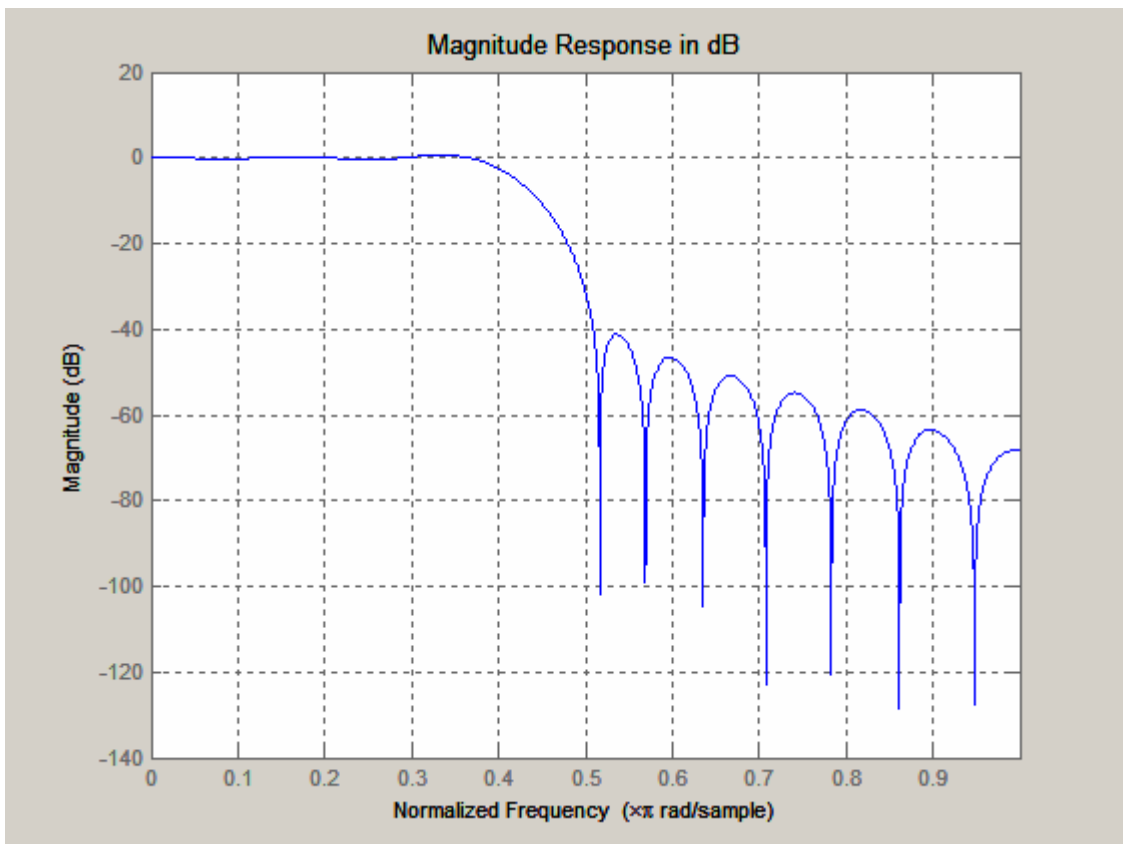


Fig. 4.9. Magnitude response of 24th order filter.

The reason for designing two filters of the same type but different orders was to determine which feature is more important in decimate filter for scalable coding (see Fig. 4.1): Whether it is the better magnitude response of the filter, but longer distortions around the edges or not so good a magnitude response, but shorter distortions around the edges. The experiment described in Chapter 6.4 shows that there is a difference in coding efficiency for both of the decimation filters. The results related to those experiments are in the Table 4.1 below.

Table 4.1 Influence of designed decimation filters on the encoding efficiency.

test sequence	short decimation filter		long decimation filter		bitrate overhead [%]
	PSNR [dB]	bitrate [kbps]	PSNR [dB]	bitrate [kbps]	
city (± 64 search range)	36,49	3144,96	36,49	3164,10	0,61
city (± 4 search range)	34,31	1411,80	34,31	1427,35	1,11
crew (± 64 search range)	38,82	1978,62	38,82	1991,47	0,65
crew (± 4 search range)	37,25	1047,72	37,25	1058,37	1,02
harbour (± 64 search range)	35,82	5356,81	35,82	5388,98	0,60
harbour (± 4 search range)	34,57	2861,45	34,57	2883,78	0,78
ice (± 64 search range)	40,71	1387,94	40,72	1403,76	1,14
ice (± 4 search range)	38,90	935,49	38,91	949,45	1,50

The conclusion of analysis of these two designed filters is that the coding efficiency is higher for designed decimation filter with lower order. The bitrate overhead for designed higher order filter is between 0.6% up to 1.5%.

For the scalable H.264/AVC verification model proposed by the author the modified bi-cubic interpolation has been chosen, because of its considerably good tradeoff between computational cost and accuracy. The technique has been taken from [Ram99] paper. It is an edge-adaptive bi-cubic interpolation and it is an extension to the standard non-adaptive bi-cubic separable interpolation. The author has proposed this technique in [Bla03e, Dom03, and Lan04].

The algorithm can be described as follows. The interpolation of a two-dimensional image is performed in two steps: first horizontal interpolation, second vertical interpolation. Let $f(x)$ be the value to be estimated and the nearest available values are located at coordinates x_k (left) and x_{k+1} (right). Let $s = x - x_k$, $1 - s = x_{k+1} - x$, where $0 \leq s \leq 1$. By bi-cubic separable interpolation, there is

$$f(x) = f(x_{k-1})(-s^3 + 2s^2 - s)/2 + f(x_k)(3s^3 - 5s^2 + 2)/2 + f(x_{k+1})(-3s^3 + 4s^2 + s)/2 + f(x_{k+2})(s^3 - s^2)/2,$$

where x_{k-1} , x_k , x_{k+1} and x_{k+2} are the positions of four neighboring known pixels.

In the edge-adaptive scheme, a modified value s' is used instead of s .

$$s' = s - kAs(s - 1),$$

where k is a positive parameter that controls the intensity of warping and A is a function of asymmetry of the data in the neighborhood of x :

$$A = (|f(x_{k+1}) - f(x_{k-1})| - |f(x_{k+2}) - f(x_k)|) / (L - 1),$$

where $l = 256$ for 8-bit sample representation. In order to obtain value k several experiments have been performed.

Figs. 4.10 and 4.11 show comparison of coding efficiency when adaptive and non-adaptive interpolation filter are used. The detailed description of the experiments and its results are presented in Chapter 6.4.

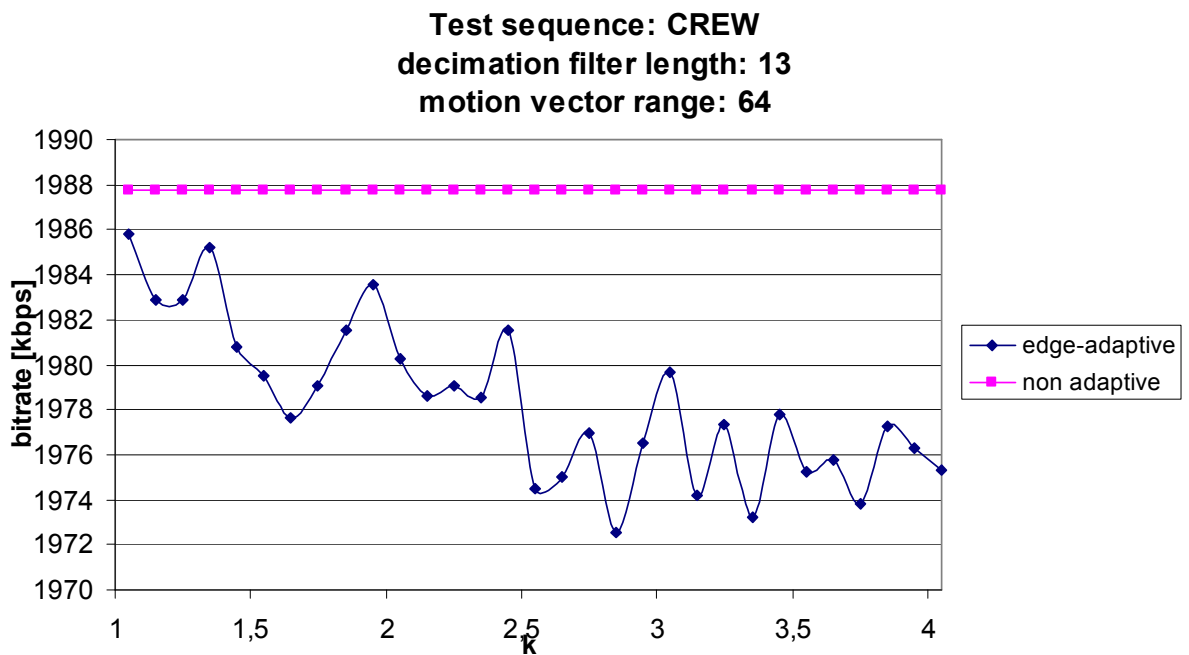


Fig. 4.10. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filters for CREW test sequence.

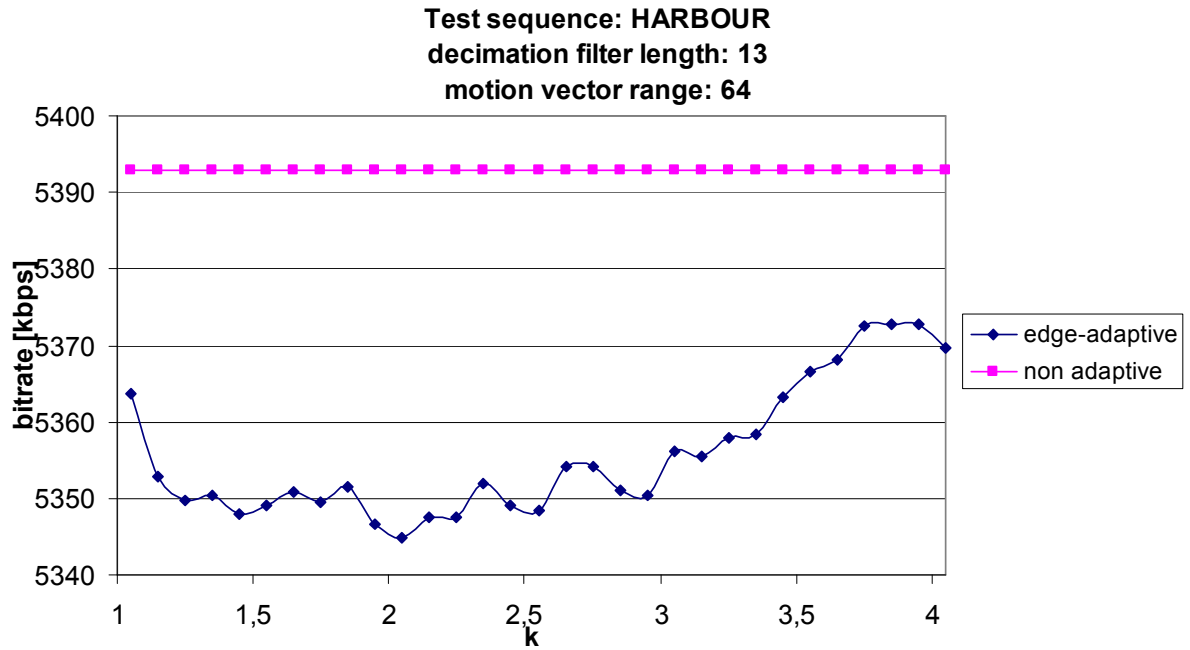


Fig. 4.11. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filters for HARBOUR test sequence.

As it may be noticed on above Figures the adaptation of interpolation increases encoding efficiency. Moreover, this efficiency strongly depends on the value of parameter k . The gain of coding efficiency, when adaptation is used, may be even up to 1%.

Chapter 5

Spiral scan

5.1. Introduction

Since the beginning of the digital image filtering and compression, people have been used to taking the upper left corner of the picture as a starting point for image processing. The image then is processed row by row, from the left to the right and from the top to the bottom. It is called raster scan. This processing order stems from human habit and does not have any reasonable explanation in the signal processing theory. If the filtering process of the image starts from lower right corner, in the opposite direction to the one described above, the results will be the same as before.

In the case of the video sequence compression, individual frames are divided into blocks of 16×16 pixels called macroblocks. The images are processed macroblock by macroblock. The processing order of those macroblocks is the same as it was for filtering process from the left to the right and from the top to the bottom. This coding order is widely used in many techniques of video compression and in all video coding standards including MPEG-2, MPEG-4 [ISO94, ISO98] and in the newest standard AVC/H.264 [ISO-AVC] as well.

Here a question appears: Is it possible to use other orders of macroblocks' processing which could be useful in video compression? And if so, what are its implications and applications?

The DCT-based video coding technique partitions the image into macroblocks and then encodes them one by one. The commonly used codecs like MPEG-1,2,4 or H.263

use the raster scan of macroblocks. But there is no reason why other scans are not used. The question is how such the order influences the encoding process, which eventually leads to the question about coding efficiency. The idea is:

- to find a new macroblock coding order which will be useful for some purposes,
- to modify coding process, for this macroblock coding order, in such a way that the coding efficiency remains unchanged.

The consequences of coding order modification strongly depend on the coding algorithm. In the new coding standard AVC/H.264 the dependency between neighboring macroblocks is very strong, when data are encoded. So, changing the macroblock coding order strongly influences coding efficiency. The MPEG-2, where the dependency between the neighboring macroblocks is not so strong, the macroblock coding order does not influence so strongly coding efficiency.

5.2. *Spiral scan in video compression*

It is important to define some specific features of encoded video shots. In general, it is possible to divide the video shots into two main groups:

- The video shot containing one or a few objects placed in some background;
- The video shot containing only the background without explicit objects.

An object can be defined as part of a video frame which, from the human point of view, has to be recognizable. The object may be a human being, an animal, a building, etc. Moreover, those objects have to be major regions of interest on the image for the human viewer. The background can be defined as part of image containing no recognizable objects or containing so many objects that it is impossible to decide which ones of them are the most important.

An example of a video frame containing one object and background can be the video sequence from a news channel where the news presenter is speaking. In such a case the news presenter is the object and the rest of the image is the background (see Fig. 5.1).

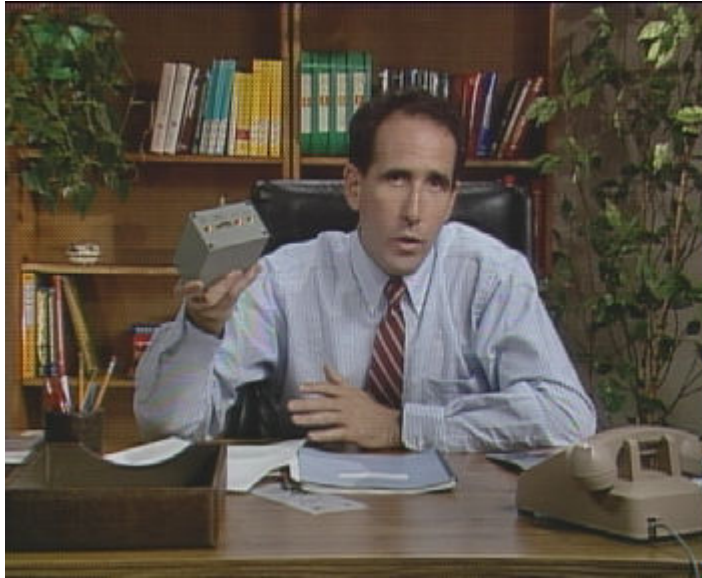


Fig. 5.1. The object of interest placed in the center of the image.

A video frame may consist of more than one object, for example there can be a video where two people are talking (see Fig. 5.2).



Fig. 5.2. Two objects of interest inside one image.

Video with no objects (only the background) can be, for example, the shots of a forest, mountains, etc. (see Fig. 5.3).



Fig. 5.3. Image containing no objects of interest.

The psycho-visual effect of human perception is focusing on a part of the image instead of the whole image. In most cases, a cameraman shoots video in such a way that the most interesting object is in the center of the picture. If there is more than one object they are placed around the center. A person focuses on one of those objects at a time.

The idea is to encode a video sequence in such a way that the most important parts of the images, here, these are the objects, are processed first and the less important part is processed later. As a solution of this idea, the spiral scan of encoded macroblocks is proposed. The classic scan and spiral scan are shown in Fig. 5.4.

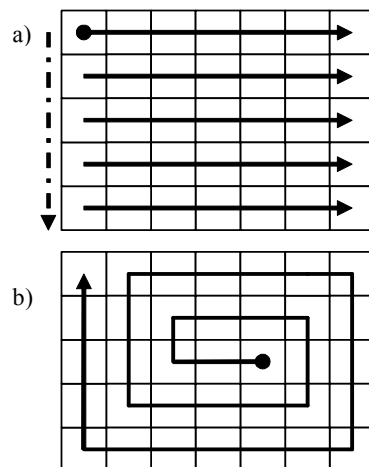


Fig. 5.4. a) Raster scan of macroblocks, b) Spiral scan of macroblocks.

Similar solution has been proposed in [Par02], called water ring scan order. The Fig. 5.5 shows the basic idea.

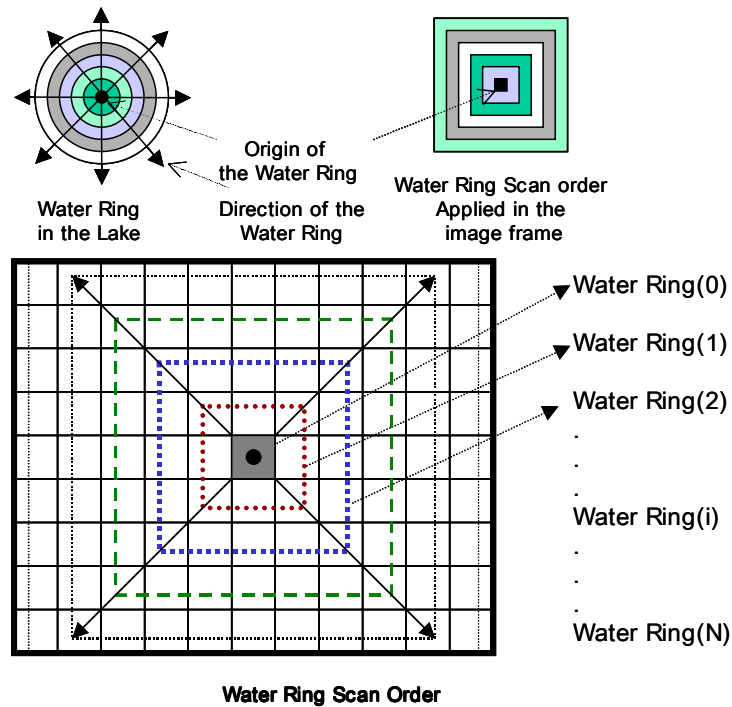


Fig. 5.5. Basic idea of Water Ring scan order (Figure taken from [Par02]).

However, the water ring scan order technique is less efficient as compared to the spiral scan order, because of lack of context modification. The water ring technique does not modify the coding standard, so it is easier to use, but it is not so efficient as the technique proposed in this dissertation. The modification proposed here results in a possibility to use any continuous order of coding macroblocks without losing coding efficiency. Thus, the water ring technique may become as efficient as the spiral scan technique, when combined with context modification.

The spiral scan could be used, for example, for SNR scalability. When there is one object one spiral could be used starting in the centre of the images of the video sequence. Then the outer parts of images may be represented with lower bitrate if the overall bitrate must be reduced. Often the respective quality deterioration, caused by bitrate reducing, is not perceived by a viewer. If the classic scan was used the visual effect is poorer. The comparison of visual effect of bitrate reducing for spiral scan and for the raster scan is shown in Fig. 5.6.

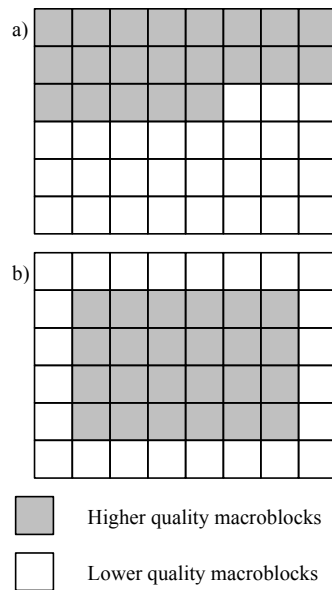


Fig. 5.6. A decoded image using a) raster scan, b) spiral scan.

This effect may be used in scalable video coding, where low-bitrate bitstreams are embedded in a high-bitrate bitstream. The spiral scan of macroblocks may be used and the respective bitstream may be cut after arbitrary number of macroblocks, thus giving good quality in the area of interest in images (see Fig. 5.6 b). In the decoder, the macroblocks which have not been decoded are reconstructed from the low-quality base layer. The standard macroblock order would yield a high quality area to be on the top of the image (Fig. 5.6 a). In the case of more than one object several spirals can be used. An example of the image with two spiral scan areas is shown in Fig. 5.7.

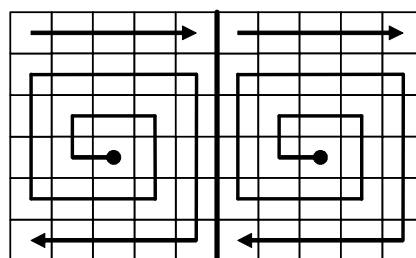


Fig. 5.7. Two spiral scans per one frame.

For this example the areas of good quality, after bitstream cut, are shown in Fig. 5.8. The reason why there are two good quality regions is that each spiral is received at the decoder side as an individual unit, and so they may be cut independently.

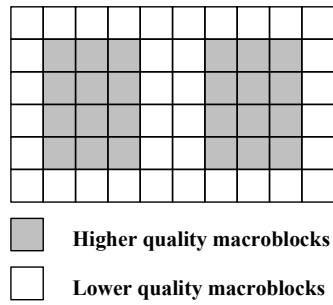


Fig. 5.8. Example of high importance bit allocation for two spiral scans.

The starting point of the spiral scan depends on the position of the object inside the image. It should be located in the centre of the object. An example of spiral scans with various starting points is shown in Fig. 5.9.

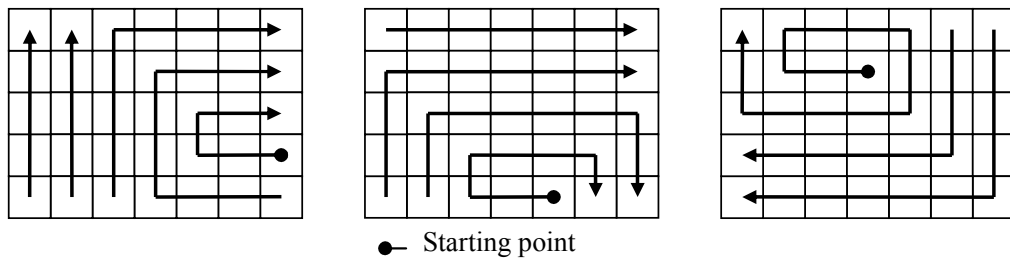


Fig. 5.9. Example of various starting points for the spiral scan.

The areas of good image quality for such spirals are shown in Fig. 5.10.

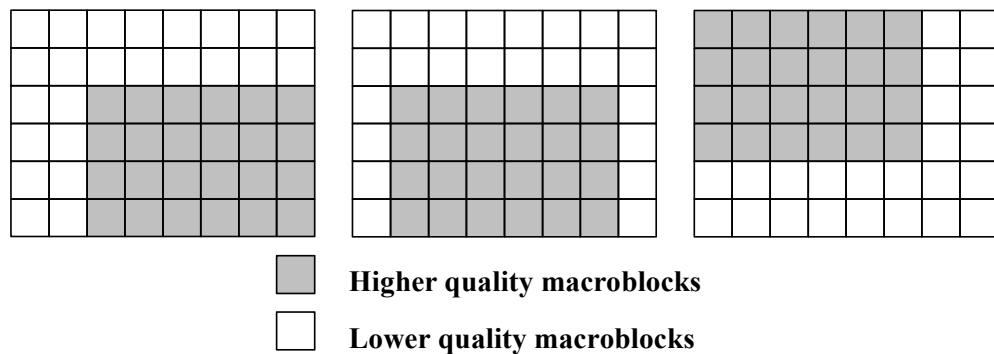


Fig. 5.10. Example of high importance bit allocation for spiral scans with various starting points.

Additionally, for the spiral scan the aspect ratio may be defined, which is directly correlated with the encoded object shape. This aspect ratio defines the proportion of the count of horizontal macroblocks to the count of vertical macroblock. By the use of different aspects ratios it is possible to make a better fit of the spiral scan of

macroblocks to the shape of the encoded object. The better is the fit of the macroblocks scan to the shape of an object the better is the bit allocation. Some examples of various aspect ratios for spiral scan are shown in Fig. 5.11.

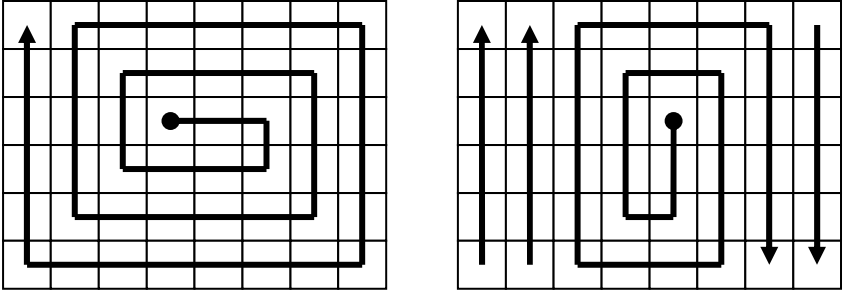


Fig. 5.11. Spiral scan for two different aspect ratios.

In a video sequence, the frames are partitioned into slices. The ability to partitioning the image into slices is very useful for the streaming process. Each slice is transmitted as a separate unit. This means that if in the communication channel there is a packet loss, it is still possible to receive and decode a part of the image. If there is only one slice per image, then in the case of some errors during the transmission process whole frame is lost. Moreover, each slice may be transmitted with various priorities; this means that the different protection methods may be used for each slice. Also for the spiral scan of the macroblocks the slice partitioning may be applied. Here, two types of the slices may be defined. One type of the slice initiate new spiral scan (see Fig. 5.12) with its own parameters: the starting point, the aspect ratio and the slice id. The second type of the slice continues the spiral scan it belongs to by the use of the slice id parameter and the starting point (see Fig. 5.13).

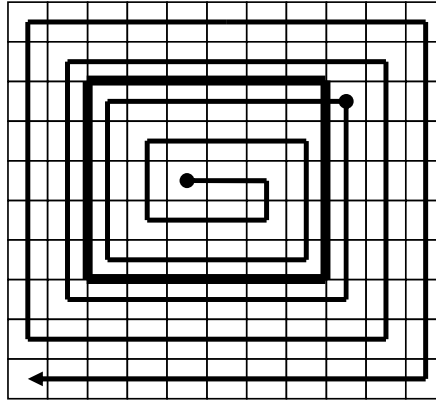


Fig. 5.12. Exemplary frame partitioning into two slices using one spiral scan of macroblocks.

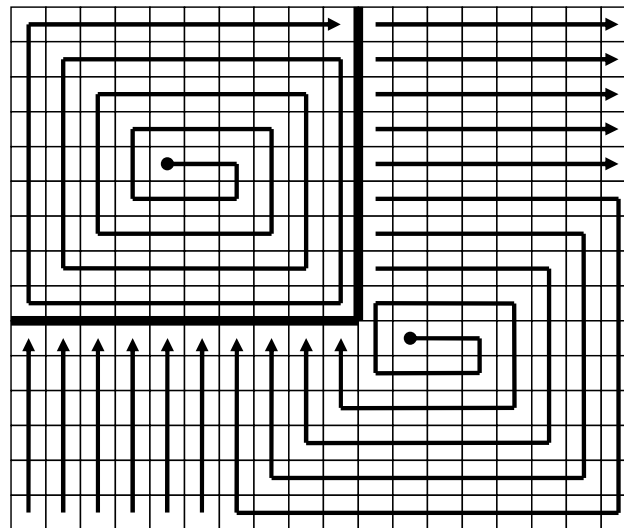


Fig. 5.13. Exemplary frame partitioning into two slices using two spiral scans of macroblocks.

5.3. *Spiral Scan for Quality Scalability in AVC Codecs*

5.3.1 Introduction

A macroblock is the basic coding unit, but, as it was mentioned before, the neighboring macroblocks influence the encoding process of current macroblock. Already the MPEG-2 standard takes advantages of predictive encoding of DC coefficients and predictive encoding of motion vectors. For the raster (linear) scan, the direction of prediction is constant, while it must be adapted to the current direction of

processing in the spiral scan. Therefore, some modifications in the coding algorithm are needed in order to preserve high compression efficiency.

In this dissertation the AVC/H.264 encoder has been adapted to be able to encode image in the spiral order of macroblocks. The AVC/H.264 is a new technique which is much more efficient than older ones because many more elements are encoded using sophisticated adaptive predictions with contexts defined in several ways. Therefore, the respective modifications, for new coding order, are much more complex.

For the prediction of various syntax elements, the data from neighboring macroblocks are taken. For macroblocks, the AVC standard defines four possible neighboring macroblocks, which can be used in prediction process. This are: left, up, up left and up right neighbors (see Fig. 5.14).

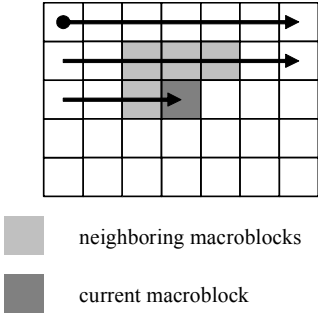


Fig. 5.14. Neighborhood defined in AVC/H.264 codec.

In case of the spiral scan the position of the neighborhood depends directly on the direction of processing of the current macroblock on the spiral. Only the already processed macroblocks may be used for the prediction. The Fig. 5.15 shows the neighboring placement, which is used as a context for macroblock, depending on the position on the spiral.

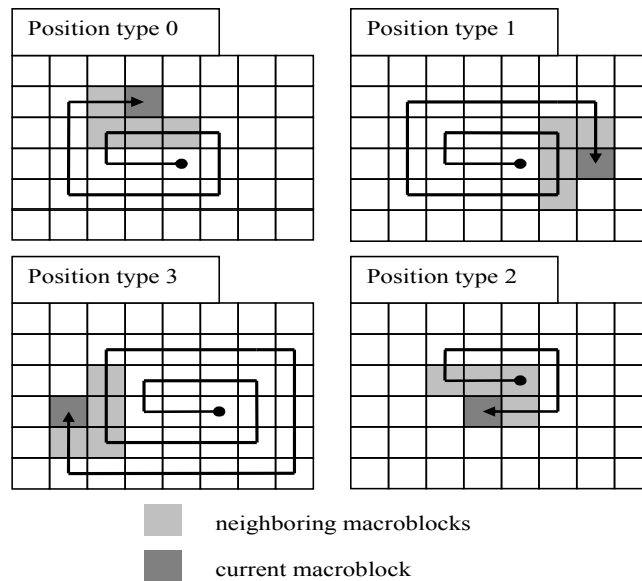


Fig. 5.15. Neighborhood for the spiral scan.

Because of the fact that the H.264 encoder uses fixed neighborhood, as was shown before in Fig. 5.14, the new neighborhood in the case of spiral scan cannot be used directly by this encoder. So, the following prediction tools need adaptation to spiral scan:

- prediction of (4×4)-pixel, (8×8)-pixel and (16×16)-pixel luminance blocks for intra frame coding,
- prediction of chrominance blocks for intra frame coding,
- motion vectors prediction for all block sizes,
- prediction of macroblock encoding parameters,
- context prediction for CABAC coding:
 - block-based prediction,
 - bit-based prediction.

The modifications (see Annex A) do not influence the bitstream syntax, only the semantics of the bitstream elements change.

An alternative technique proposed in [Par02] has a very similar order of macroblocks processing, but it has also one major disadvantage as compared to the technique proposed in this doctoral dissertation: it does not exploit fully the available context for predictive coding. In the case of water ring scan and the spiral scan without context usage modification there is a decrease of coding efficiency compared to raster scan.

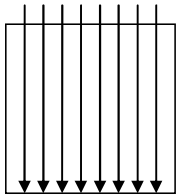
5.3.2 Intra-frame prediction

For the intra prediction in the AVC/H.264 encoder there is a spatial prediction of the pixels of whole macroblock or a block (which is part of the macroblock) by the use of the available pixels from the boundaries of the neighboring macroblocks or blocks. Depending on the size of the block, the intra prediction modes in the AVC/H.264 encoder may be partitioned into three groups:

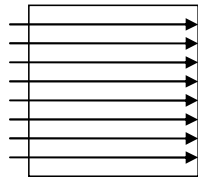
- prediction of (16×16) -pixel blocks,
- prediction of (8×8) -pixel blocks,
- prediction of (4×4) -pixel blocks.

For each size of block individual prediction algorithms are defined. In the case of the (16×16) -pixel blocks there are four possible predictions which may be graphically represented as:

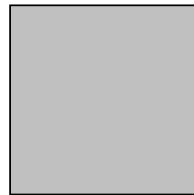
vertical
prediction,



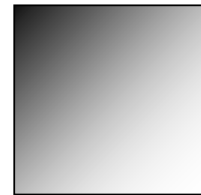
horizontal
prediction,



prediction of DC
coefficient,



plane prediction.



For the spiral scan of macroblocks those predictions have to be modified independently for each position on the spiral. The Fig. 5.16 A – D show four groups of predictions for each of the cases.

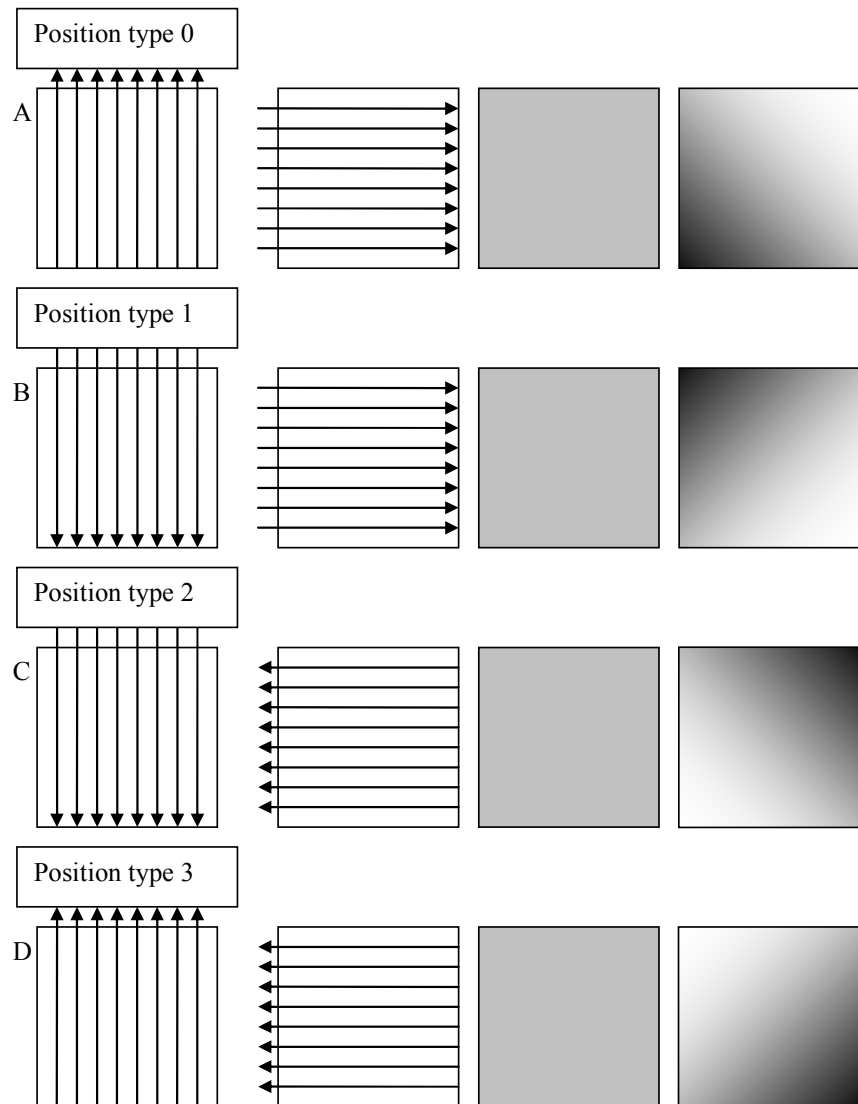


Fig. 5.16. Four groups of prediction modes for spiral scan.

For the prediction of (8×8) -pixel blocks and (4×4) -pixel blocks, the adaptation to the spiral scan is more complex. Here, the prediction is based on blocks instead of macroblocks. It means that for classic scan the blocks are processed in the order shown in Fig. 5.17.

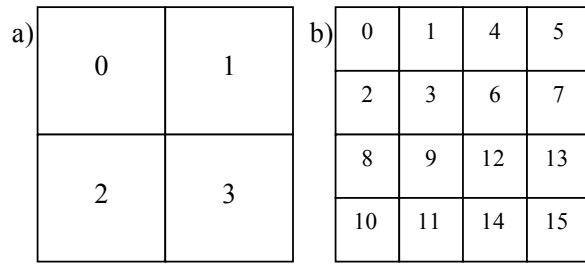


Fig. 5.17. Block coding order defined in AVC/H.264 codec.

But, for the spiral scan, the block coding order as well as macroblock coding order has to be modified. Depending on the currently encoded macroblock position on the spiral, four block coding orders are possible. Fig. 5.18 A – D show the block coding orders (within macroblock) for the spiral scan.

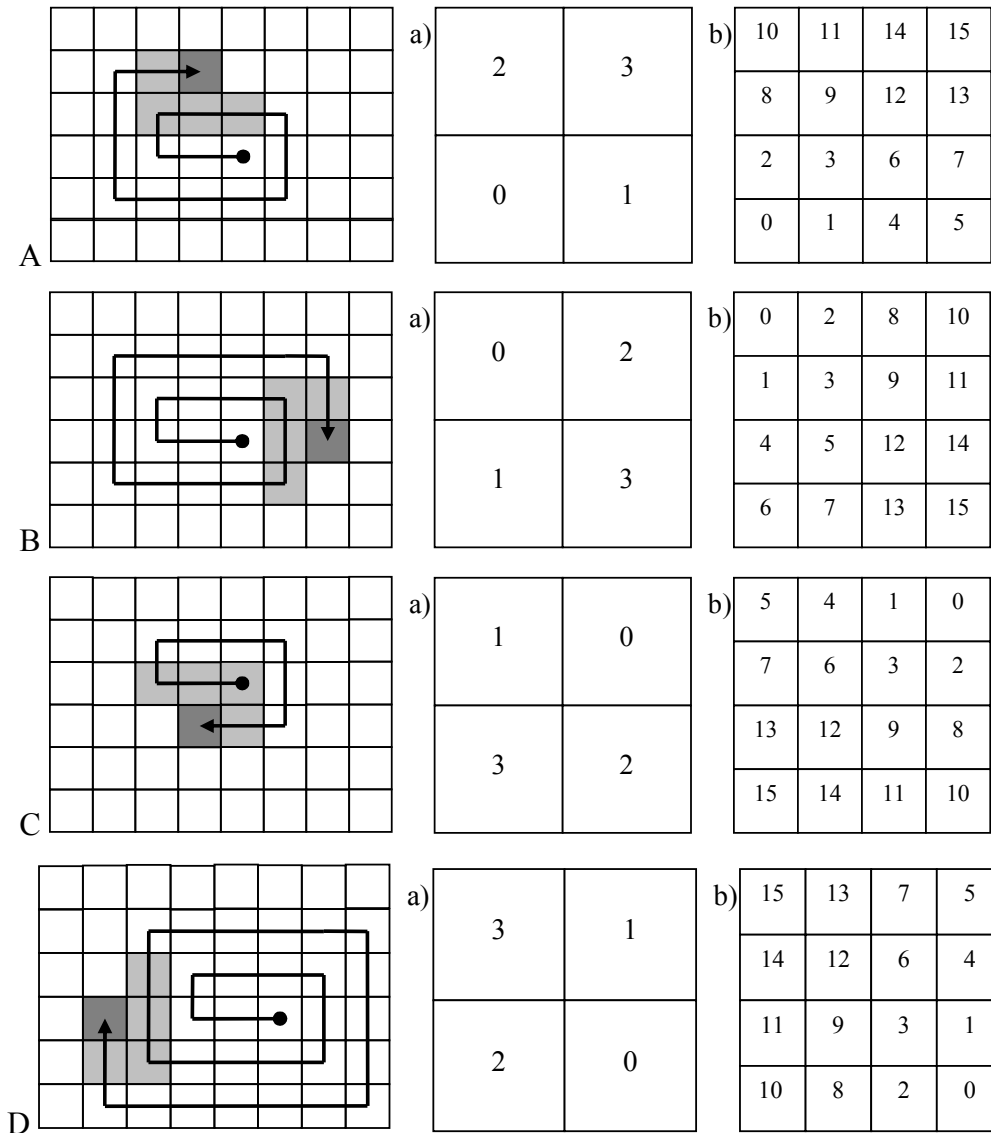
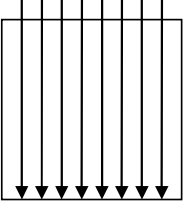
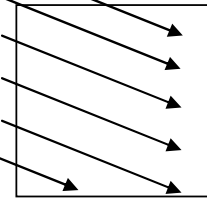
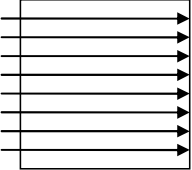
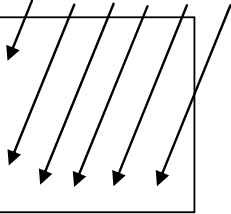
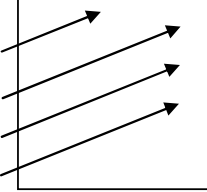
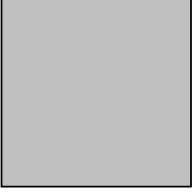
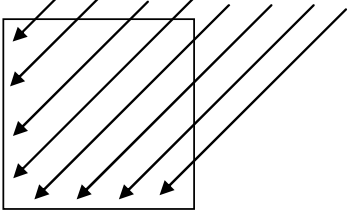
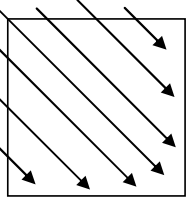
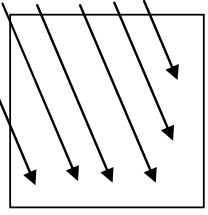


Fig. 5.18. Block coding order for the spiral scan. a) for 8×8 pixel blocks, b) for 4×4 pixel blocks.

For the (8×8)-pixel blocks and the (4×4)-pixel blocks there are nine possible predictions (a DC coefficient prediction and eight directional predictions), which may be graphically represented as:

Table 5.1. Spatial prediction modes.

<p>vertical prediction</p> 	<p>horizontal down prediction</p> 	<p>horizontal prediction</p> 
<p>vertical left prediction</p> 	<p>horizontal up prediction</p> 	<p>prediction of DC coefficient</p> 
<p>diagonal down left prediction</p> 	<p>diagonal down right prediction</p> 	<p>vertical right prediction</p> 

The adaptation of spatial prediction for proposed macroblock coding order is done by modification of the direction of prediction. A set of available directions of prediction depends on the position of available neighboring blocks or macroblocks. Thus, there are four groups of prediction defined:

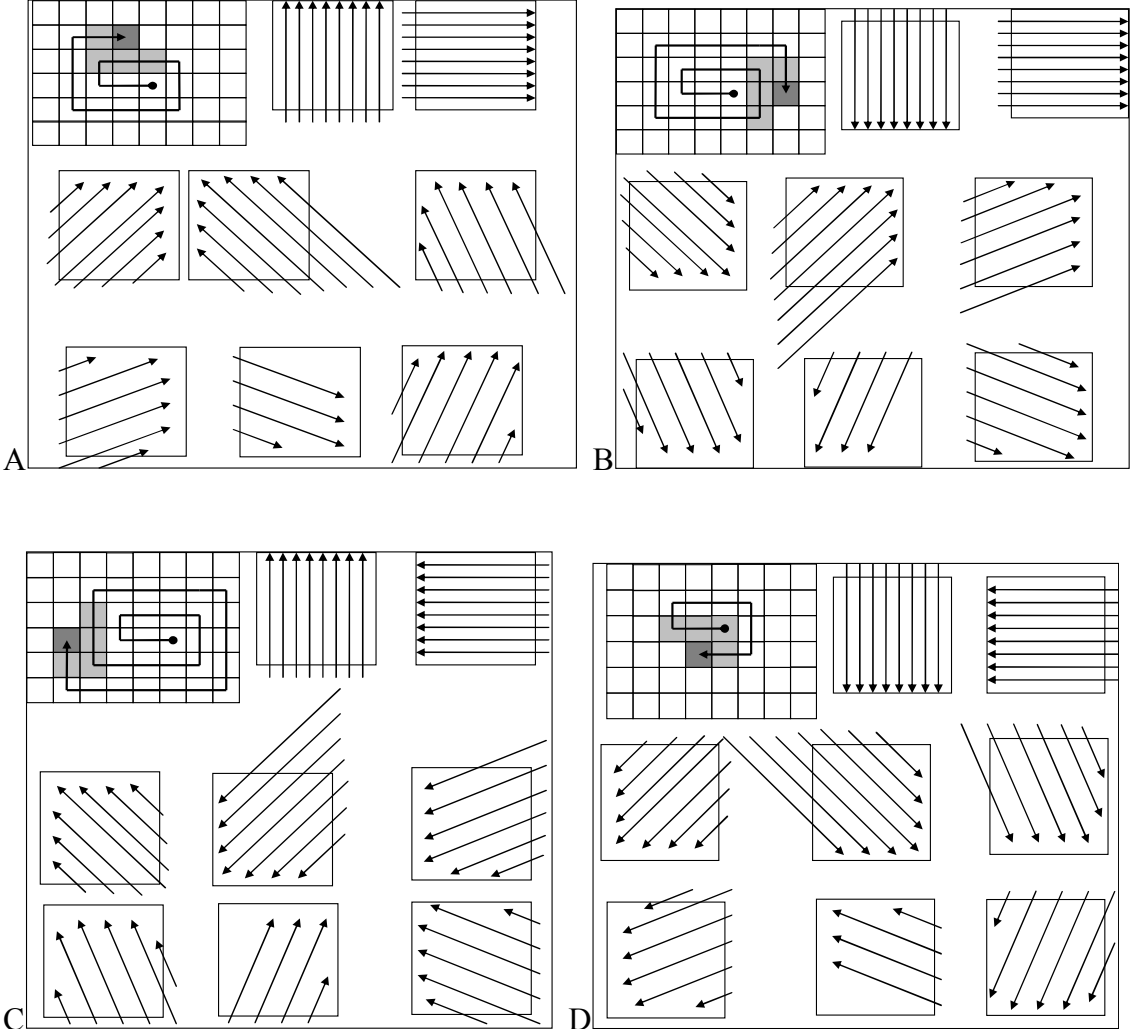


Fig. 5.19. Intra 4x4 and 8x8 prediction for the spiral scan.

5.3.3 Inter-frame prediction

The temporal prediction in H.264/AVC encoder is done at block level. The encoded macroblock is partitioned into sub-blocks. For each block independent motion estimation is performed. Motion vectors are predicted from neighboring macroblocks and the difference between motion vector and its prediction is encoded. The prediction

is done by the use of the motion vectors from the neighboring blocks or macroblocks. Possible macroblock partitioning and the coding order of the partitions is shown in Fig. 5.20.

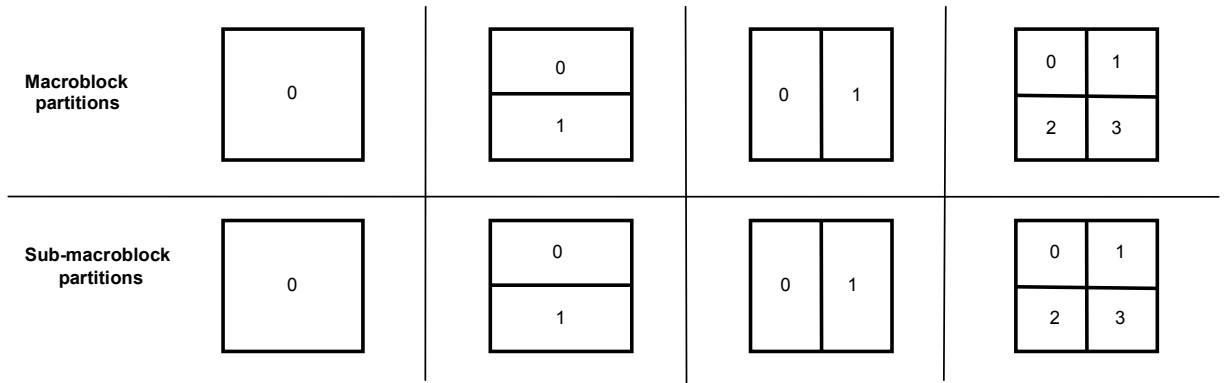


Fig. 5.20. Macroblock partitioning and coding order defined in H.264/AVC.

For the spiral scan, the author has defined four possible orders of coding the partitions (see. Fig. 5.15), depending on the macroblock position in the proposed scan order. It is shown in Fig. 5.21.

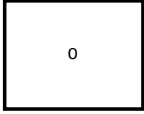
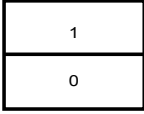
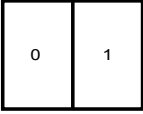
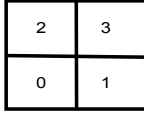
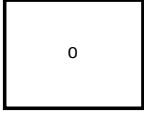
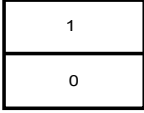
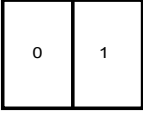
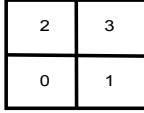
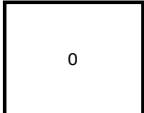
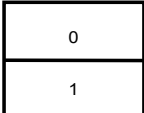
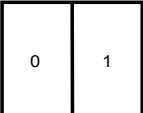

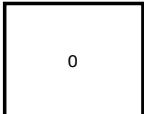
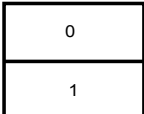
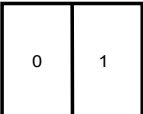
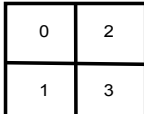


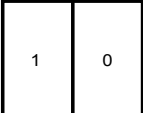


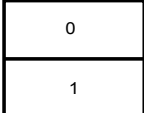
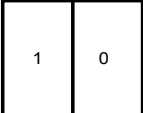

Macroblock position type 0	1 macroblock partition of 16*16 luma samples and associated chroma samples	2 macroblock partitions of 16*8 luma samples and associated chroma samples	2 macroblock partitions of 8*16 luma samples and associated chroma samples	4 sub-macroblocks of 8*8 luma samples and associated chroma samples
Macroblock partitions				
Sub-macroblock partitions				
Macroblock position type 1	1 macroblock partition of 16*16 luma samples and associated chroma samples	2 macroblock partitions of 16*8 luma samples and associated chroma samples	2 macroblock partitions of 8*16 luma samples and associated chroma samples	4 sub-macroblocks of 8*8 luma samples and associated chroma samples
Macroblock partitions				
Sub-macroblock partitions				
Macroblock position type 2	1 macroblock partition of 16*16 luma samples and associated chroma samples	2 macroblock partitions of 16*8 luma samples and associated chroma samples	2 macroblock partitions of 8*16 luma samples and associated chroma samples	4 sub-macroblocks of 8*8 luma samples and associated chroma samples
Macroblock partitions				
Sub-macroblock partitions				

Fig. 5.21 Macroblock partitioning and coding order proposed by the author.

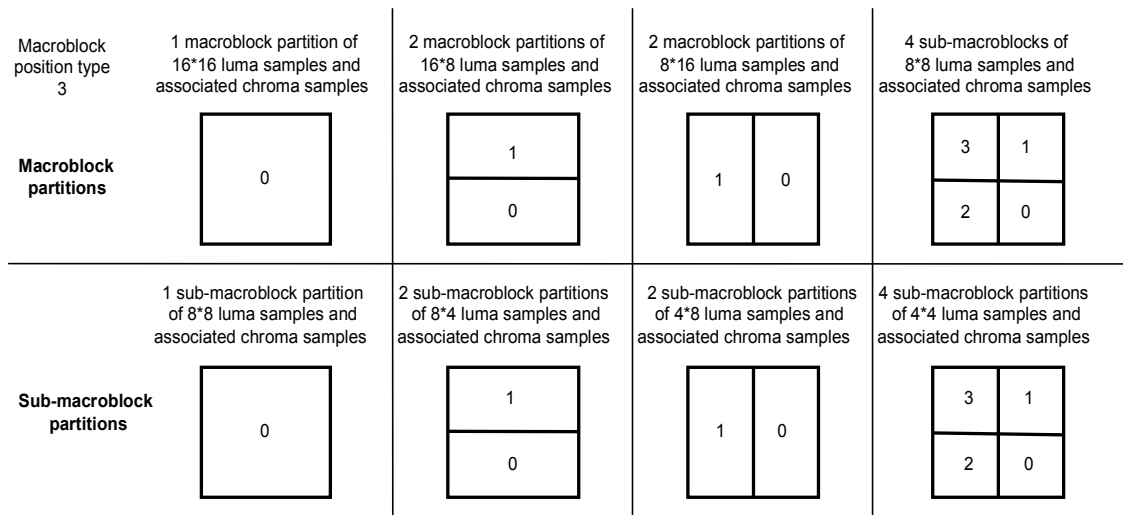


Fig. 5.21. (cont.) Macroblock partitioning and coding order proposed by the author.

The prediction of motion vectors is done by the use of available neighboring blocks which may be situated somewhere around the currently encoded block, and not, as it is defined in H.264/AVC, at the left or above this block.

5.3.4 CABAC coding

The AVC/H.264 encoder uses Context-based Adaptive Binary Arithmetic Coding scheme (CABAC) to encode almost all data which have to be sent to the receiver. During this process, the data called the syntax elements which are to be encoded by the CABAC are binarized. Then each bit of the binarized value is directed one by one into the context modeler. The context modeler is responsible for choosing the appropriate index of the context for currently encoded binary symbol. The context contains two pieces of information: one is the value of the most probable symbol for currently encoded binary symbol, and second is the least probable symbol probability (LPS). The algorithm of choosing the context is defined for each syntax element independently.

For some syntax elements the algorithms do not use the information from neighboring blocks. Therefore, the coding process for such syntax elements does not depend on the macroblock coding order. For example, in the case of coding the transform coefficients there are three types of the syntax elements which are to be encoded by the CABAC. The *significant_coeff_flag* and the *last_significant_coeff_flag* assign the scanning position as an index of the context. And the context for the

coeff_abs_level_minus1 syntax element which depends on the number of previously decoded/encoded significant coefficients.

For other syntax elements the algorithms of choosing the context depend on the content of neighboring macroblocks or blocks. Therefore, for those syntax elements the macroblock scan order influences on the coding process. It means that the appropriate data from neighboring blocks or macroblocks are used as a context. The syntax elements for which the data from neighboring blocks or macroblocks are the context are following:

- *mb_skip_flag*,
- *mb_field_decoding_flag*,
- *mb_type*,
- *coded_block_pattern*,
- *mp_qp_delta*,
- *ref_idx_l0* and *ref_idx_l1*,
- *mvd_l0* and *mvd_l1*,
- *intra_chroma_pred_mode*,
- *coded_block_flag*.

The standard AVC/H.264 defines two neighbors which may be used as a context. The neighbors may be the left block and the upper block for block-based prediction or the left bit and the upper bit for bit-based prediction. The Fig. 5.22 shows the neighborhood for the macroblock.

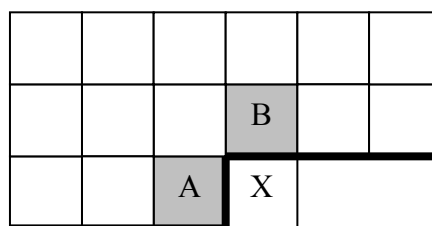


Fig. 5.22. Standard neighborhood for CABAC coding.

While coding the macroblocks in spiral scan order, the default neighbors defined by AVC/H.264 standard are usually not available. Lack of available neighbors decreases the efficiency of CABAC coding. So, in order to achieve the same coding efficiency as for the raster scan it is needed to use data which are available for currently encoded syntax element. Four cases of available neighborhood may be defined for the CABAC

coder when the spiral scan is used. It may be presented graphically in the following way:

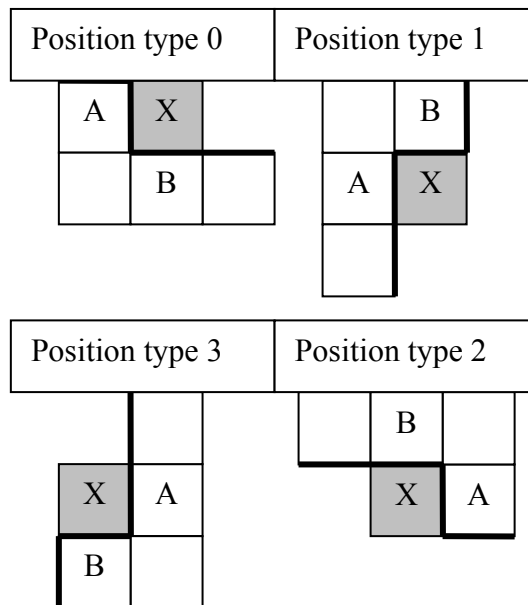


Fig. 5.23. Neighborhood for CABAC with spiral scan.

For some cases the context depends on the direction of available neighboring data for current syntax element. It means that there exists geometrical asymmetry for the context choosing which is caused by the initial values of the contexts. The syntax elements may be partitioned into two groups:

- symmetric syntax elements,
- asymmetric syntax elements.

The syntax elements for which contexts are symmetric use the following condition:

$$context = neighborA + neighborB, \text{ where } neighborA \text{ and } neighborB \text{ are } 0 \text{ or } 1$$

The possible results are:

- 0, when $neighborA$ and $neighborB$ were 0,
- 1, when $neighborA \neq neighborB$,
- 2, when $neighborA$ and $neighborB$ were 1.

To this group the following syntax elements belong:

- *mb_skip_flag*,
- *mb_field_coding_flag*,
- *mb_type*,
- *mb_qp_delta*,

- *intra_chroma_pred_mode*,

The syntax elements for which the contexts are asymmetric use the following condition:

$$context = neighborA + 2 \cdot neighborB, \text{ where } neighborA \text{ and } neighborB \text{ are } 0 \text{ or } 1$$

The possible results are four integer values in range [0..3].

The following syntax elements belong to this group:

- *coded_block_pattern*,
- *ref_idx_l0* and *ref_idx_l1*,
- *coded_block_flag*.

For the spiral scan this asymmetry has to be taken into account and the appropriate context must be used.

5.4. Model of codec with quality scalability

5.4.1 Overview

As it has been proved earlier, the spiral scan may be used as an alternative for other macroblock scans, like the raster scan. The spiral scan may be perceived as a tool for coding the regions of interest (RoI). In the domain of still image compression, the coding of the region of interest has already been exploited in approved international image compression standard JPEG2000 [ISO00, Tau02]. In the video compression, there are only some proposals of region of interest coding, like in [Zhou03]. Here, the author proposes a tool for such functionality.

In the video coding, the RoI is a region where the quality should be the highest, compared to the quality of remaining part of the image. An alternative definition of a RoI is that it is the region surrounding the centre of interest (CoI) which is the central point the human observer focuses on (see Fig. 5.24).

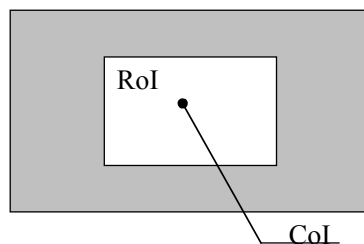


Fig. 5.24. Region of interest (RoI) with its centre of interest (CoI).

The *CoI* may be a geometrical centre of the most interesting object (from human point of view). When the spiral scan is used as a tool for a *RoI* coding the *CoI* may be a starting point for the spiral. Thus, the square region around the *CoI* is defined (see Fig. 5.25).

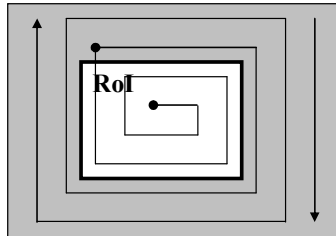


Fig. 5.25. Spiral scan for ROI coding.

Here, the variable bit allocation may be used to achieve high quality for the macroblocks inside the region of interest and the low quality for remaining part of the image. But, the spiral scan may be also used as a tool for obtaining the fine granularity scalability (FGS) – quality (SNR) scalability.

Let us assume that a coder produces layered video representation. Each layer represents a different level of quality. If the most important macroblocks, in the enhancement layer are encoded at the beginning and the less important macroblocks later, then at the decoder side it is possible not to receive some of the less important macroblocks and still achieve the good subjective quality of image. Here, in place of the macroblocks which were not received in the decoder, the corresponding data from lower layer (low quality) are taken.

The block diagrams of the encoder and the decoder with *RoI* coding are shown in Fig. 5.26 and Fig. 5.27.

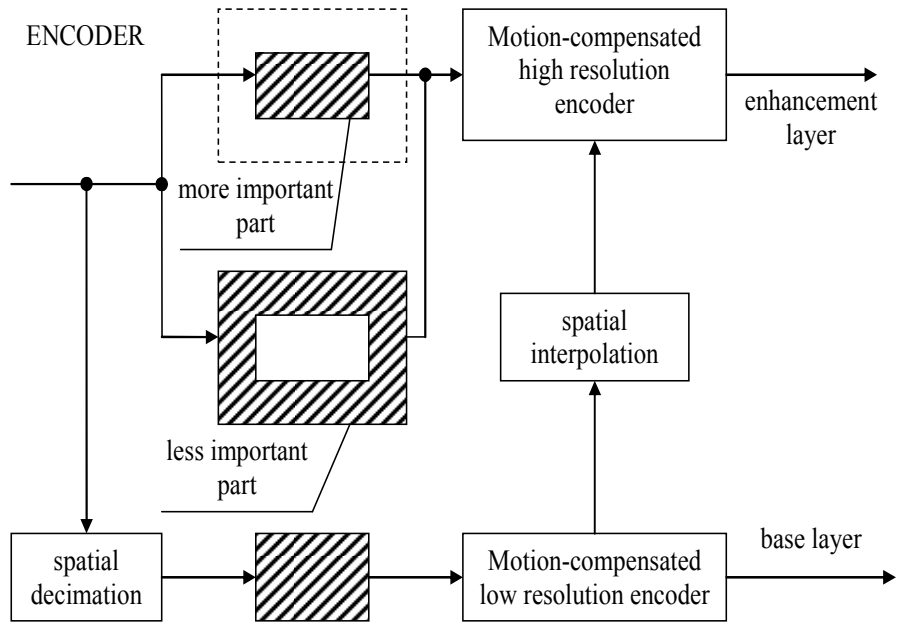


Fig. 5.26. Quality (SNR) scalability with region of interest encoding.

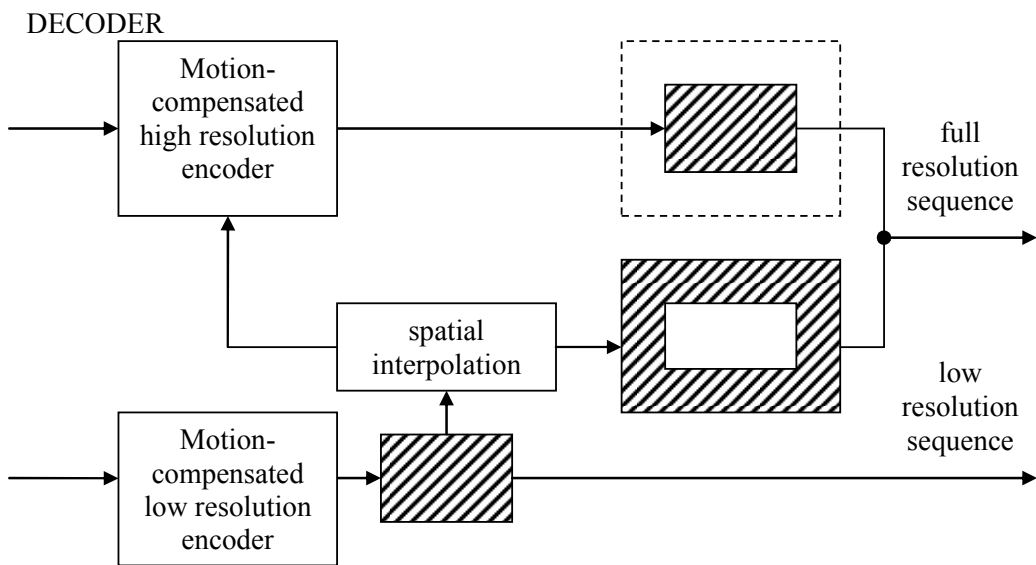


Fig. 5.27. Decoding of a quality scalable bitstream with region of interest decoding.

The proposed codec with spiral scan of macroblocks provides a functionality of coding the regions of interest and fine granularity scalability (quality scalability).

5.4.2 Spiral Scan in AVC - Complexity

In order to estimate the possible overhead of complexity for the AVC codec using the spiral scan with the context modification, the number of basic operations has been roughly counted on the basis of algorithm analysis. The two exemplary coder paths are presented in Table 5.2 and Table 5.3. The tables show the number of logical, arithmetical, comparison operation, memory access for the coder with the raster scan used and then with the spiral scan used. The Table 5.2 shows the number of operations for encoding one picture when the 4×4 block size spatial prediction is used for macroblock coding.

Table 5.2. Comparison of estimated number of operations needed for go through the prediction 4×4 path for raster scan and spiral scan for intra prediction.

operation	number of operations	
	raster scan	spiral scan
+, -, <<, >>, ++, --, !, &,	8230	8240
*, /, %	2250	2260
Comparison	1700	1750
memory access	11320	11340

The Table 5.3 shows the number of operations for encoding one picture when temporal prediction of 16×16 block size is used for encoding one macroblock.

Table 5.3. Comparison of estimated number of operations needed for the 16×16 inter-frame prediction for the raster scan and the spiral scan (block match search with range of 64 pixels).

operation	number of operations	
	raster scan	spiral scan
+, -, <<, >>, ++, --, !, &,	266000	266300
*, /, %	2000	2150
Comparison	267500	268000
memory access	266800	267200

By the use of the estimated codec complexity for the spiral and the raster scan the comparison of both may be made. Complexity overhead for the codec with spiral scan used seems to be no more than 1% when the number of operations is compared.

Chapter 6

Codecs implementation – parameter setting

6.1. Introduction

The models of codecs proposed by the author had to be implemented for verification of its real coding efficiency. The coding efficiency of specific encoder depends on the set of parameters. These parameters, like codes for encoded symbols, interpolation filter characteristic or encoding modes hierarchy may strongly influence on encoder behavior. Here, in this dissertation several models of encoders have been tested. For each of them, various parameters have to be experimentally determined. For each parameter the appropriate experiments have been prepared and performed. Here, in this chapter, the results of several experiments are presented. On the basis of these results the following parameters has been determined: Huffman codes for symbols representing encoding modes for scalable H.263; Encoding modes hierarchy for scalable H.264 codec; K parameter that controls the intensity of warping in edge-adaptive bi-cubic interpolation for scalable H.264 codec.

6.2. Determining Huffman codes for symbols representing encoding modes for scalable H.263 codec

When the H.263 codec is extended with scalable functionality, the new encoding modes appear. The scalable model structure used by the author introduces in this codec the following encoding modes:

- interpolation mode,
- averaging mode,
- copying from interpolated frame,
- copying from average.

For those modes, the Huffman codes of the symbol representing the chosen mode have to be sending to the receiver. In order to determine the symbols codes, the probability of choosing each of the symbols has to be known. To estimate this probability the encoding process has been done where the modes were chosen, but not encoded. The following test conditions have been set for the experiment:

- number of frames encoded was 60,
- motion vector estimation was full-pel and half-pel,
- bitrate:
 - base layer 100 kbps – enhancement layer 300 kbps,
 - base layer 200 kbps – enhancement layer 600 kbps,
 - base layer 300 kbps – enhancement layer 900 kbps,
- motion vector search range was ± 15 ,
- four vectors per macroblock enabled,
- interpolation filter was Johnston 12th order,
- open GOP: IPPP.

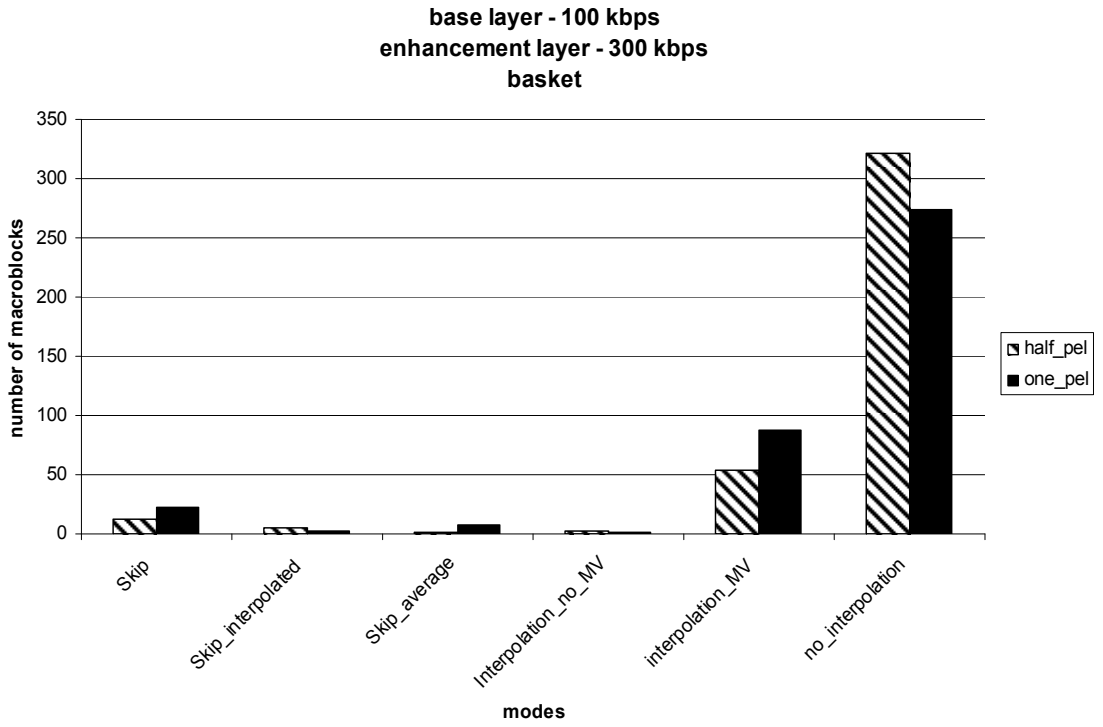


Fig. 6.1 Basket (base layer - 100 kbps, enhancement layer - 300 kbps).

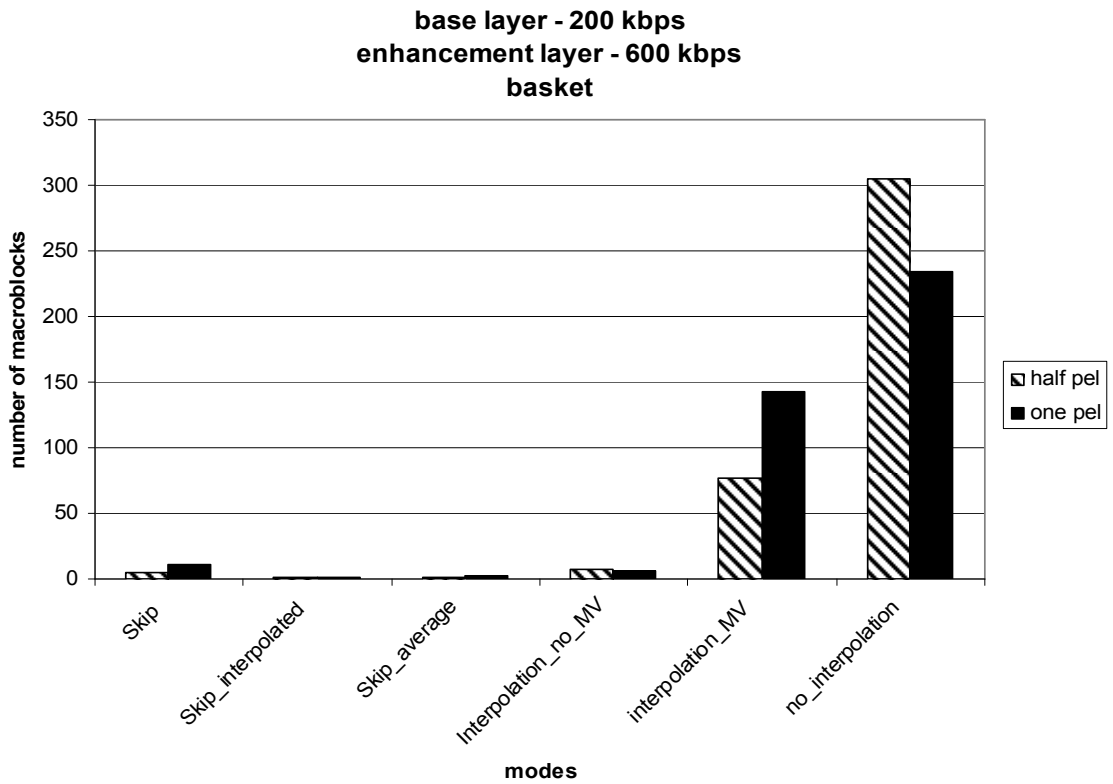


Fig. 6.2 Basket (base layer - 200 kbps, enhancement layer - 600 kbps).

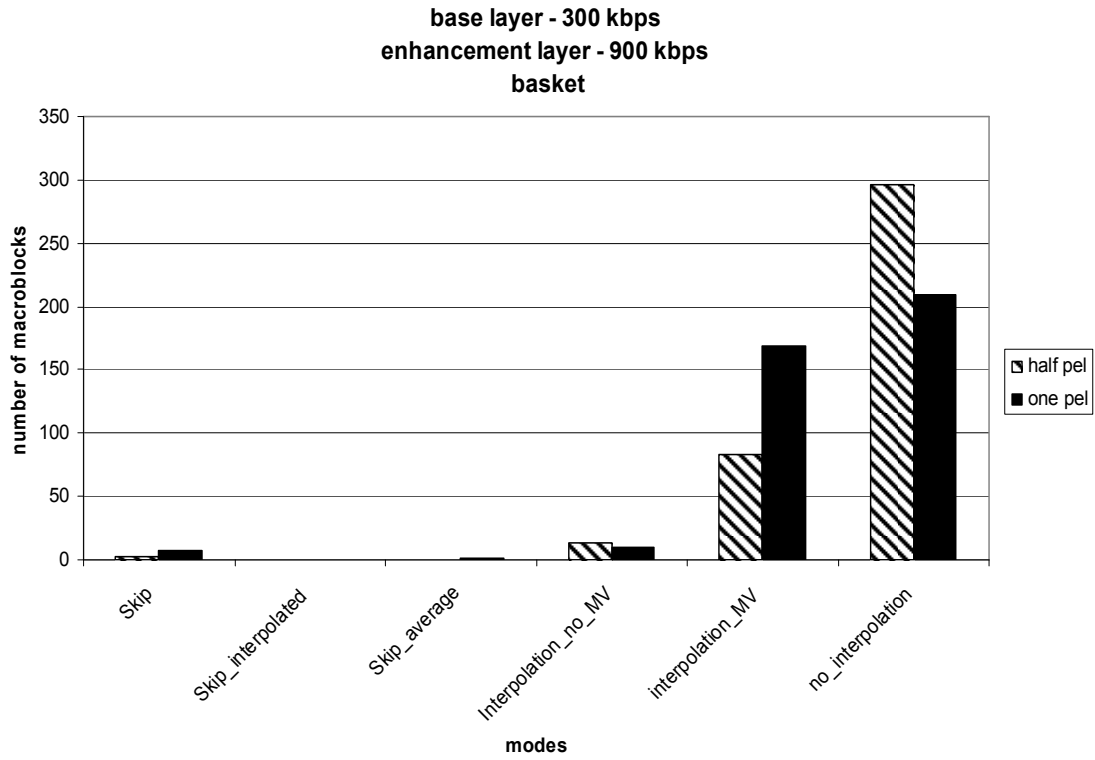


Fig. 6.3 Basket (base layer - 300 kbps, enhancement layer - 900 kbps).

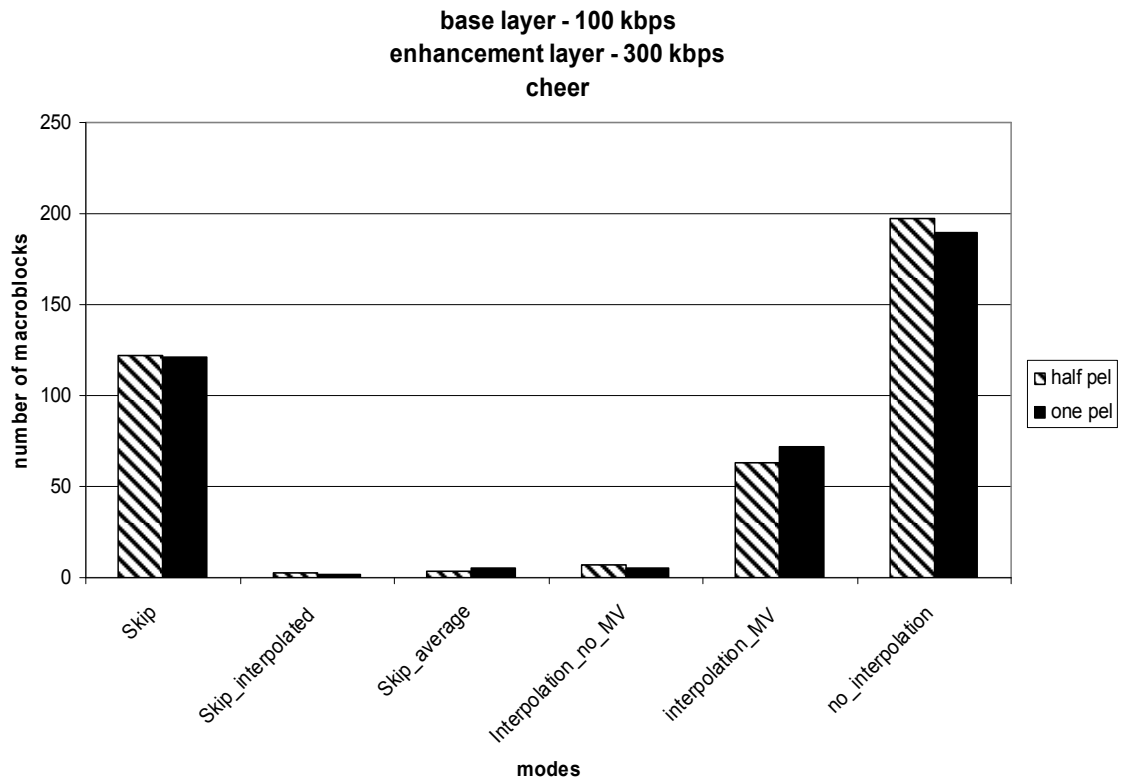


Fig. 6.4 Cheer (base layer - 100 kbps, enhancement layer - 300 kbps).

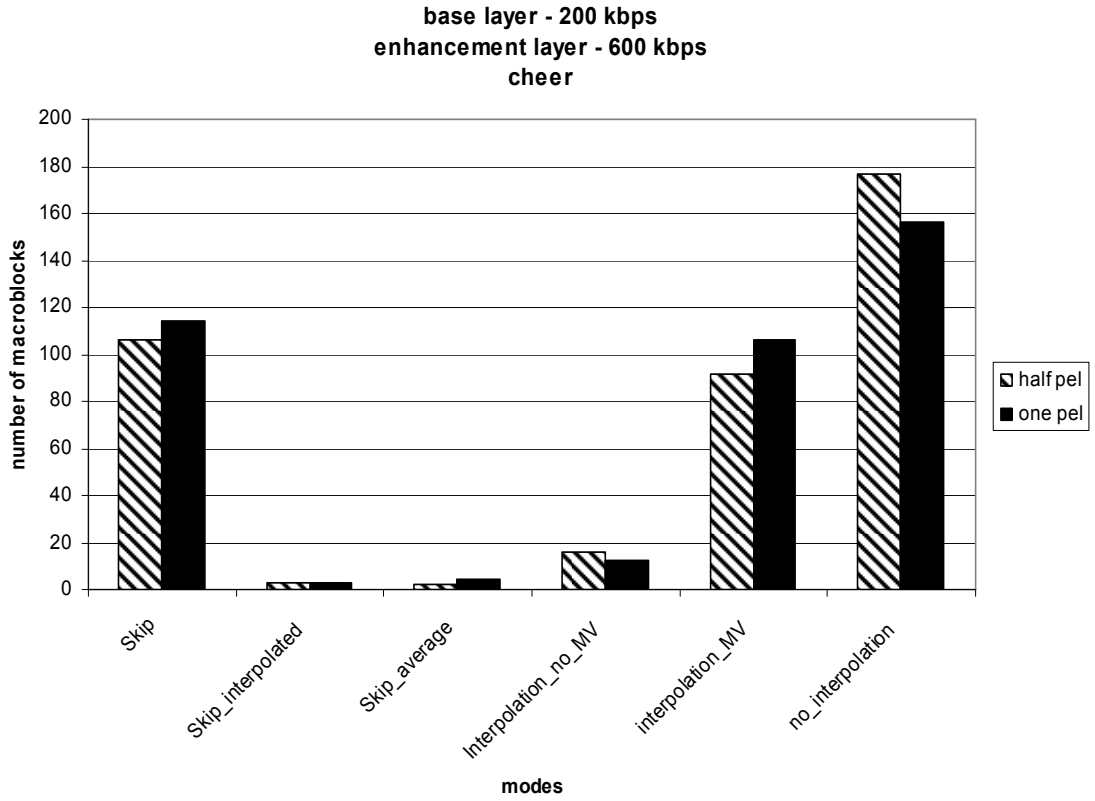


Fig. 6.5 Cheer (base layer - 200 kbps, enhancement layer - 600 kbps).

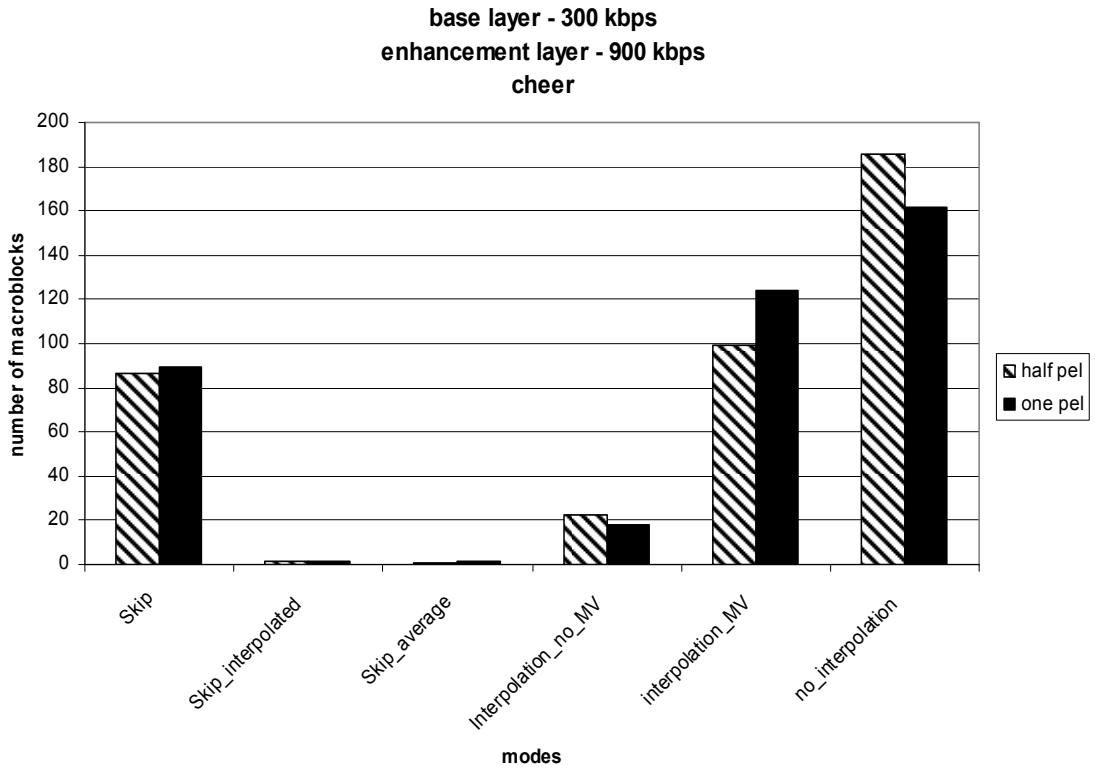


Fig. 6.6 Cheer (base layer - 300 kbps, enhancement layer - 900 kbps).

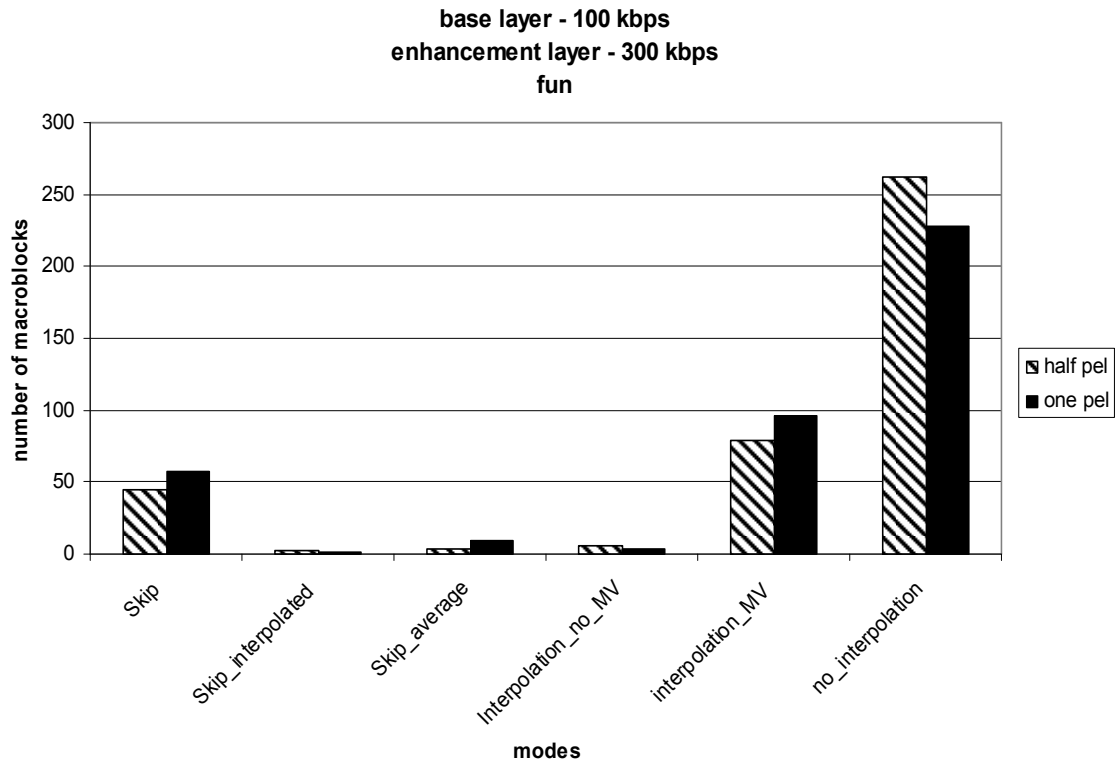


Fig. 6.7 Fun (base layer - 100 kbps, enhancement layer - 300 kbps).

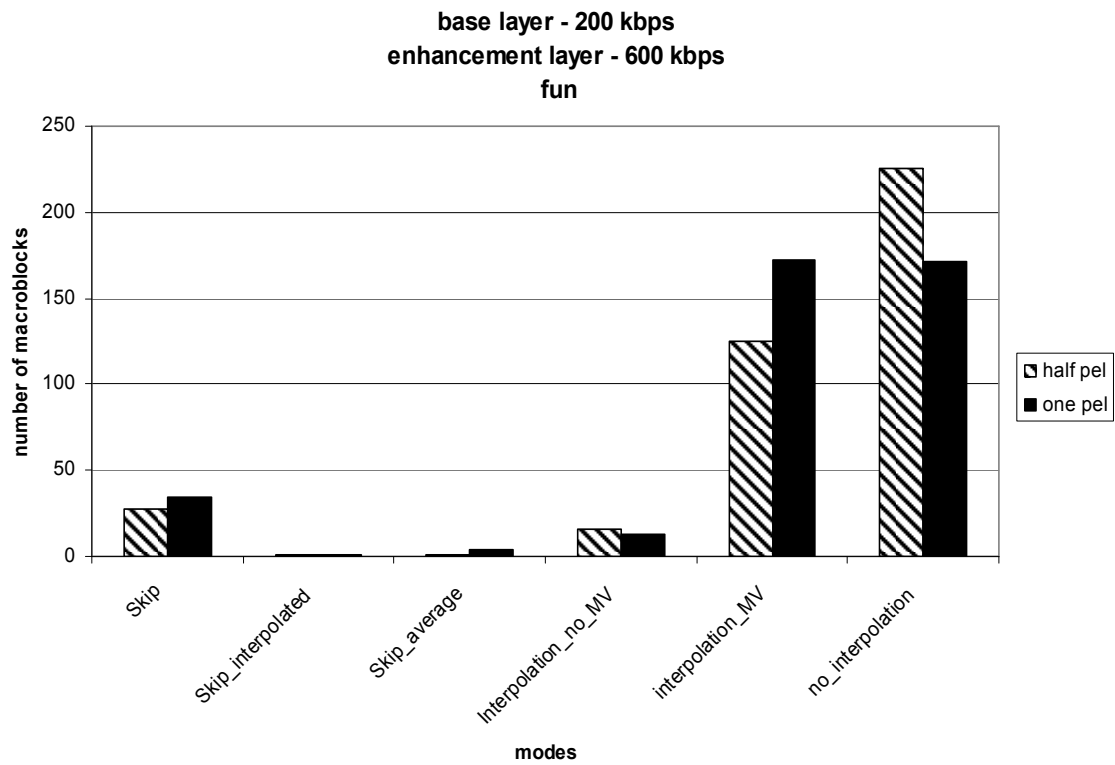


Fig. 6.8 Fun (base layer - 200 kbps, enhancement layer - 600 kbps).

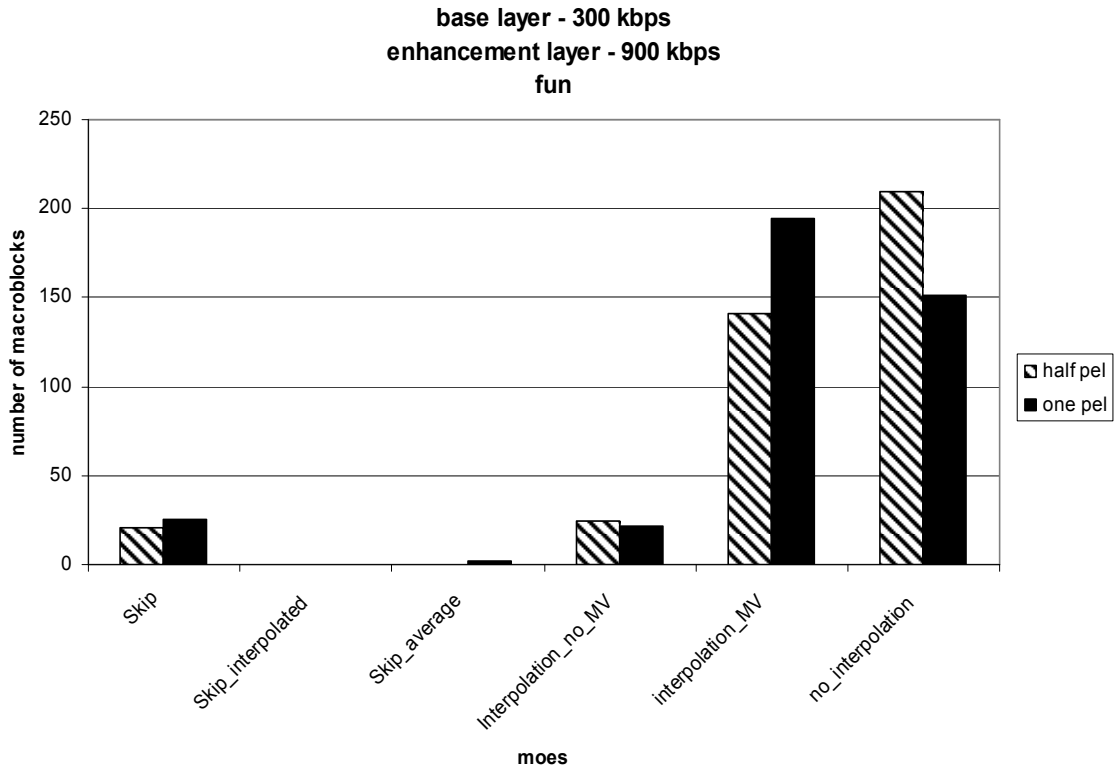


Fig. 6.9 Fun (base layer - 300 kbps, enhancement layer - 900 kbps).

Average results for all test sequences:

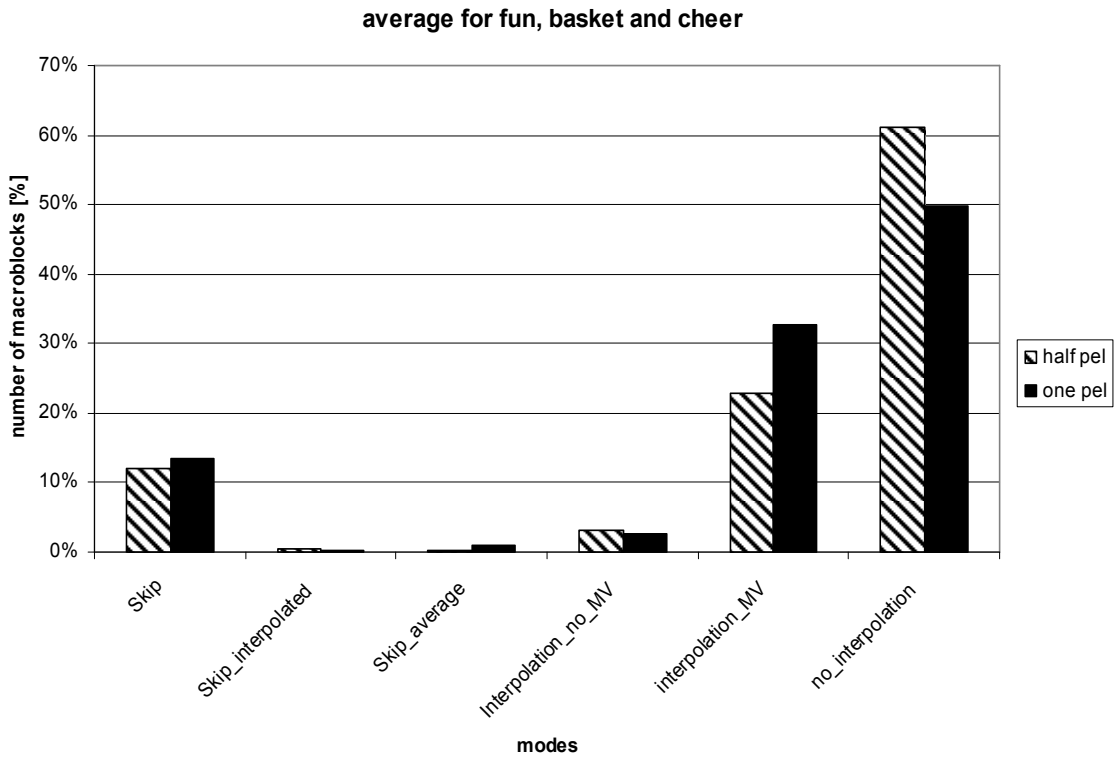


Fig. 6.10 Average results.

Fig. 6.10 shows the average result of all tests presented in Figs. 6.1. – 6.9. There are six modes presented:

- *skip* – it copies a macroblock from reference frame at the same coordinates,
- *skip_interpolated* – it copies an interpolated macroblock from base layer,
- *skip_average* – it copies a macroblock which is an average of an interpolated macroblock from base layer and a macroblock from reference frame at the same coordinates,
- *interpolation_no_MV* – it copies an interpolated macroblock from base layer and adding the residual data,
- *interpolation_MV* – it copies a macroblock which is an average of an interpolated macroblock from base layer and a macroblock from reference frame at the coordinates pointed by motion vectors; the residual data are added,
- *no_interpolation* – it is standard coding by the use of motion-compensated prediction from reference frame.

These average results show that most probable mode is a prediction from previous reference frame and the least probable mode is a copy from lower layer frame. The all mode hierarchy with corresponding Huffman codes is listed in Table 6.1. These codes extend standard codes used by H.263 standard.

Table 6.1 Huffman codes for scalable H.263.

mode	code
prediction from previous reference frame	1
prediction from average of lower layer frame and previous reference frame	00
copy from previous reference frame	010
prediction from lower layer frame	0110
copy from average of lower layer frame and previous reference frame	01110
copy from lower layer frame	01111

6.3. Determining encoding modes hierarchy for scalable H.264 codec

In order to determine the encoding modes hierarchy for scalable H.264 model the following experiment has been performed. The scalable model of encoder, based on non-scalable JM 2.1 software, has been modified to skip encoding mode type cost in mode decision algorithm. Then, count the frequency of choosing the particular modes. On the basis of the results obtained for several test sequences, the modes hierarchy for bitstream coding may be determined. The following test condition has been set:

- Scalable JM 2.1 software based encoder with two layers has been used.
- The CIF input sequences were used.
- The number of encoded frames was 60.
- The GOP structure was open GOP: IBPBPB for base layer and IBBBPBBBP for enhancement layer.
- Each test sequence has been encoded three times for the following fixed quantization parameters: 12, 16, 21.
- The search range was 16 full-pixels.
- Entropy coding method was CABAC.
- Motion vector accuracy was $\frac{1}{4}$ pixel.
- Number of reference frames was 5.
- RD-optimized mode decision was not used.

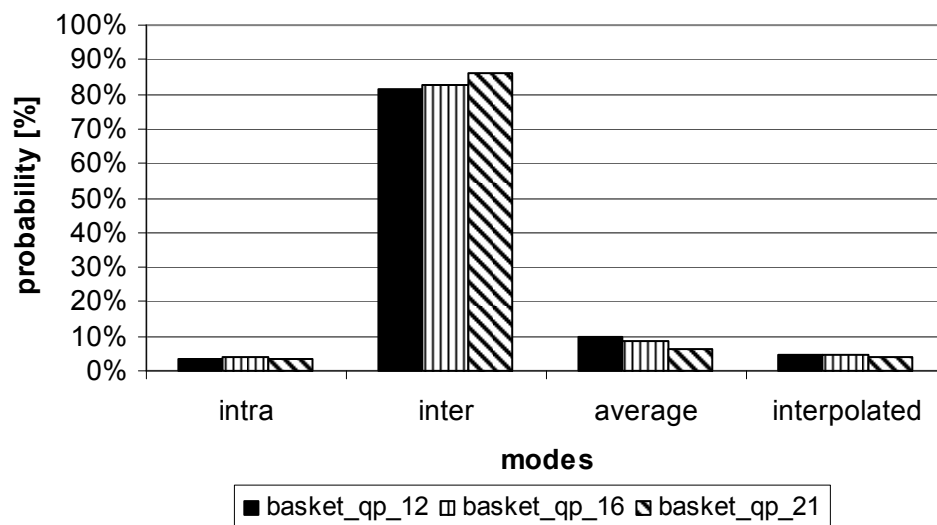


Fig. 6.11 Probability of modes selection for BASKET test sequence.

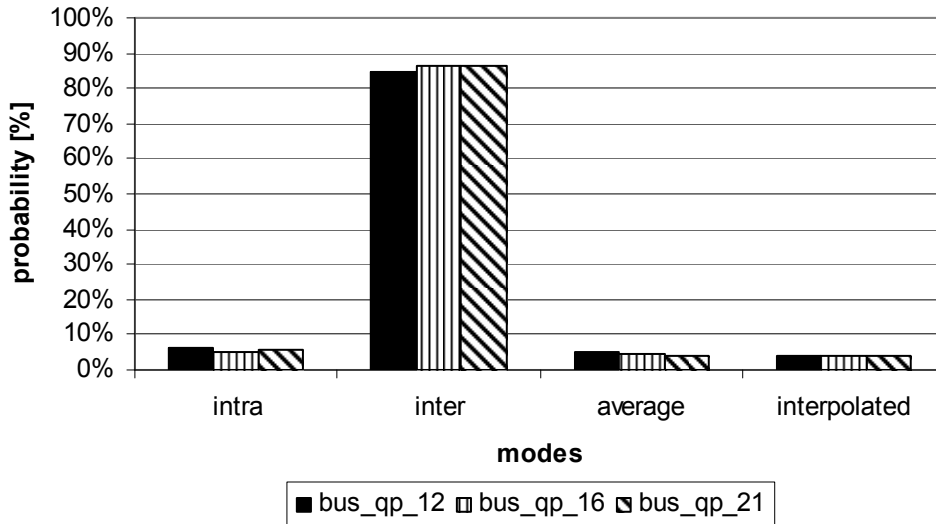


Fig. 6.12 Probability of modes selection for BUS test sequence.

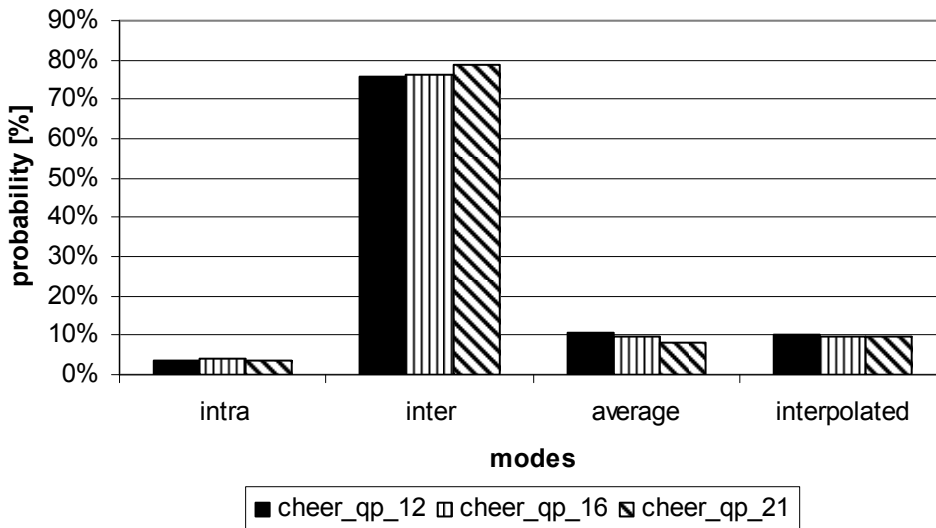


Fig. 6.13 Probability of modes selection for CHEER test sequence.

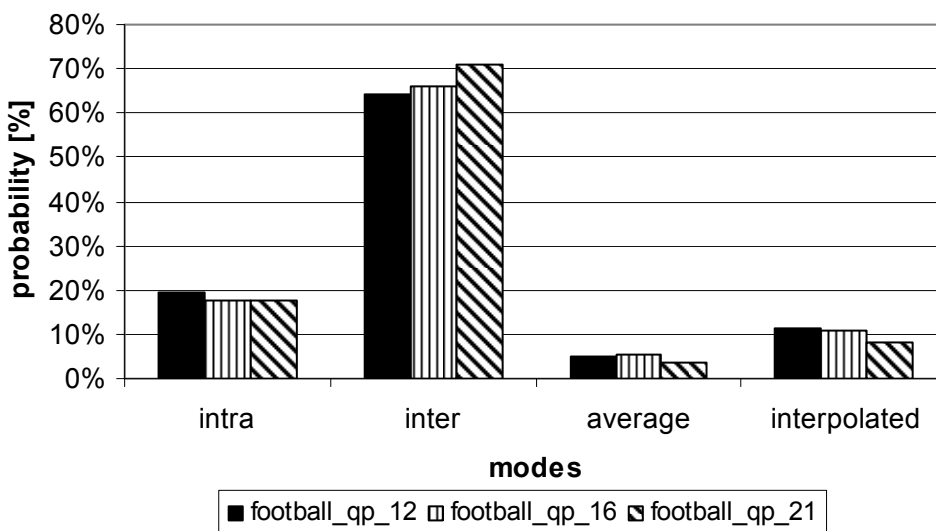


Fig. 6.14 Probability of modes selection for FOOTBALL test sequence.

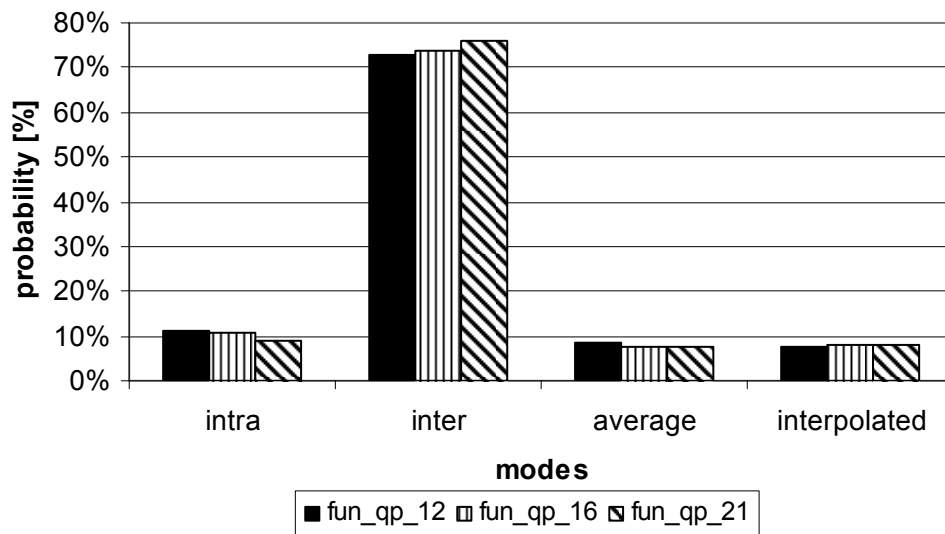


Fig. 6.15 Probability of modes selection for FUN test sequence.

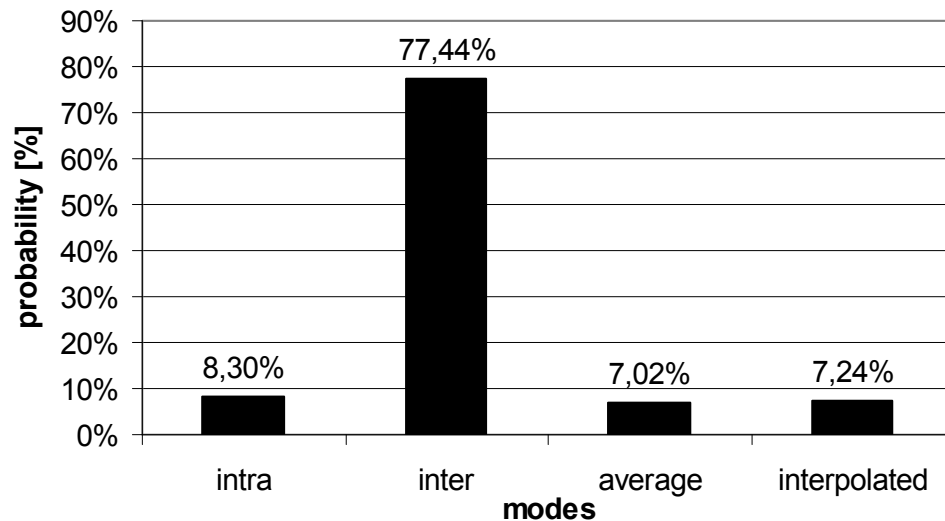


Fig. 6.16 Average probability of modes selection for all test sequences.

Average results for frequency of choosing particular prediction modes from Fig. 6.16 shows that all inter-frame prediction modes together are more probable than intra-frame prediction. For the inter-frame prediction, the most probable mode is prediction from the previous reference frame, then from the interpolated frame, and finally the prediction from the frame that results from averaging the previous reference frame and the interpolated frame from lower layer. A result of this experiment is the prediction mode hierarchy shown in Table 6.2.

Table 6.2 Prediction mode hierarchy.

Frame type	Prediction modes
Intra (I)	<ol style="list-style-type: none"> 1. Spatial interpolation from base layer (16×16 block size). 2. All standard intra prediction modes.
Inter (P)	<ol style="list-style-type: none"> 1. Prediction (forward) from the nearest reference frame. 2. Spatial interpolation from base layer (16×16 - 4×4 block size). 3. Average of spatially interpolated base layer frame and the nearest reference frame. 4. Temporal prediction modes from other reference frames in the order defined in AVC specification. 5. All standard intra modes.
Inter (B)	<ol style="list-style-type: none"> 1. Prediction (forward, backward and bidirectional) from the nearest reference frame. 2. Spatial interpolation from base layer (16×16 - 4×4 block size). 3. Average of spatially interpolated base layer frame and the nearest reference frame. 4. Temporal prediction modes from other reference frames in the order defined in AVC specification. 5. All standard intra modes.

6.4. Determining k parameter for edge-adaptive bi-cubic interpolation for scalable H.264 codec

The aim of the experiment was to determine the k parameter for the interpolation technique described in Chapter 4 Section 4.3.2. The experiment has been done for two decimation filters: 12th order and 24th order FIR filters and two motion vector search ranges: 4 pixels and 64 pixels. The design process of these filters was described in Chapter 4 Section 4.3.2. The short range of motion vector search raises the probability of *interpolation* mode and *average* mode selection and thus the influence of parameter k for encoding efficiency. Several values of parameter k starting from value 1,05 up to 4,05 with step of 0,1 have been used. The experiment was done by the use of three

layered codec based on non-scalable JM 2.1 software. The test conditions were the following:

- I-frame period was 0,5 sec.
- Base layer GOP = IPPP.
- First enhancement layer GOP = IBPBPBPB.
- Second enhancement layer GOP = IBBBPBBBP.
- Input sequences were 30Hz 4CIF format.
- Fixed quantization parameters: QP for I and P frames was 26 and QP for B frames was 30.

Result group 1:

Result for short range search of motion vectors and 12th order decimation FIR filter.

This group of results shows the decrease of the bitrate for test sequences for low order decimation filter (12th order) and short motion vector search range. The search range was up to 4 full pixel accuracy. The results in Fig. 6.17-20 show that using adaptive interpolation may give the bitrate gain between 0,28% up to 3,57%.

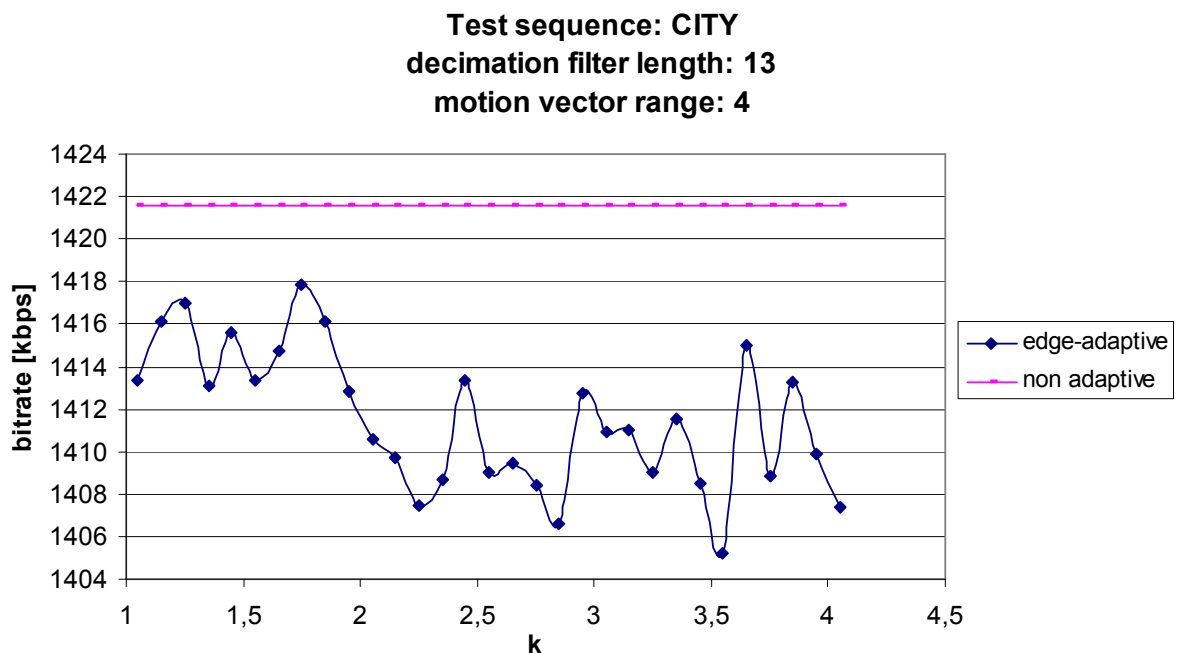


Fig. 6.17. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Test sequence: CREW
decimation filter length: 13
motion vector range: 4

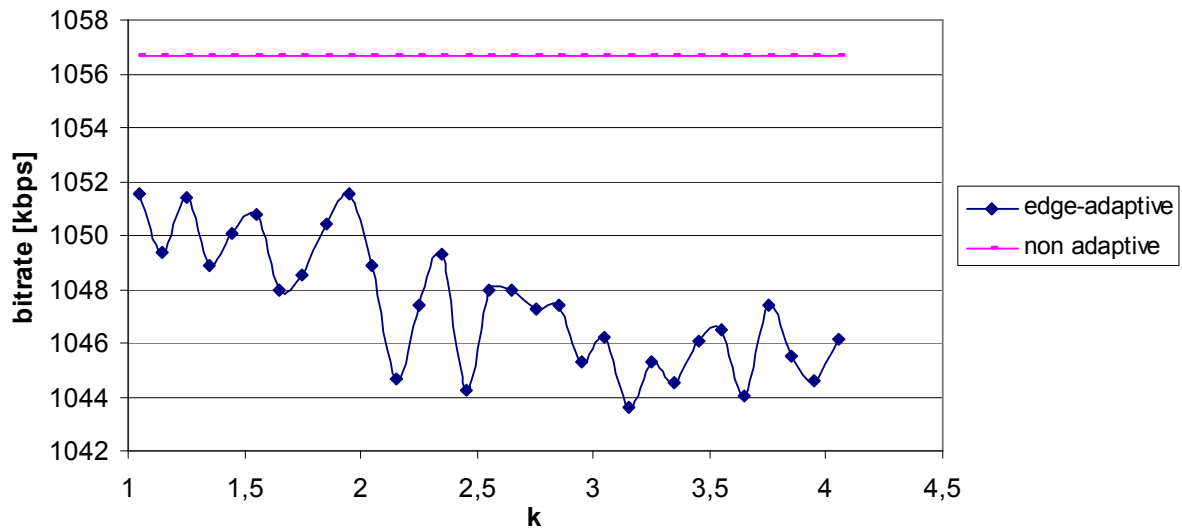


Fig. 6.18. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Test sequence: HARBOUR
decimation filter length: 13
motion vector range: 4

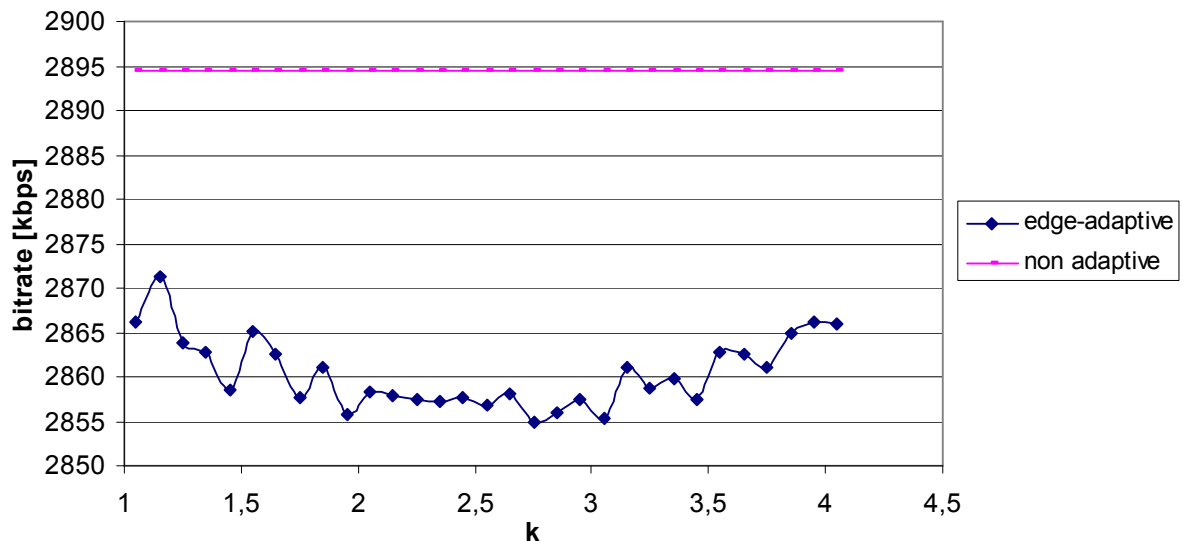


Fig. 6.19. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Test sequence: ICE
decimation filter length: 13
motion vector range: 4

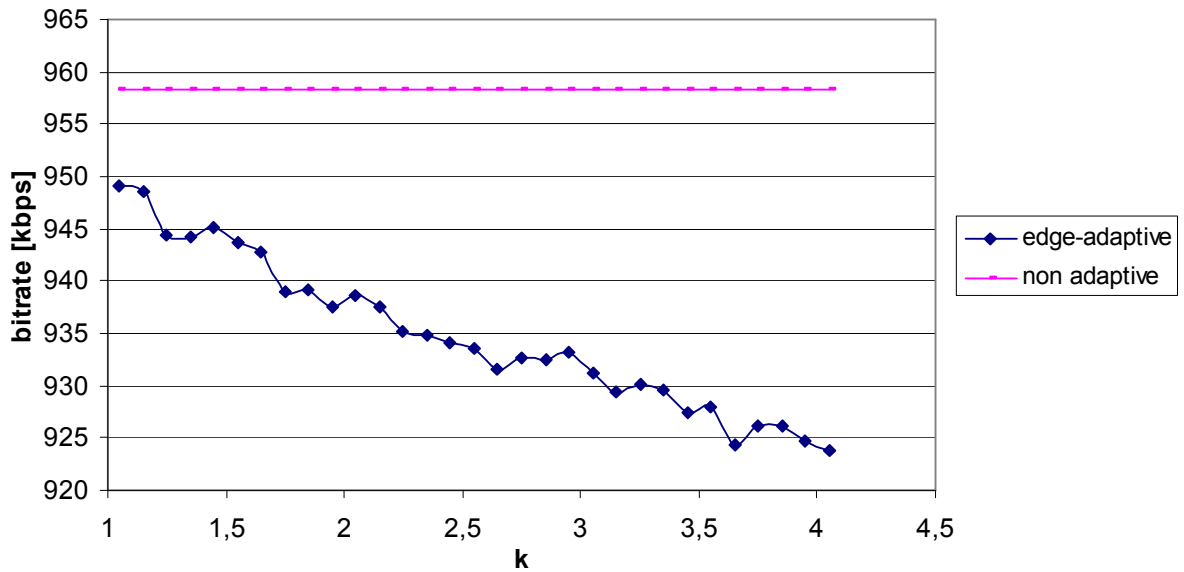


Fig. 6.20. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Result group 2:

Result for wide range search of motion vectors and 12th order decimation FIR filter.

This group of results shows the decrease of the bitrate for test sequences for low order decimation filter (12th order) and long motion vector search range. The search range was set up to 64 full pixel accuracy. The results in Fig. 6.21-24 show that using adaptive interpolation may give the bitrate gain between -0.16% up to 2.03%. For some k parameter the bitrate may be more than for non adaptive interpolation, but for most of cases the adaptive interpolation gives better results.

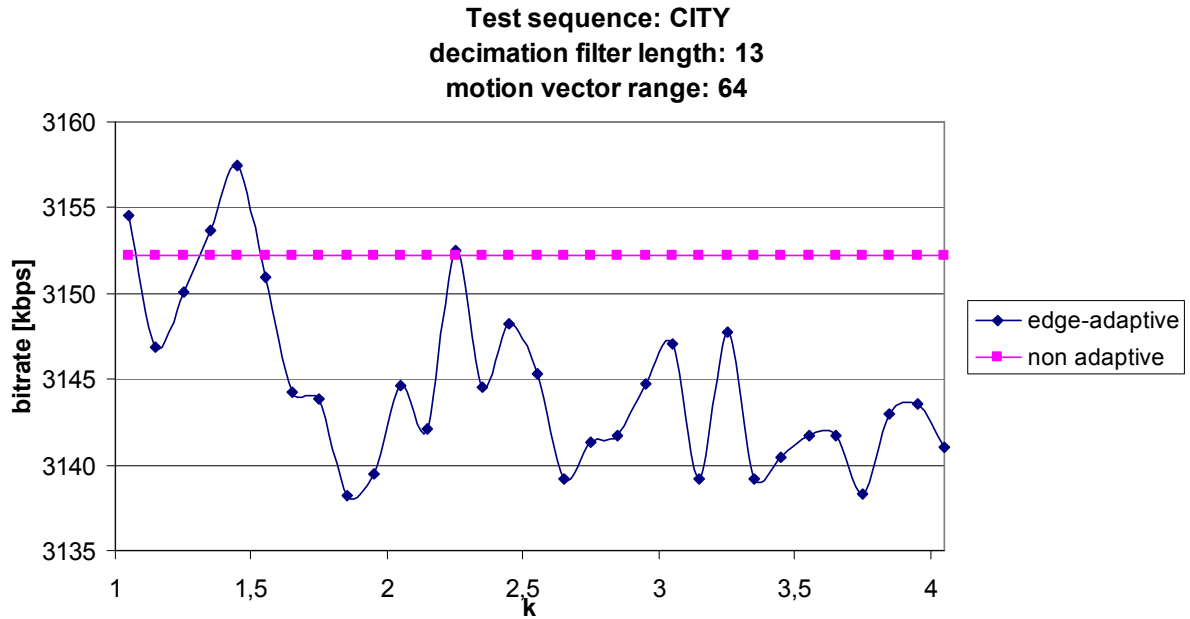


Fig. 6.21. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

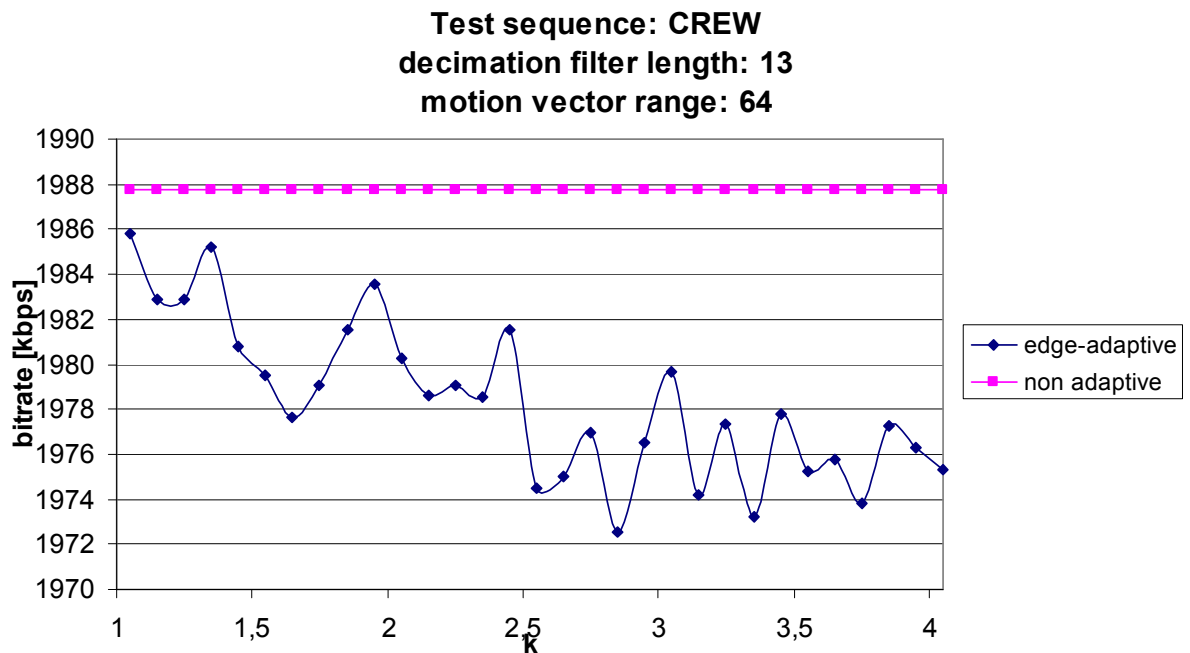


Fig. 6.22. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

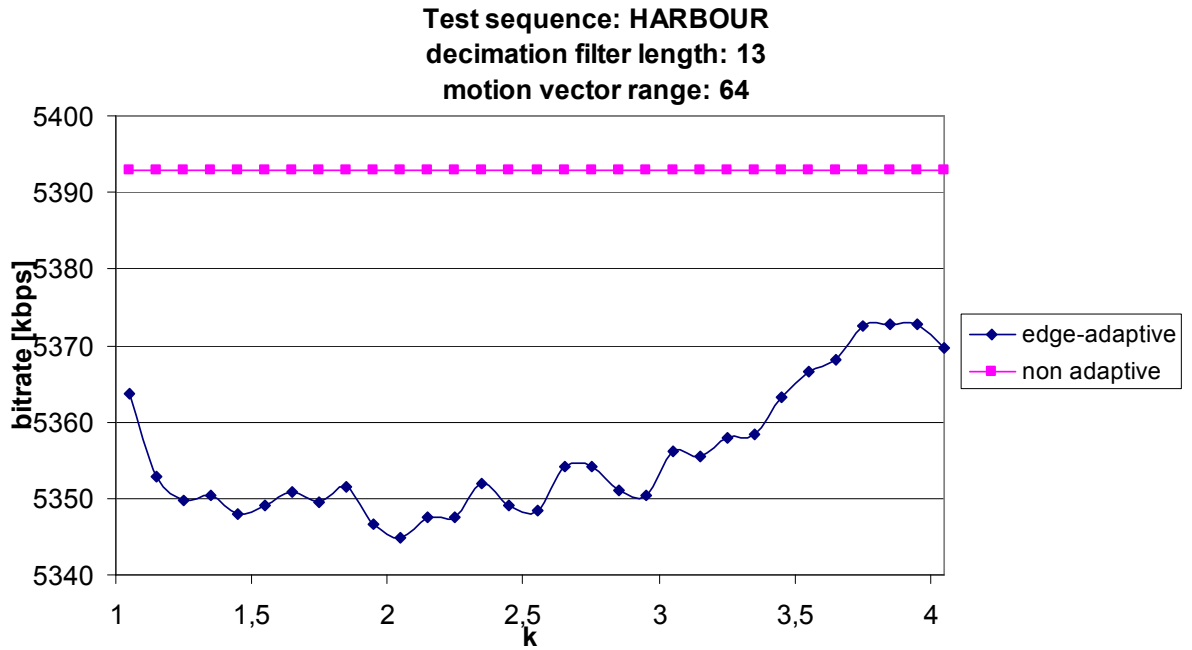


Fig. 6.23. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

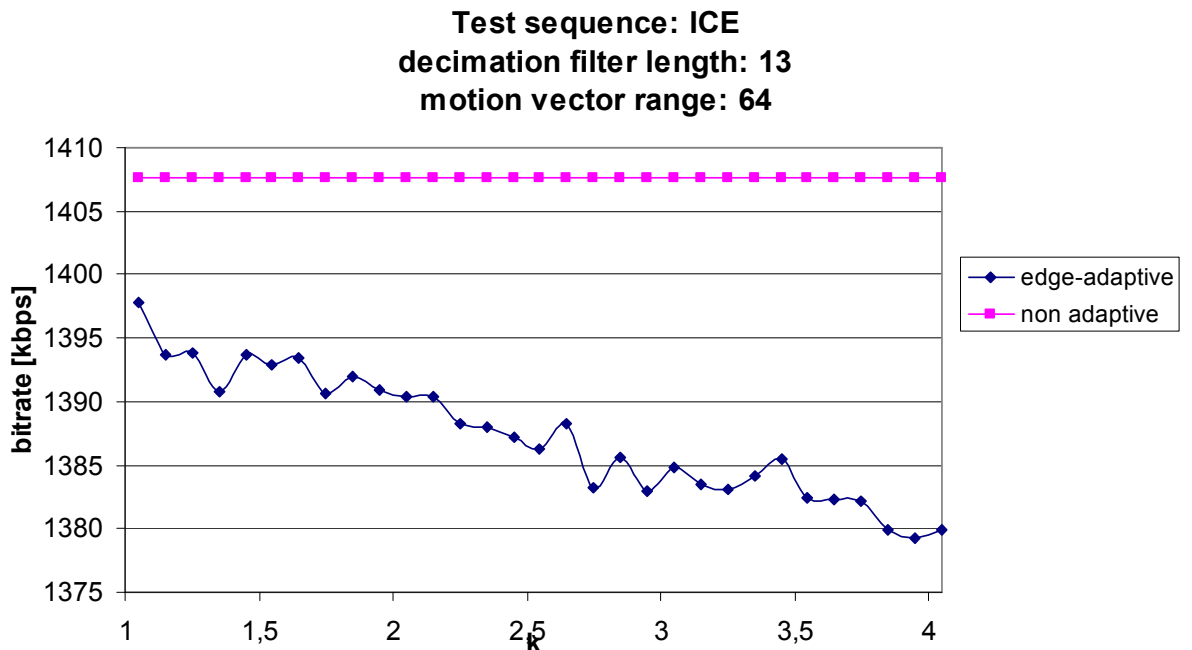


Fig. 6.24. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

Result group 3:

Result for short range search of motion vectors and 24th order decimation FIR filter.

This group of results shows the decrease of the bitrate for test sequences for high order decimation filter (24th order) and short motion vector search range. The search range was up to 4 full pixel accuracy. The results in Fig. 6.25-28 show that using adaptive interpolation may give the bitrate gain between 0,14% up to 3,72%.

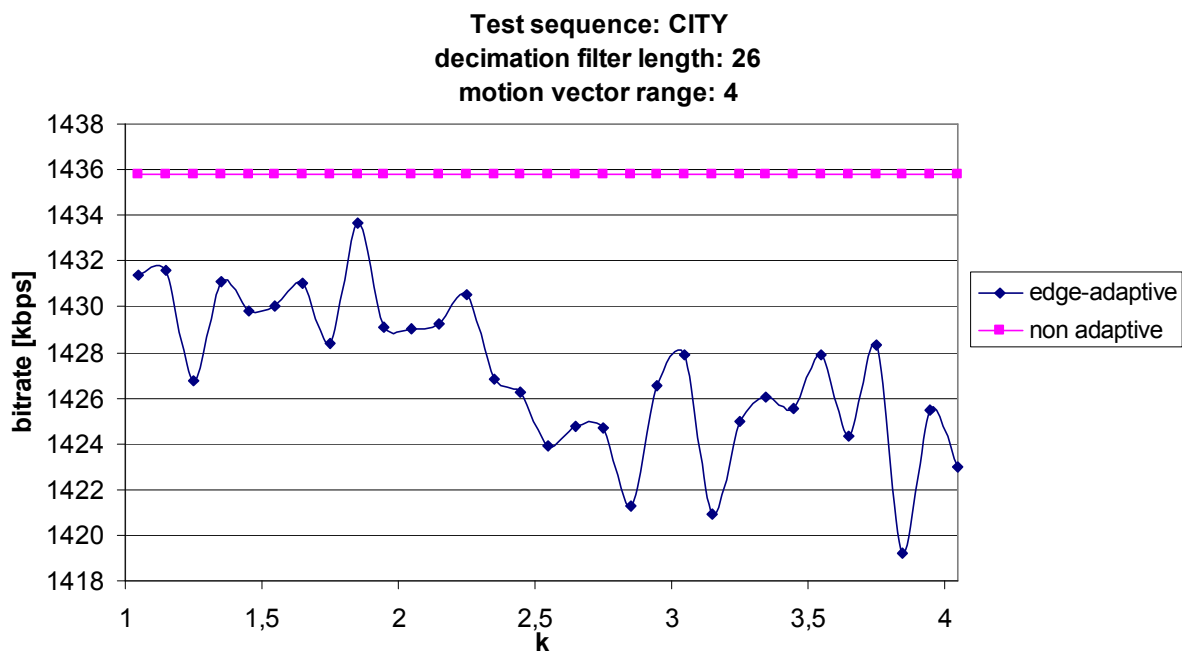


Fig. 6.25. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

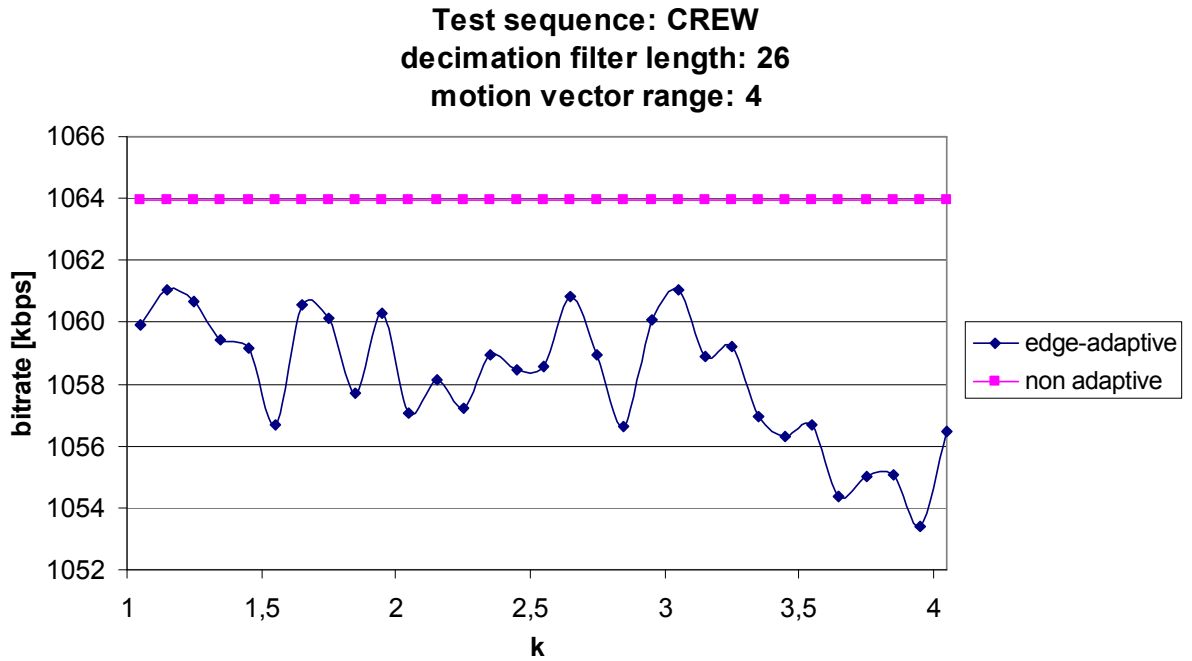


Fig. 6.26. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

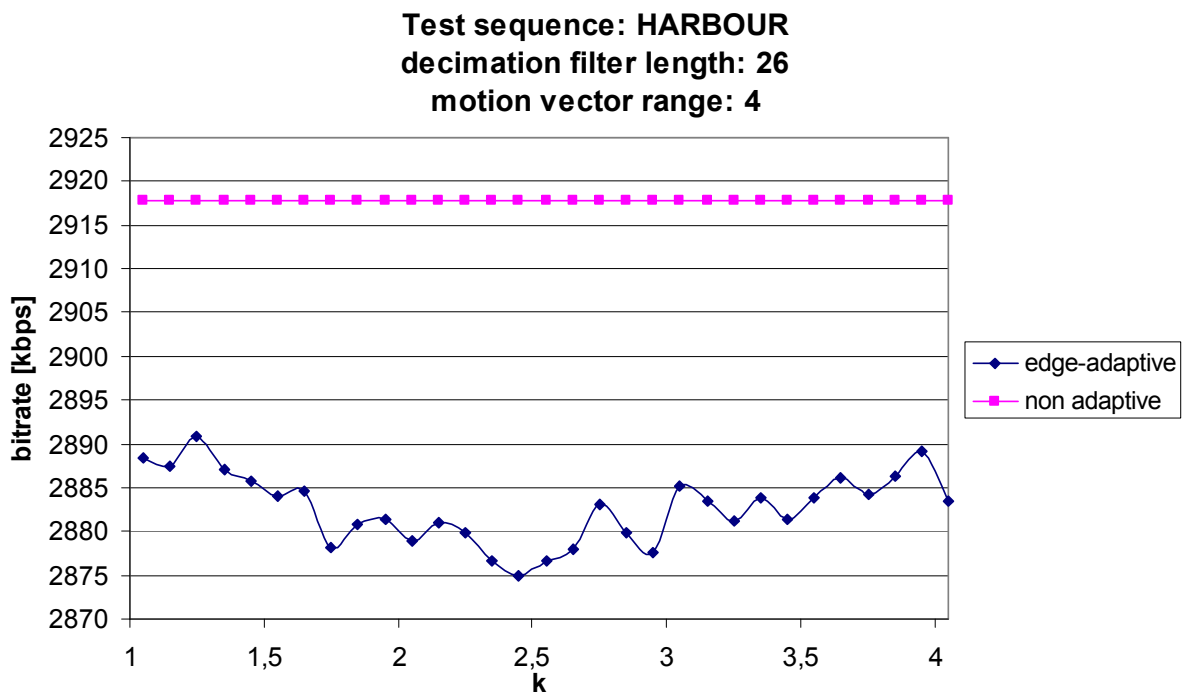


Fig. 6.27. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

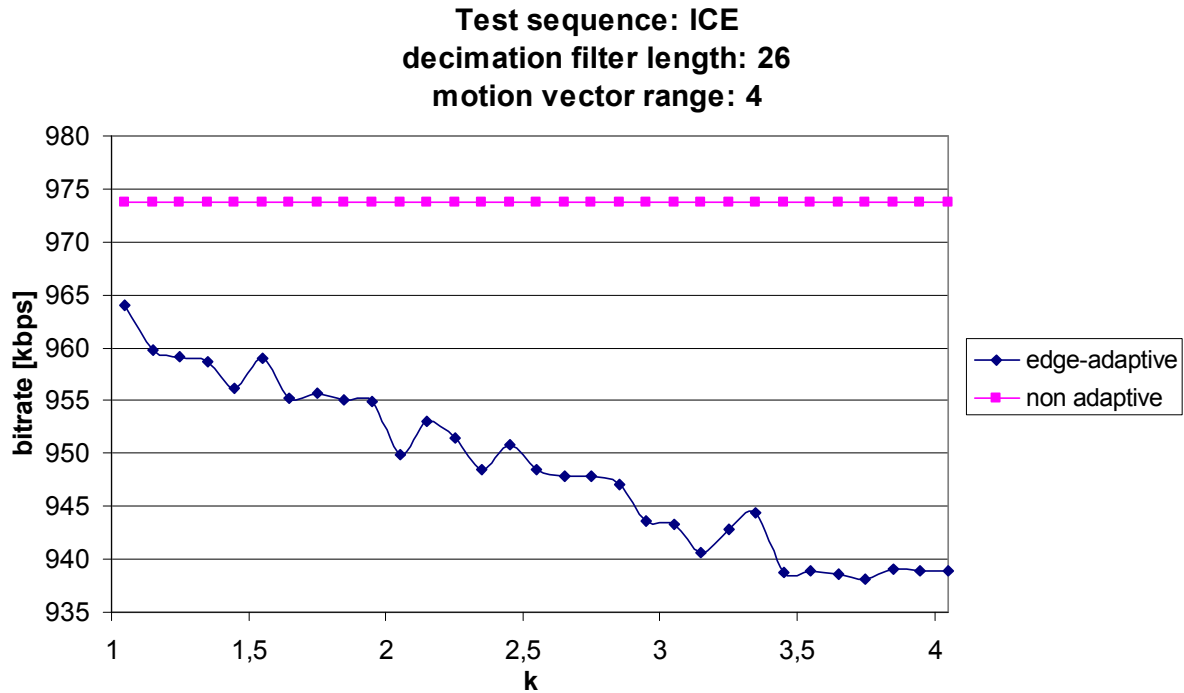


Fig. 6.28. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Result group 4:

Result for wide range search of motion vectors and 24th order decimation FIR filter.

This group of results shows the decrease of the bitrate for test sequences for low order decimation filter (24th order) and long motion vector search range. The search range was set up to 64 full pixel accuracy. The results in Fig. 6.29-32 show that using adaptive interpolation may give the bitrate gain between 0.0% up to 2.01%.

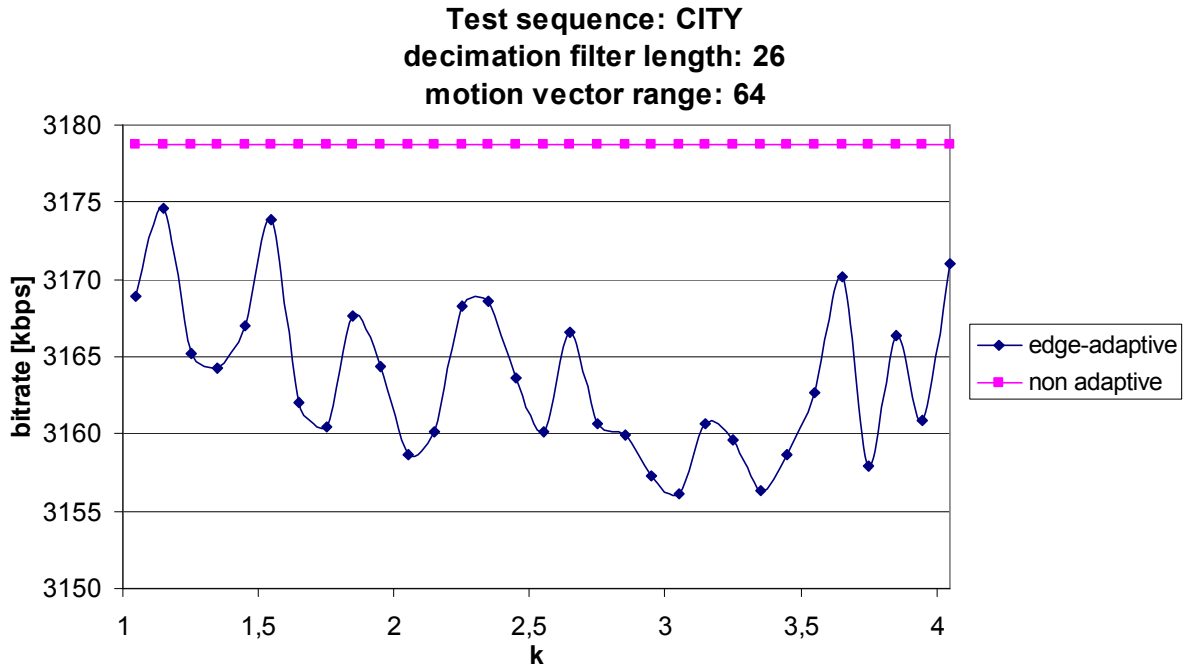


Fig. 6.29. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

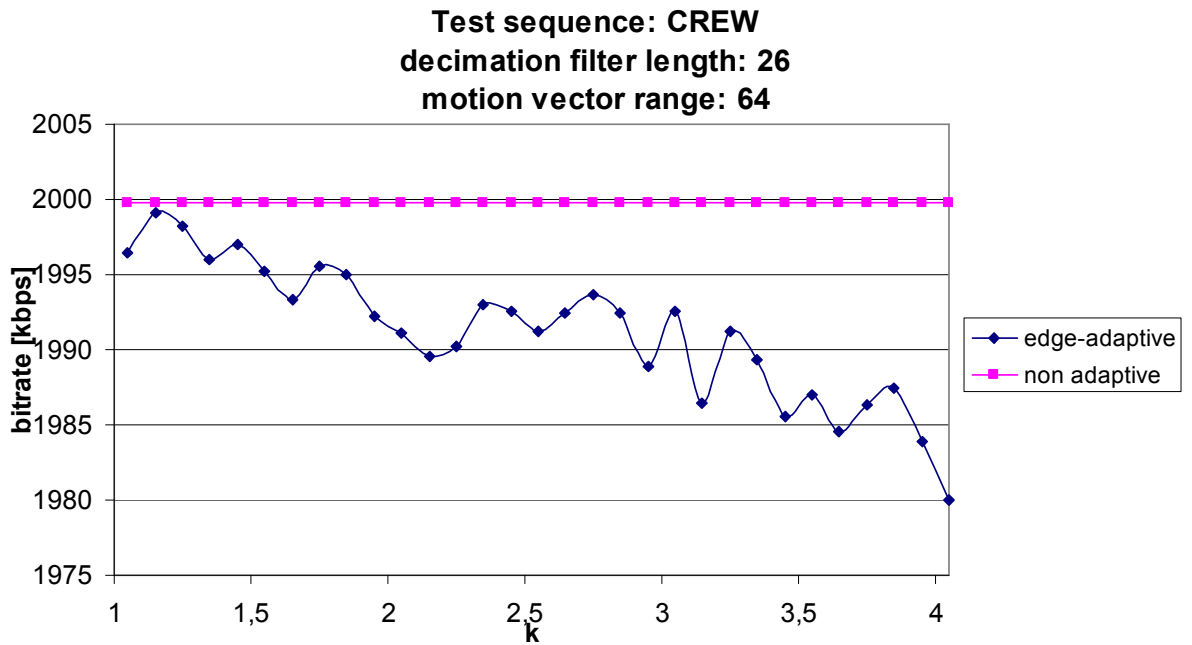


Fig. 6.30. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

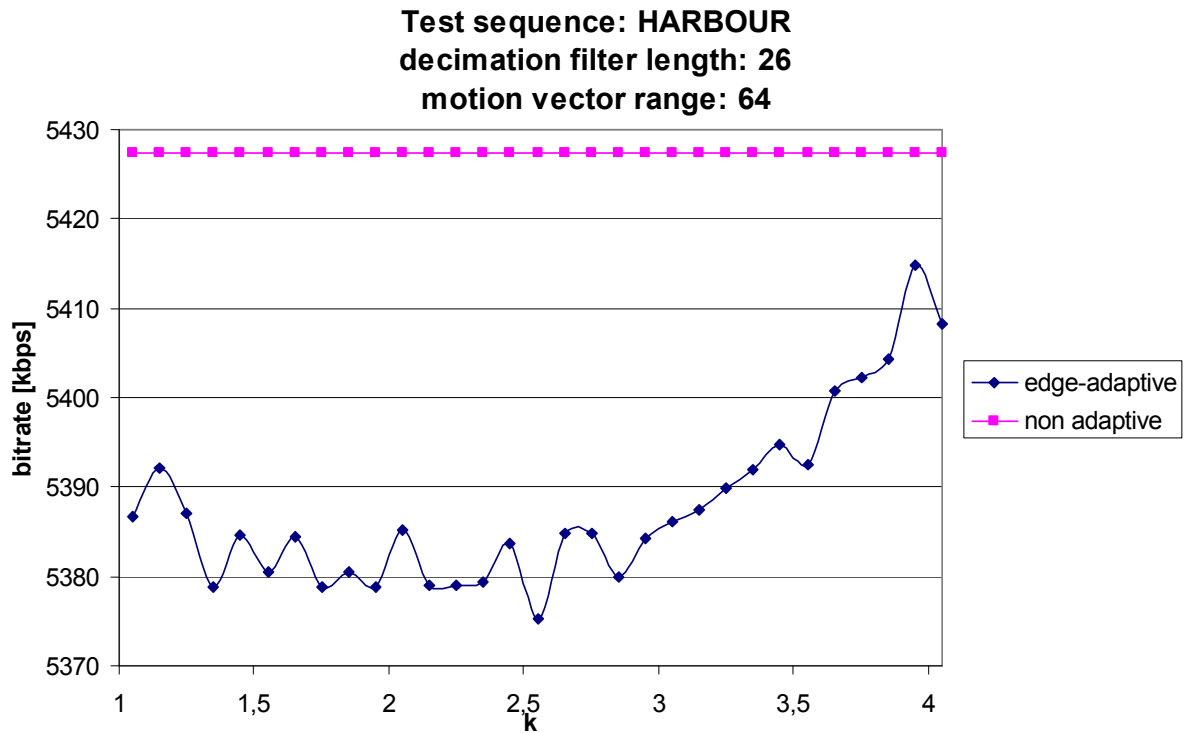


Fig. 6.31. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

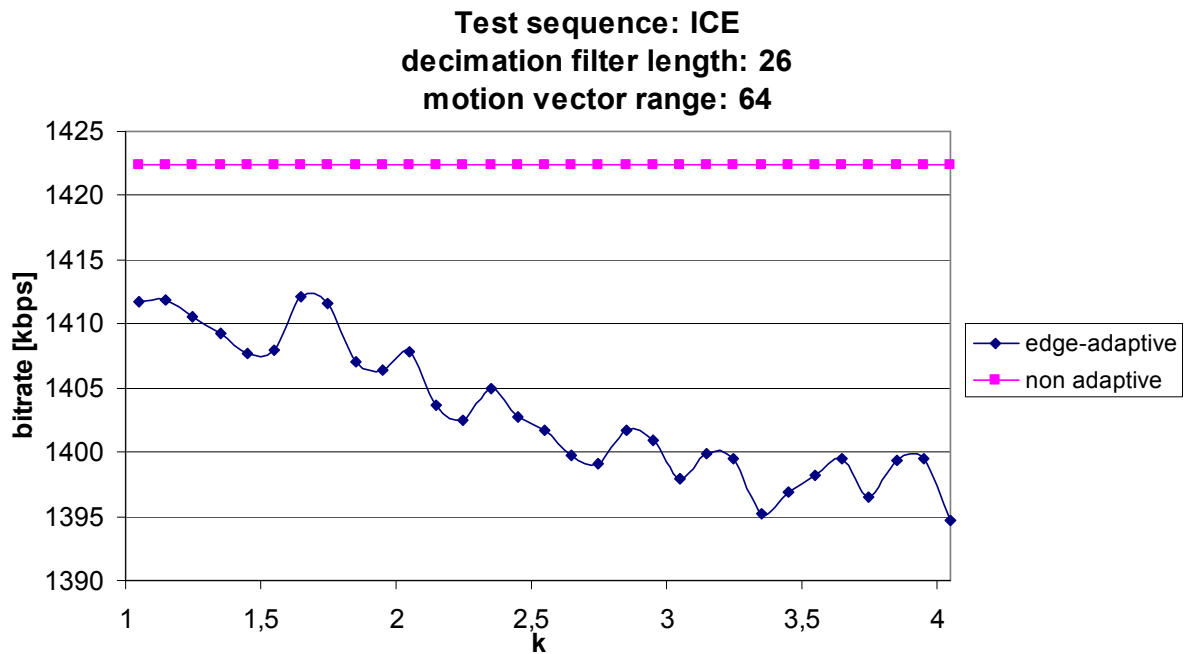


Fig. 6.32. Comparison of usage non-adaptive and edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

The average results:

The results for all test sequences for adaptive interpolation filter for a range of k parameter are compared to results for non-adaptive interpolation filter. On the Figs. 6.33-36, average results for four cases are shown:

- low order decimation filter and short motion vector range,
- low order decimation filter and long motion vector range,
- high order decimation filter and short motion vector range,
- high order decimation filter and long motion vector range.

**Interpolation filter test
decimation filter length: 13
motion vector range: 4**

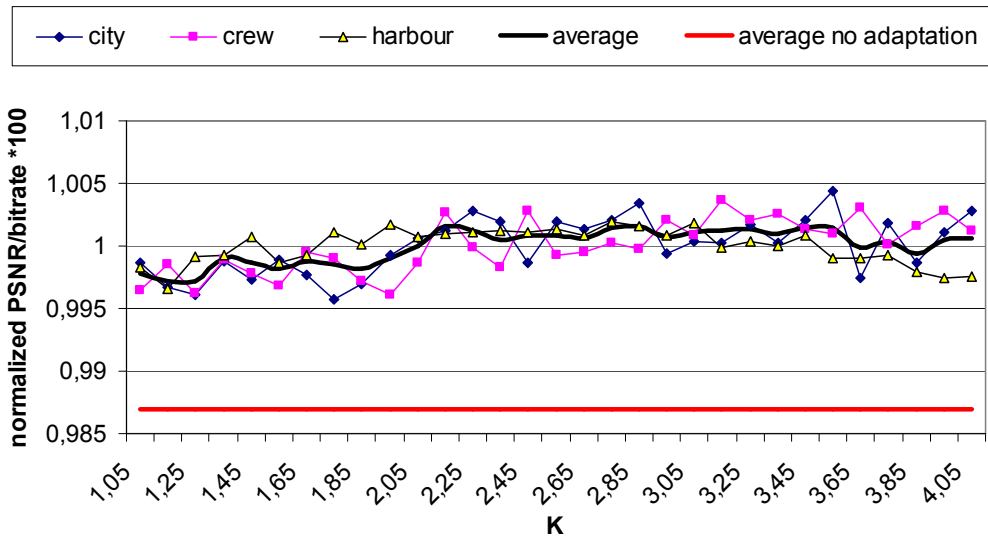


Fig. 6.33 Comparison average gain of PSNR/bitrate for edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Interpolation filter test
decimation filter length: 13
motion vector range: 64

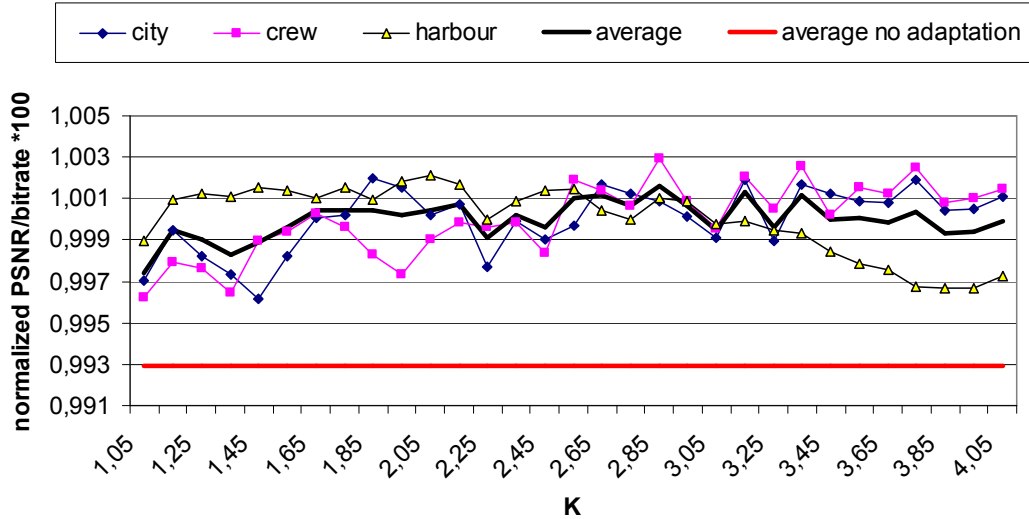


Fig. 6.34 Comparison average gain of PSNR/bitrate for edge-adaptive bi-cubic interpolation filter and 13 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

Interpolation filter test
decimation filter length: 26
motion vector range: 4

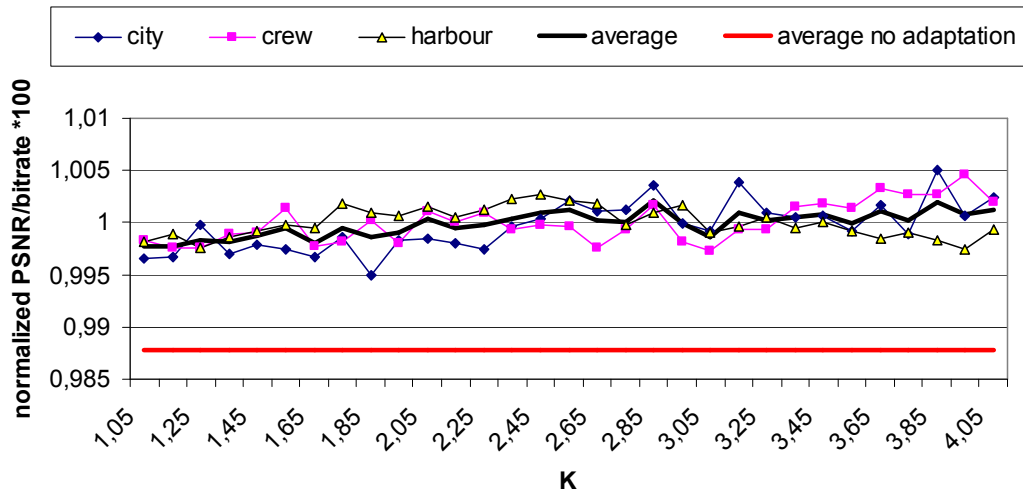


Fig. 6.35 Comparison average gain of PSNR/bitrate for edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 4.

Interpolation filter test
decimation filter length: 26
motion vector range: 64

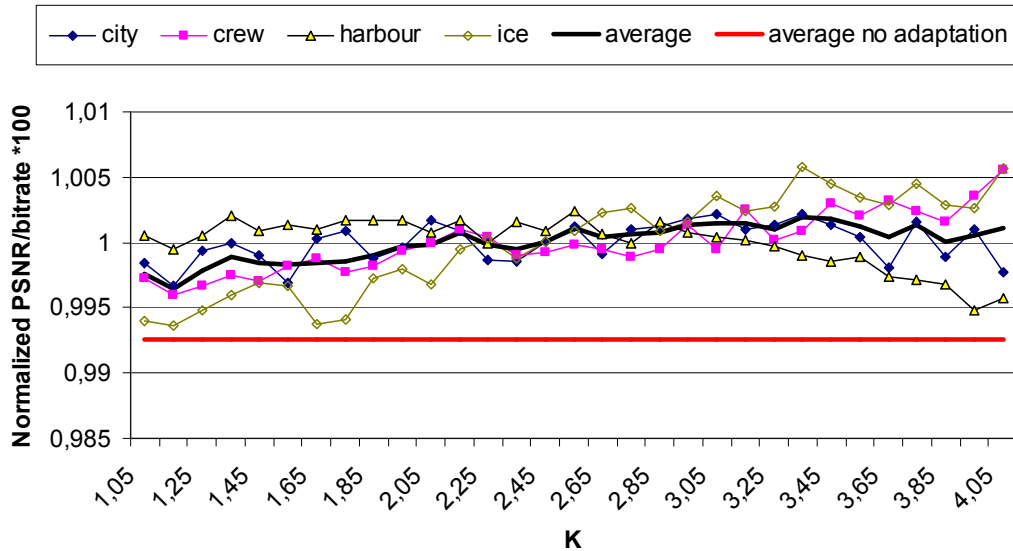


Fig. 6.36 Comparison average gain of PSNR/bitrate for edge-adaptive bi-cubic interpolation filter and 26 tap decimation FIR filter in scalable H.264 encoder, where motion vector search range was set to 64.

The results show that changing the parameter k , for interpolating filter, may significantly improve coding efficiency. The gain of encoding efficiency of H.264 scalable encoder when adaptive filter is used depends on the video content and the bitrate. Here, in this experiment this gain was even up to 3.4%. The parameter k , for which the average gain was the highest, is 2.85. For this parameter value the encoding efficiency gain oscillated between 0.35% up to 2.77%. The average encoding efficiency gain for all test sequences is 1.12%.

Chapter 7

Experimental assessment of the scalable video codecs

7.1. Introduction

In this Chapter descriptions of tests of the scalable video models are presented. There are two groups of experiments. The first group was performed in order to achieve objective assessments for encoding efficiency. And the second group of experiments was performed in order to achieve subjective assessments for encoding efficiency. The following models of scalable encoders have been tested:

- own implementation of H.263[ISO96] codec with spatio-temporal scalability,
- H.264[ISO-AVC] codec with spatio-temporal and quality scalability based on reference software JM 7.3,
- H.264 codec with spatio-temporal and quality scalability based on reference scalable model JSVM 1.0

7.2. Assessment using objective measure

For the group of objective tests, the PSNR measure has been chosen. The PSNR is commonly used for measuring the objective quality for image processing including

video coding techniques. This is not the best technique of measuring, but is commonly used; it is objective and easy to use. Because it is commonly used, it is easy to compare the obtained results to the results for other techniques.

7.2.1. Experiments for H.263 codec with spatio-temporal scalability

The aim of the experiment was to compare the encoding efficiency of a proposed scalable model of H.263 against the single layer non-scalable H.263 encoder. The scalable model of H.263 was prepared and the following test conditions were defined:

- CIF input sequences,
- open GOP structure: IPPP
- base layer was time decimated by factor 2,
- $\frac{1}{2}$ pixel motion estimation accuracy,
- motion vector search range was ± 15 ,
- interpolation and decimation filter was Johnston 12th order,

Table 7.1. Comparison of scalable and non-scalable H.263 encoder.

H.263 - based coder for CIF (352 × 288) sequences		<i>Football</i>	<i>Basket</i>	<i>Cheer</i>	<i>Fun</i>	<i>Bus</i>
Single-layer coder (H.263)	Bitstream [kbps]	428.64	422.14	402.33	424.86	414.61
	Average luminance PSNR [dB]	29.58	26.04	25.62	26.30	27.57
Proposed scalable coder	Low resolution layer bitstream [kbps]	99.27	103.17	98.28	103.89	99.70
	Low resolution layer average PSNR [dB] for luminance	26.76	24.64	23.19	24.17	26.06
	High resolution layer bitstream [kbps]	319.20	330.67	315.41	313.13	302.61
	Average PSNR [dB] for luminance recovered from both layers	29.55	26.02	25.58	26.29	27.59
	Bitrate overhead [%]	-2,37	2,77	2,82	-1,85	-2,97
Single-layer coder (H.263)	Bitstream [kbps]	875.77	775.02	692.31	831.36	790.55
	Average luminance PSNR [dB]	32.38	27.82	27.82	28.85	29.69
Proposed scalable coder	Low resolution layer bitstream [kbps]	190.22	194.22	179.08	197.55	190.99
	Low resolution layer average PSNR [dB] for luminance	29.32	26.68	25.31	26.61	28.37
	High resolution layer bitstream [kbps]	578.17	617.17	546.91	594.46	567.83
	Average PSNR [dB] for luminance recovered from both layers	32.42	27.83	27.83	28.85	29.68
	Bitrate overhead [%]	-12,26	4,69	4,86	-4,73	-4,01
Single-layer coder (H.263)	Bitstream [kbps]	1271.28	1139.21	998.14	1216.35	1121.57
	Average luminance PSNR [dB]	34.24	29.28	29.55	30.62	31.14
Proposed scalable coder	Low resolution layer bitstream [kbps]	282.83	287.39	260.14	292.54	283.68
	Low resolution layer average PSNR [dB] for luminance	31.21	28.42	27.06	28.51	30.25
	High resolution layer bitstream [kbps]	851.84	902.34	788.41	877.47	835.02
	Average PSNR [dB] for luminance recovered from both layers	34.27	29.28	29.59	30.62	31.15
	Bitrate overhead [%]	-10,75	4,43	5,05	-3,81	-0,26

As it can be noticed, the coding efficiency of a scalable model of H.263 strongly depends on the video content. For some sequences like FOOTBALL, FUN, BUS there may be a gain of encoding efficiency as compared to non-scalable H.263. But for other test sequences: BASKET, CHEER, there is a deterioration of encoding efficiency. Also the quality at given bitrate for scalable and non-scalable H.263 depends on the bitrate level.

Summarizing, even in the worst case, the price to pay which is the bitrate overhead for given quality is not so high. The promising results for simple H.263 codec suggest that for more complex group of encoders which are advanced video coders the introduction of the scalability may be possible at acceptable bitrate overhead for given quality.

7.2.2. Testing of H.264 codec with spatio-temporal scalability

7.2.2.1. Comparison to simulcast and non-scalable H.264

The scalable test model has been implemented as an extension of the AVC reference software version 7.3. Both the coder and the decoder have been implemented.

In order to test compression efficiency of spatial-temporal scalable encoder, the comparison to non-scalable and simulcast coding schemes has been performed. Bitrates of all coding scenarios have been compared for the same decoded video quality measured as PSNR of luminance for the full resolution output. The following coding parameters were set for all scenarios:

- CIF input sequences,
- three fixed sets of quantization parameters separately for I, P and B frames,
- 1/4 –pixel motion estimation in all layers,
- all prediction modes,
- CABAC entropy coder.

Table 7.2. Comparison the coding efficiency for scalable H.264 to simulcast technique and non-scalable H.264 encoder.

Test sequence	<i>Bus</i>		<i>Cheer</i>		<i>Football</i>		<i>Fun</i>		<i>Basket</i>	
Original frame rate [fps]	30		30		30		25		25	
$QP_I=10, QP_P=11, QP_B=12$										
	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]
Base layer	37.17	502.11	38.60	279.35	37.72	751.80	39.76	354.37	37.17	502.11
Enhancement layer	38.06	2215.86	39.06	1533.10	38.56	3030.11	40.75	1388.49	38.06	2215.86
Whole scalable codec	38.06	2717.97	39.06	1812.45	38.56	3781.91	40.75	1742.86	38.06	2717.97
Simulcast	38.02	3039.57	39.05	1939.98	38.54	4384.43	40.75	2081.16	38.02	3039.57
Nonscalable codec	38.02	2537.46	39.05	1660.63	38.54	3632.63	40.75	1726.79	38.02	2537.46
$QP_I=15, QP_P=16, QP_B=17$										
	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]
Base layer	32.96	286.13	34.65	151.94	33.51	457.17	36.20	200.74	32.96	286.13
Enhancement layer	34.08	1189.34	35.14	816.94	34.51	1702.18	37.31	759.71	34.08	1189.34
Whole scalable codec	34.08	1475.47	35.14	968.88	34.51	2159.35	37.31	960.45	34.08	1475.47
Simulcast	34.08	1668.77	35.16	1036.59	34.55	2559.36	37.45	1165.98	34.08	1668.77
Nonscalable codec	34.08	1382.64	35.16	884.65	34.55	2102.19	37.45	965.24	34.08	1382.64
$QP_I=20, QP_P=21, QP_B=22$										
	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]
Base layer	29.06	156.33	31.03	77.85	29.65	261.20	33.12	109.05	29.06	156.33
Enhancement layer	30.28	645.36	31.52	434.19	30.76	955.72	34.17	420.95	30.28	645.36
Whole scalable codec	30.28	801.69	31.52	512.04	30.76	1216.92	34.17	530.00	30.28	801.69
Simulcast	30.33	913.55	31.63	544.96	30.89	1466.69	34.39	651.76	30.33	913.55
Nonscalable codec	30.33	757.22	31.63	467.11	30.89	1205.49	34.39	542.71	30.33	757.22

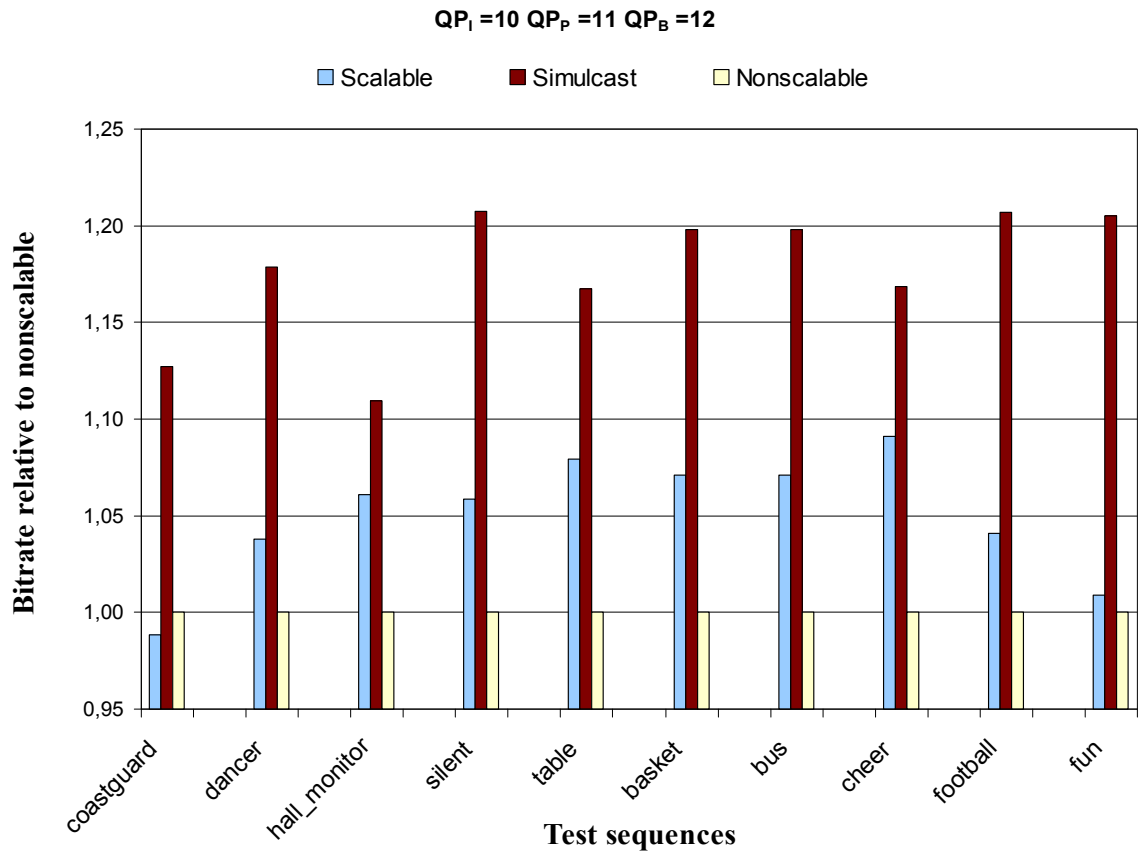


Fig. 7.1 Approximate bitrate comparison for scalable, nonscalable and simulcast coding.

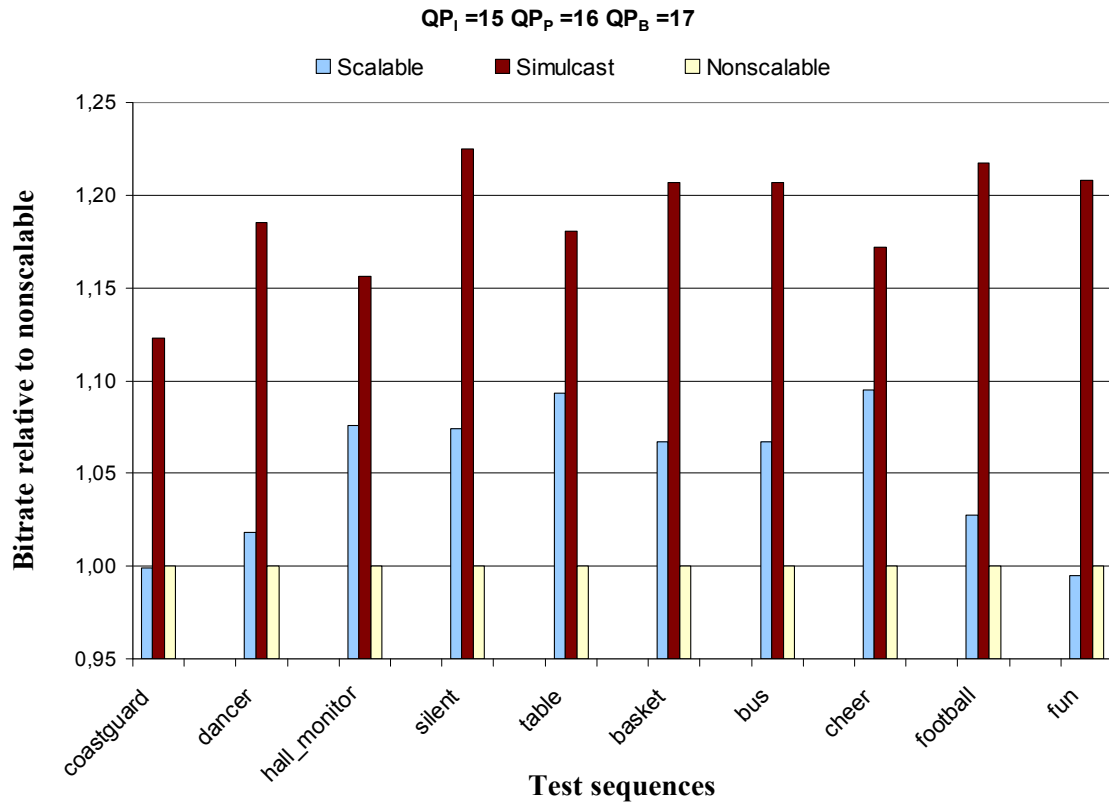


Fig. 7.2 Approximate bitrate comparison for scalable, nonscalable and simulcast coding.

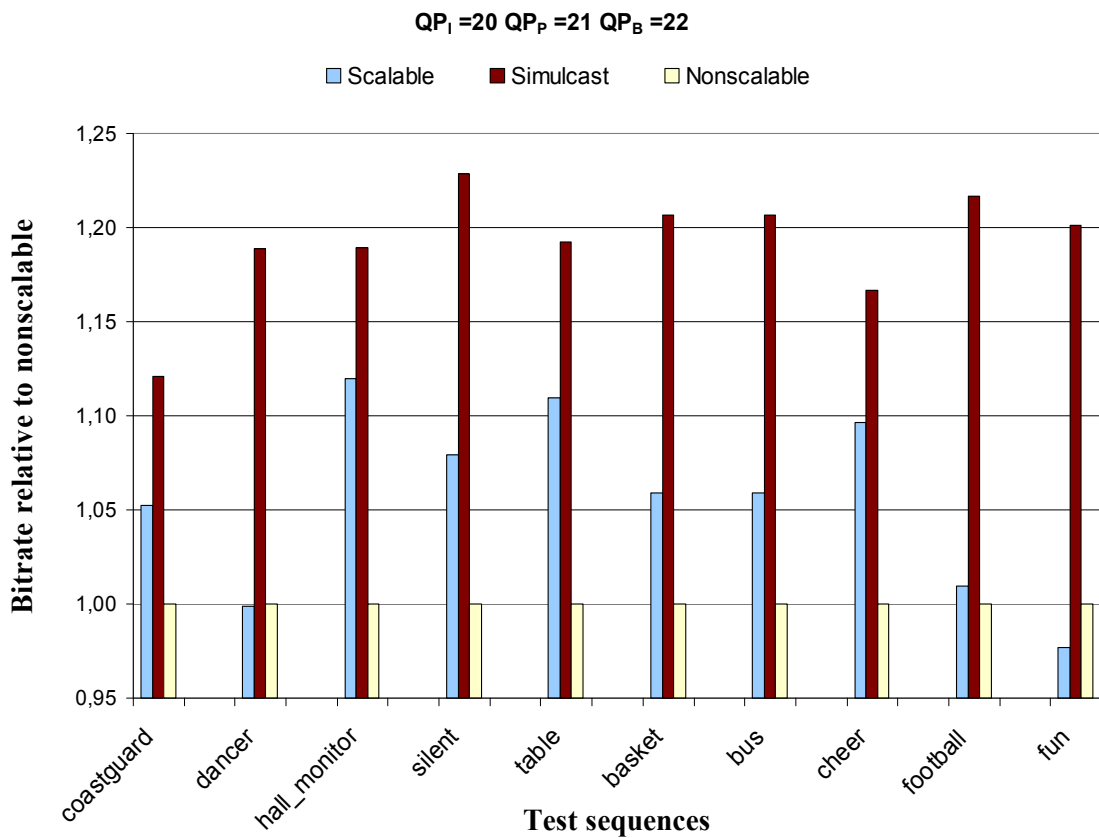


Fig. 7.3 Approximate bitrate comparison for scalable, nonscalable and simulcast coding.

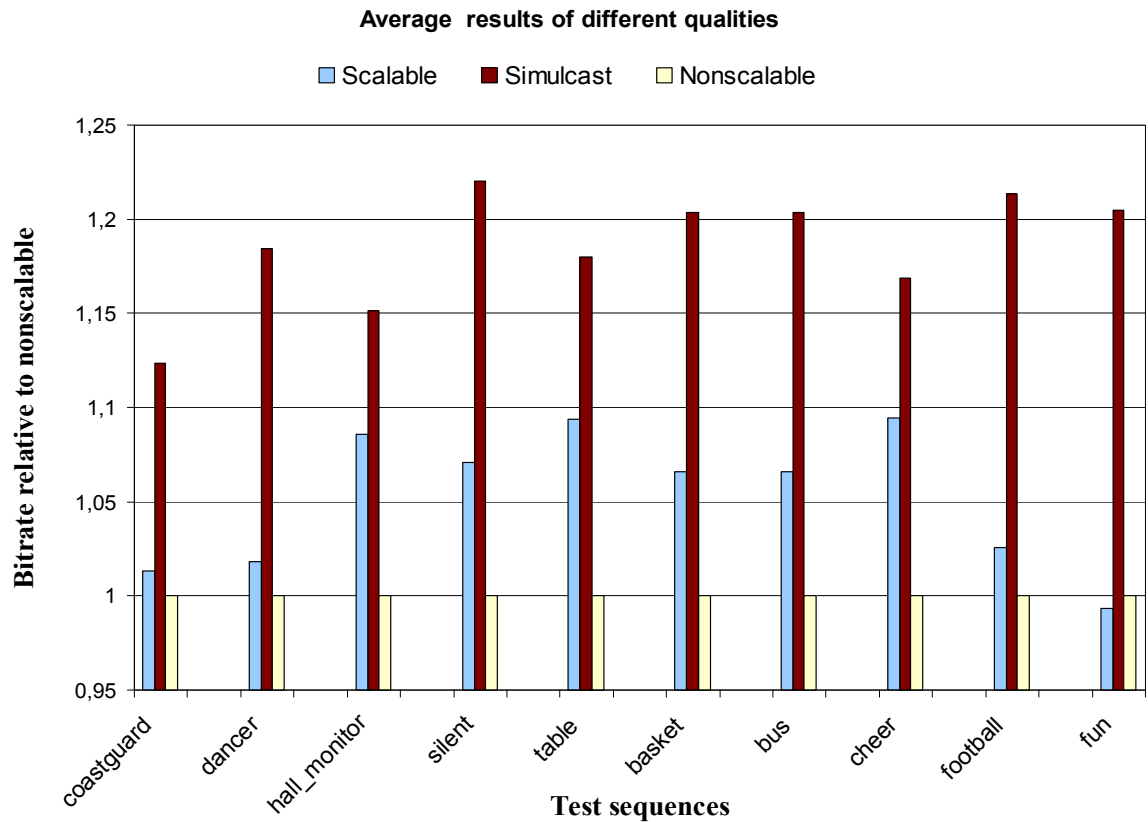


Fig. 7.4 Average bitrate comparison for scalable, nonscalable and simulcast coding.

The average bitrate overhead of scalable coding, in comparison to non-scalable (single layer) coding technique, is between -1.0% up to 10% depending on the encoded video content. The average bitrate overhead for all sequences is 5.2% for the scalable coding and 18.5% for the simulcast technique.

7.2.2.2. Comparison to encoder proposed by the MPEG: JSVM 2.0

The aim of this experiment was to compare the coding efficiency of the scalable model proposed by the author against the model based on the technique proposed later [ISO-JSVM] by the MPEG community. Each test sequence was compared at three different bitrate levels. The experiment conditions were the following:

- sequence length was 120 frames,
- two layers of scalable encoding,
- enhancement layer: CIF,
- base layer: QCIF,

- entropy coder: CABAC,
- motion estimation accuracy: $\frac{1}{4}$ pixel,
- GOP (open GOP) structure for proposed codec:
 - enhancement layer: IBBBPBBB,
 - base layer: IBPBPB,
- MCTF length filtering for MPEG scalable model:
 - enhancement layer: 4,
 - base layer: 2.

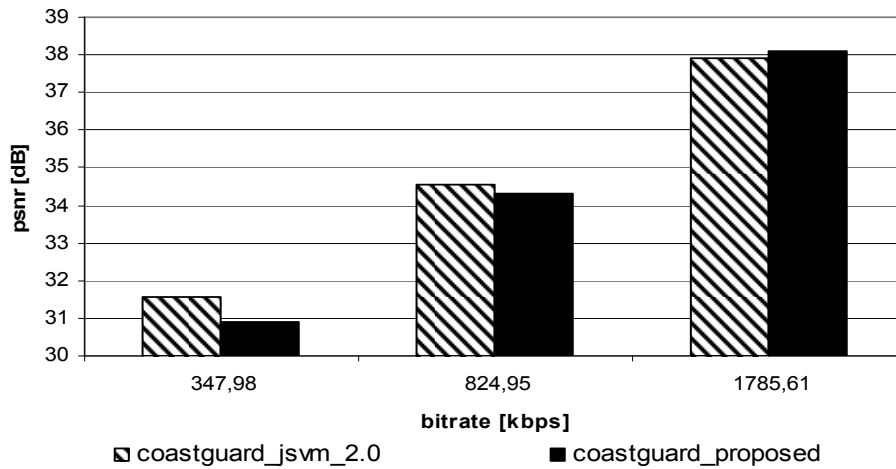


Fig. 7.5 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – COASTGUARD test sequence.

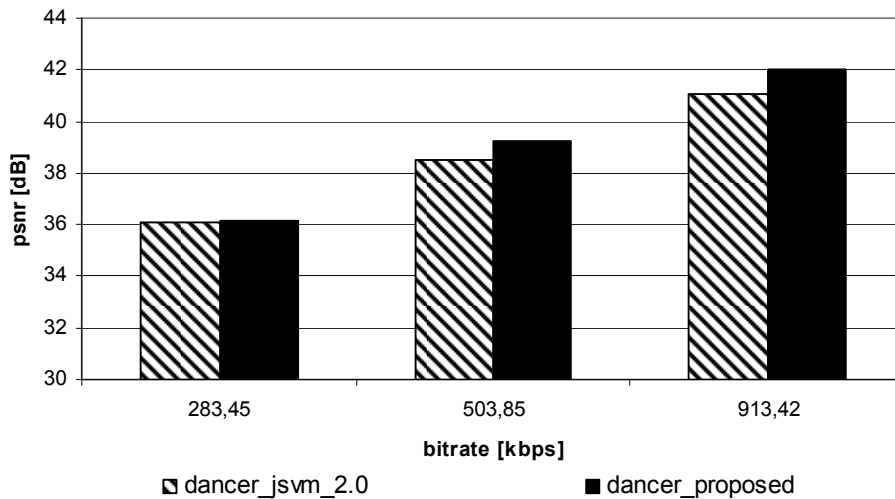


Fig. 7.6 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – DANCER test sequence.

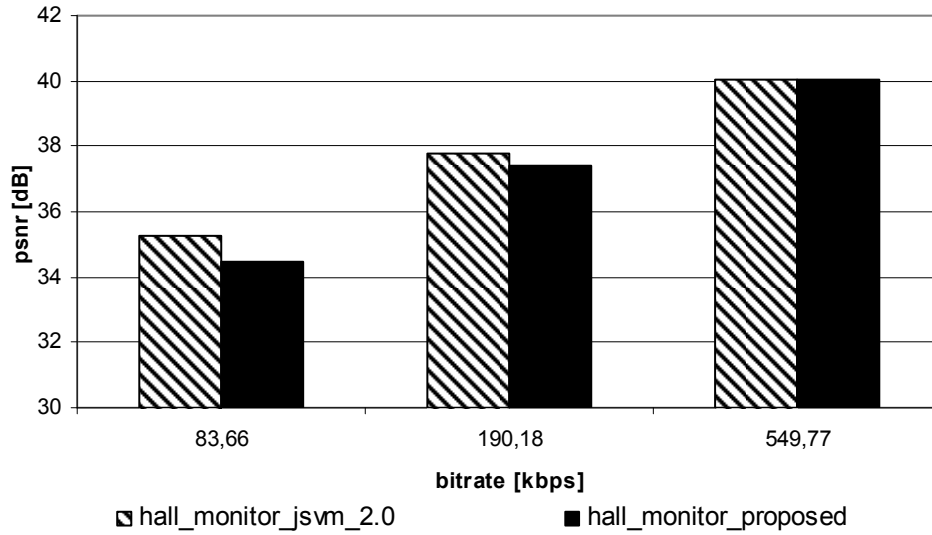


Fig. 7.7 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – HALL_MONITOR test sequence.

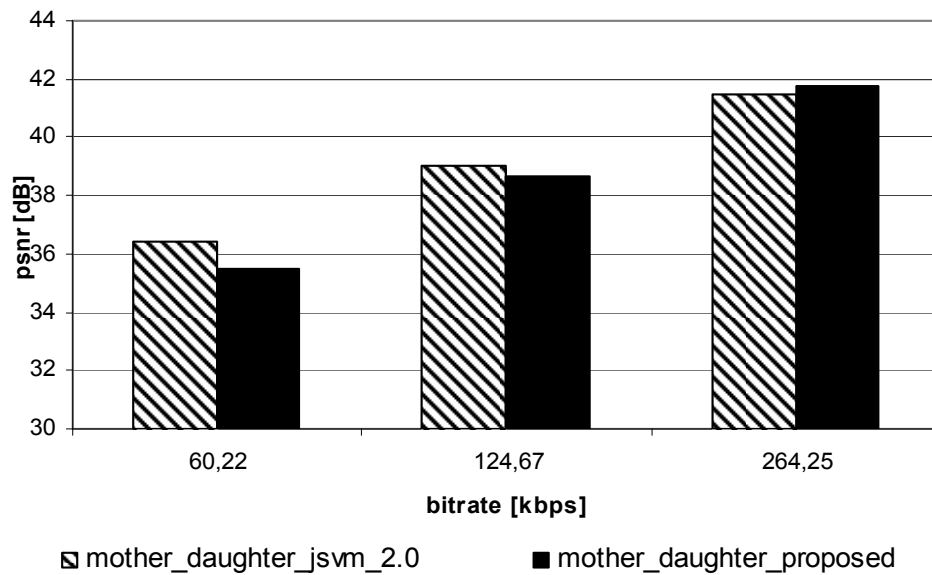


Fig. 7.8 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – MOTHER_DAUGHTER test sequence.

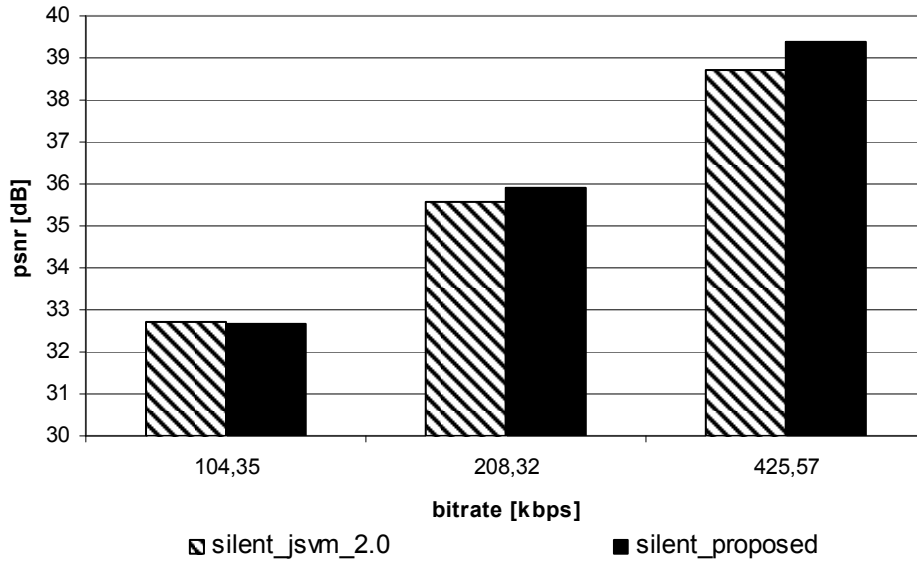


Fig. 7.9 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – SILENT test sequence.

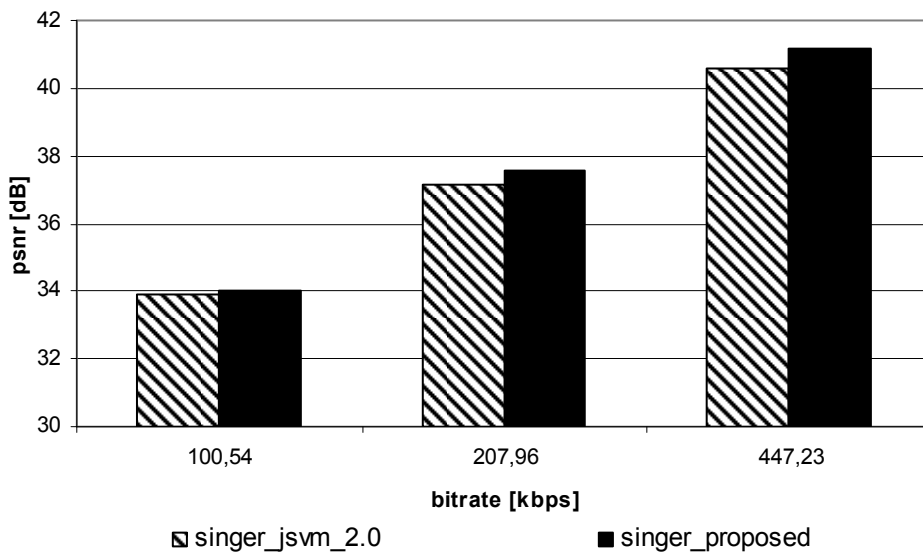


Fig. 7.10 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – SINGER test sequence.

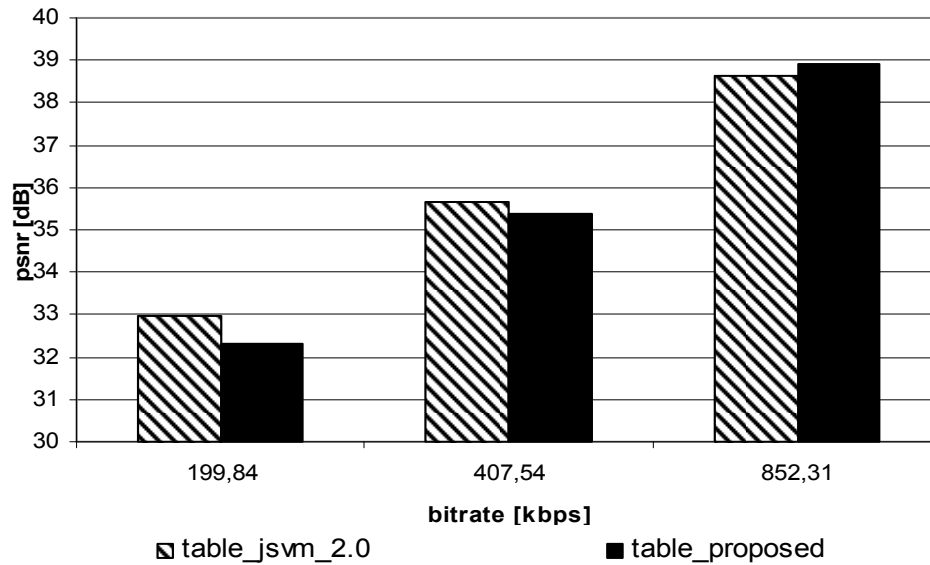


Fig. 7.11 Comparison of proposed scalable H.264 encoder and JSVM 2.0 encoder – TABLE test sequence.

The average quality loss for proposed scalable H.264 encoder against the JSVM 2.0 model is 0.1 %. Thus, the results for both encoders are almost the same. The encoding efficiency depends on the bitrate and on the content of encoded video sequence. For some cases (see Fig. 7.6, 7.9, 7.10), the proposed encoder produces bitstream more efficient, for some (see Fig. 7.7) it is less efficient. For other test sequences, the coding efficiency for both codecs changes depending on the bitrate and for some bitrates the JSVM 2.0 model is more efficient and for other bitrates the proposed codec is more efficient. The results show that the coding efficiency for various coding techniques may be similar. Encoder efficiency depends on the bitrate and input video sequence content. The fact is that for different video sequences once the codec proposed by the author has better efficiency and another time the JSVM 2.0 (that was proposed later) has better coding efficiency means that various techniques may be better tuned for different input video sequence content. But on average both techniques have similar coding efficiency.

7.2.3. Experiments: comparison of non-scalable H.264 with raster scan and with spiral scan

7.2.3.1. Intra-frame coding test results

The verification process of the spiral scan was partitioned into two parts. One was a test of the intra prediction and the other one was to test the inter prediction. To verify the coding process for intra prediction the encoder was set with the following parameters:

- the input sequence was CIF (352×288) and 4CIF (704×576),
- the frame rate of the input sequence was 30 Hz,
- there was no rate control, for each input sequence four different quantization parameters (QP) was set : 20, 30, 40, 50,
- all frames was set to be encoded as IDR (instantaneous decoding refresh) frames,
- number of frames was 100.

Results of the experiment for input sequences in 4CIF size are shown in Table 7.3. and for CIF in Table 7.4.

Table 7.3. Comparison of coding efficiency for various video image orientation for spiral scan and for raster scan (4CIF test sequences).

Sequence	Landscape raster scan		Portrait raster scan		Landscape spiral scan		Landscape spiral scan FMO	
	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]
city	403,67	23,95	402,98	23,96	402,32	23,96	495,37	23,92
	2335,61	28,88	2319,92	28,88	2331,64	28,88	2614,47	28,88
	8457,90	35,34	8451,25	35,34	8420,10	35,34	9000,97	35,32
	21999,04	42,96	22022,43	42,96	21942,13	42,96	22871,09	42,94
crew	354,63	28,16	348,56	28,16	359,03	28,13	492,84	27,97
	1036,49	32,74	1027,15	32,75	1037,78	32,75	1341,94	32,73
	3675,42	37,70	3661,74	37,70	3651,97	37,69	4305,54	37,66
	13406,57	43,63	13387,34	43,65	13329,72	43,63	14354,30	43,61
harbour	637,98	23,00	641,11	23,02	644,38	23,01	849,69	23,00
	2695,72	29,20	2697,78	29,21	2696,57	29,20	3137,99	29,16
	8237,00	35,67	8244,33	35,65	8221,90	35,68	8981,45	35,65
	21198,14	42,91	21233,68	42,91	21175,67	42,91	22330,27	42,90
soccer	340,32	25,95	347,30	25,95	351,75	25,96	535,64	25,84
	1479,60	29,95	1505,53	29,94	1500,53	29,95	1887,74	29,93
	6664,05	36,03	6778,51	36,04	6703,63	36,02	7383,02	36,01
	18105,87	43,53	18366,76	43,54	18170,60	43,52	19190,23	43,50

Table 7.4. Comparison of coding efficiency for various video image orientation for spiral scan and for raster scan (CIF test sequences).

Sequence	Landscape raster scan		Portrait raster scan		Landscape spiral scan		Landscape spiral scan FMO	
	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]
city	94,77	23,71	94,94	23,71	94,19	23,70	114,16	23,67
	620,68	28,02	622,42	28,02	625,28	28,03	684,66	28,02
	2528,78	34,53	2532,23	34,53	2529,38	34,53	2657,90	34,51
	6622,42	42,68	6634,05	42,69	6622,46	42,69	6832,66	42,65
crew	123,94	26,61	122,65	26,61	127,38	26,57	166,83	26,50
	385,87	31,23	386,10	31,24	389,55	31,24	473,90	31,23
	1408,85	37,03	1410,05	37,03	1409,07	37,03	1587,45	37,00
	4114,21	43,79	4110,67	43,79	4097,04	43,79	4387,41	43,76
harbour	216,10	21,27	217,56	21,28	221,05	21,27	271,13	21,27
	1035,73	27,02	1038,53	27,03	1039,52	27,03	1154,73	27,01
	3356,18	34,08	3350,35	34,07	3355,25	34,08	3546,03	34,07
	7834,13	42,50	7832,12	42,50	7833,86	42,50	8110,00	42,48
soccer	115,79	25,38	117,37	25,35	120,25	25,37	173,04	25,27
	434,53	29,57	440,48	29,57	442,75	29,58	546,11	29,56
	1915,40	34,97	1947,07	34,98	1934,47	34,97	2117,52	34,96
	5541,40	42,86	5603,53	42,87	5565,88	42,86	5837,52	42,83

In the above tables the results of four experiments are presented. In the first experiment the standard, not modified, encoder has been tested. As the input four test sequences: CITY, CREW, HARBOUR and SOCCER have been taken. The results are shown in the first column.

In the second experiment the same encoder as before has been taken, but the input test sequences have been rotated by 90°. In this case the input sequence size was 576×704. The rotation was made in order to check whether the frame orientation influences the coding efficiency. If it does, it means that coding efficiency for the AVC/H.264 encoder depends on the direction of processing.

In the third experiment the encoder modified by the author was used. The spiral scan was set on. The aspect ratio for the spiral scan of macroblocks was set to 4:3 and the starting point for this scan was in the center of the image. The input test sequences used was the same as for the first experiment.

For the last experiment the standard encoder was used, but the FMO (flexible macroblock order) was switched on where the order of macroblock was set to the spiral order proposed by the author. In such a case the encoder during encoding process uses standard coding algorithms and so it is not as efficient as for the standard, the raster scan of macroblocks. The input sequences were the same as for the first and third experiment.

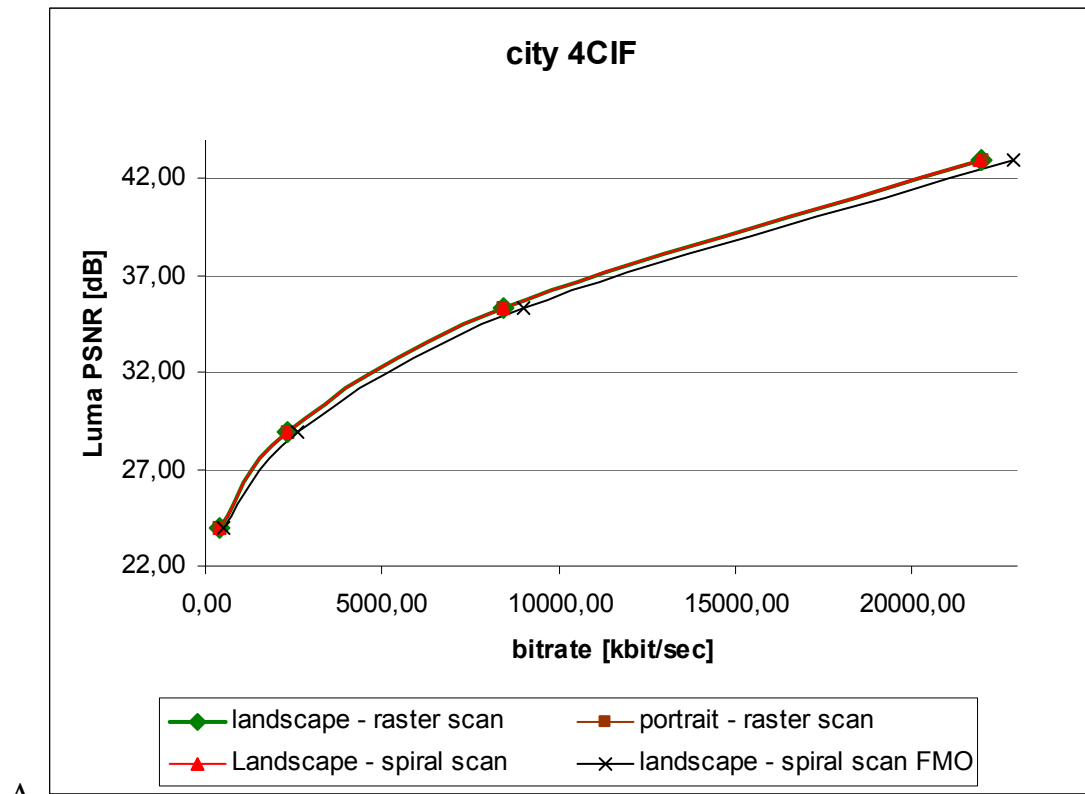
As it can be noticed in the Table 7.3, Table 7.4, Fig. 7.12 A-D and Fig. 7.13 A-D the first three experiments have almost the same bitrate at the same quality level. The differences in the bitrate are very slight. In the case of the spiral scan with FMO the bitrate overhead is quite huge in low bitrates and decreases when QP decreases. The Table 7.5 and Table 7.6 show the bitstream differences in comparison to the standard coding process in percents.

Table 7.5. Bitrate overhead for various macroblock scans (4CIF sequences).

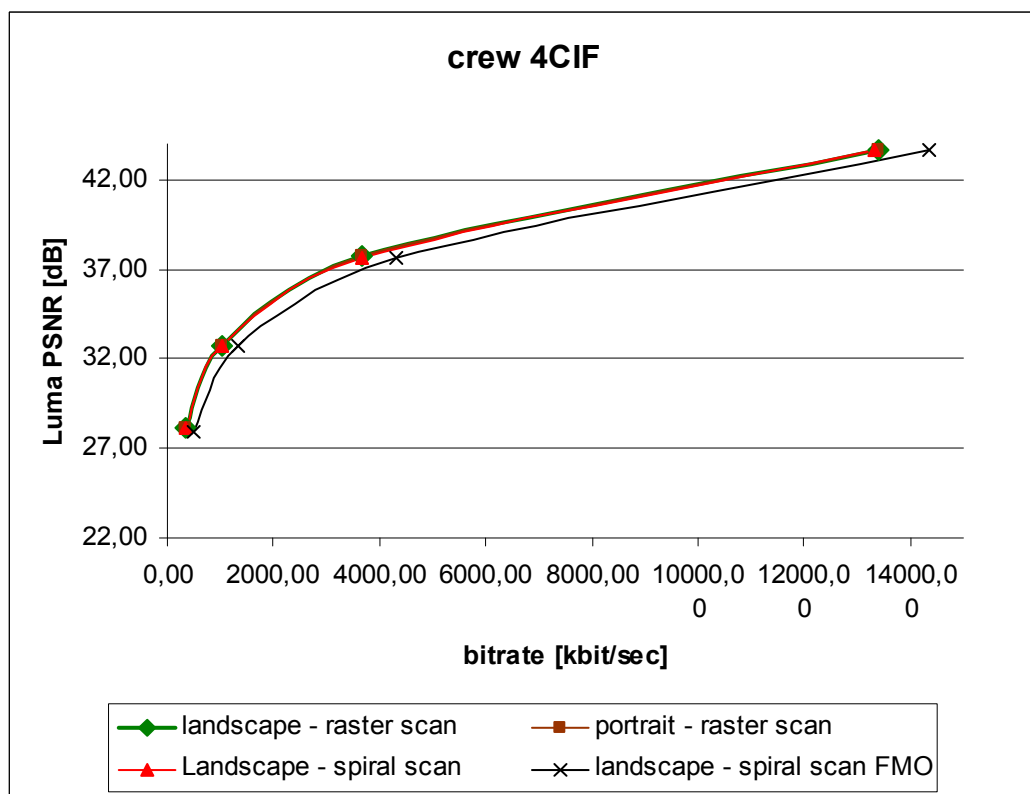
Sequence	Bitrate [kbps]	Portrait raster scan overhead [%]	Landscape spiral scan overhead [%]	Landscape spiral scan FMO overhead [%]
city	94,77	-0,17	-0,33	23,13
	620,68	-0,67	-0,17	12,13
	2528,78	-0,08	-0,45	6,90
	6622,42	0,11	-0,26	4,23
	average	-0,2025	-0,3025	11,5975
crew	123,94	-1,71	1,24	37,27
	385,87	-0,90	0,12	29,31
	1408,85	-0,37	-0,64	17,90
	4114,21	-0,14	-0,57	7,69
	average	-0,78	0,0375	23,0425
harbour	216,10	0,49	1,00	31,86
	1035,73	0,08	0,03	16,37
	3356,18	0,09	-0,18	9,24
	7834,13	0,17	-0,11	5,45
	average	0,2075	0,185	15,73
soccer	115,79	2,05	3,36	52,28
	434,53	1,75	1,41	25,80
	1915,40	1,72	0,59	10,13
	5541,40	1,44	0,36	5,61
	average	1,74	1,43	23,455

Table 7.6. Bitrate overhead for various macroblock scans (CIF sequences).

Sequence	Bitrate [kbps]	Portrait raster scan overhead [%]	Landscape spiral scan overhead [%]	Landscape spiral scan FMO overhead [%]
city	94,77	0,18	-0,61	20,46
	620,68	0,28	0,74	10,31
	2528,78	0,14	0,02	5,11
	6622,42	0,18	0,00	3,17
	average	0,195	0,0375	9,7625
crew	123,94	-1,04	2,78	34,61
	385,87	0,06	0,95	22,81
	1408,85	0,09	0,02	12,68
	4114,21	-0,09	-0,42	6,64
	average	-0,245	0,8325	19,185
harbour	216,10	0,68	2,29	25,47
	1035,73	0,27	0,37	11,49
	3356,18	-0,17	-0,03	5,66
	7834,13	-0,03	0,00	3,52
	average	0,1875	0,6575	11,535
soccer	115,79	1,36	3,85	49,44
	434,53	1,37	1,89	25,68
	1915,40	1,65	1,00	10,55
	5541,40	1,12	0,44	5,34
	average	1,375	1,795	22,7525



A



B

Fig. 7.12. Comparison of various macroblock scans for 4CIF test sequences
 A CITY, B CREW, C HARBOUR, D SOCCER.

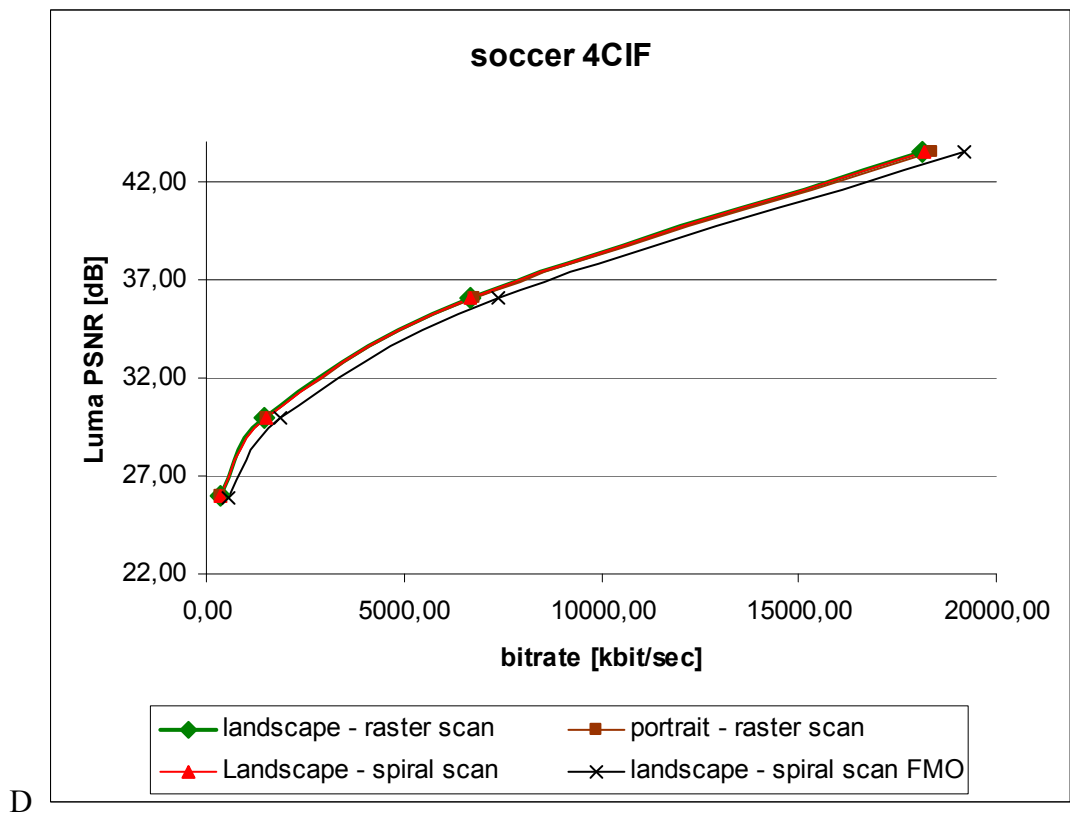
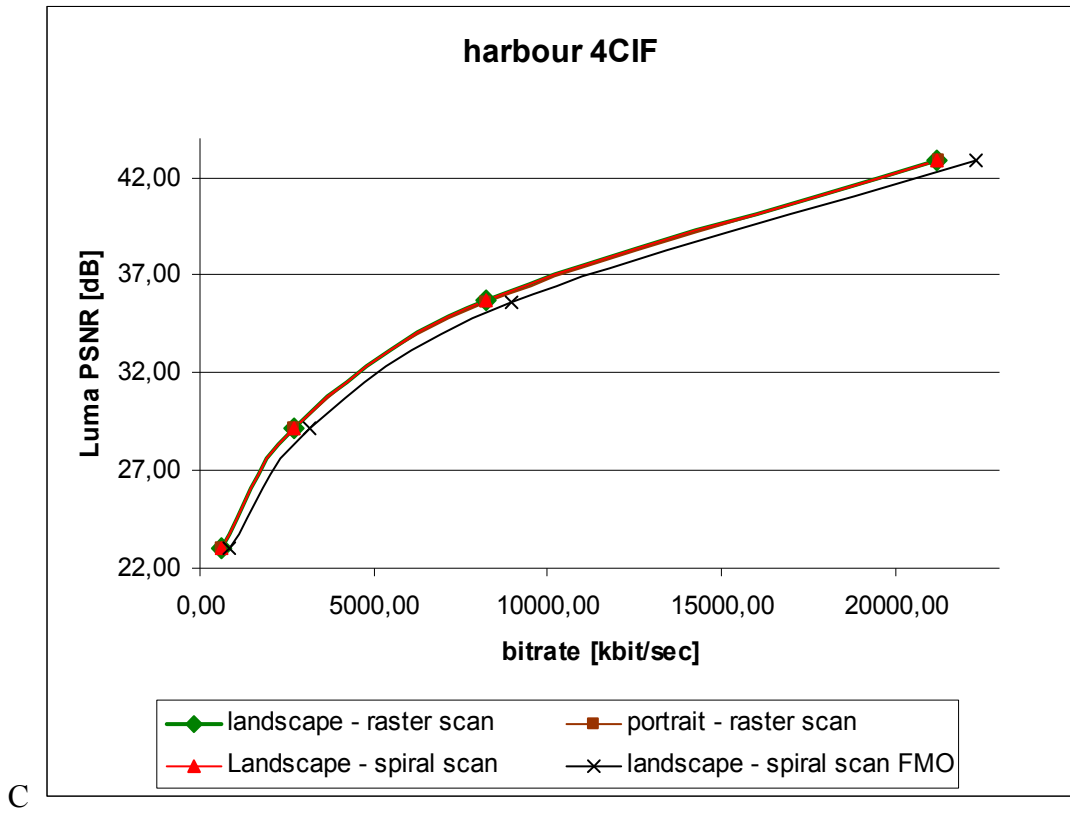


Fig. 7.12. (cont.) Comparison of various macroblock scans for 4CIF test sequences A CITY, B CREW, C HARBOUR, D SOCCER.

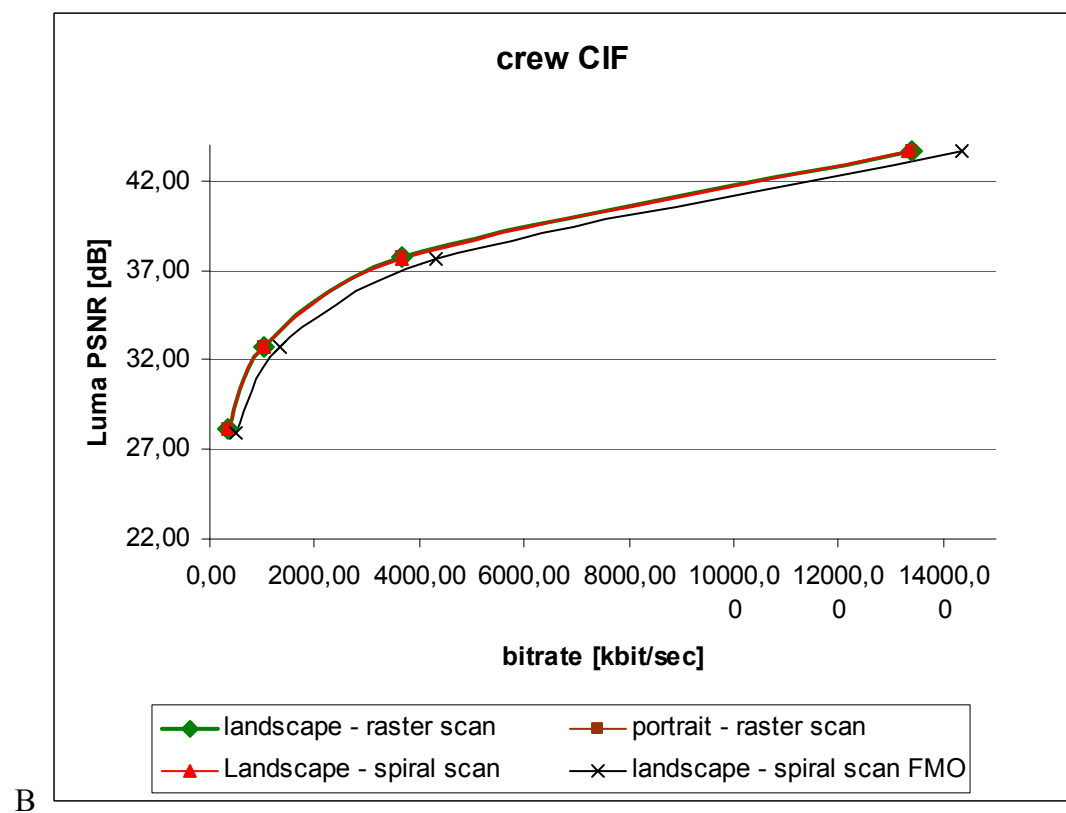
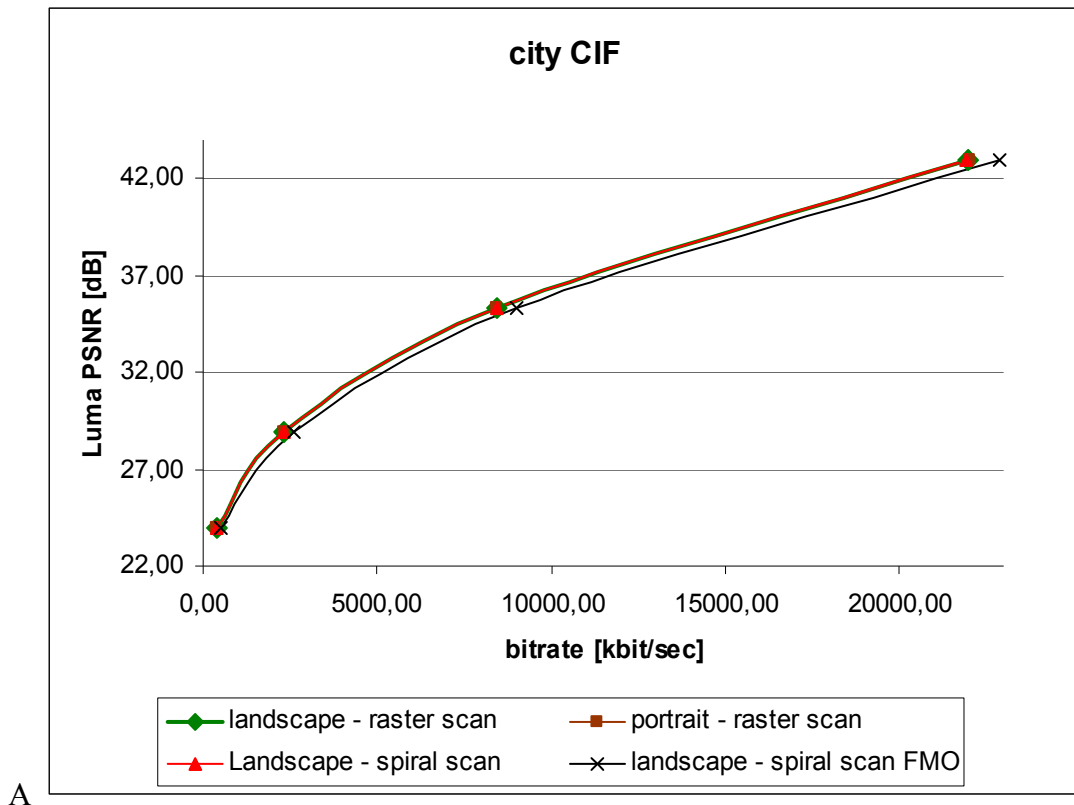
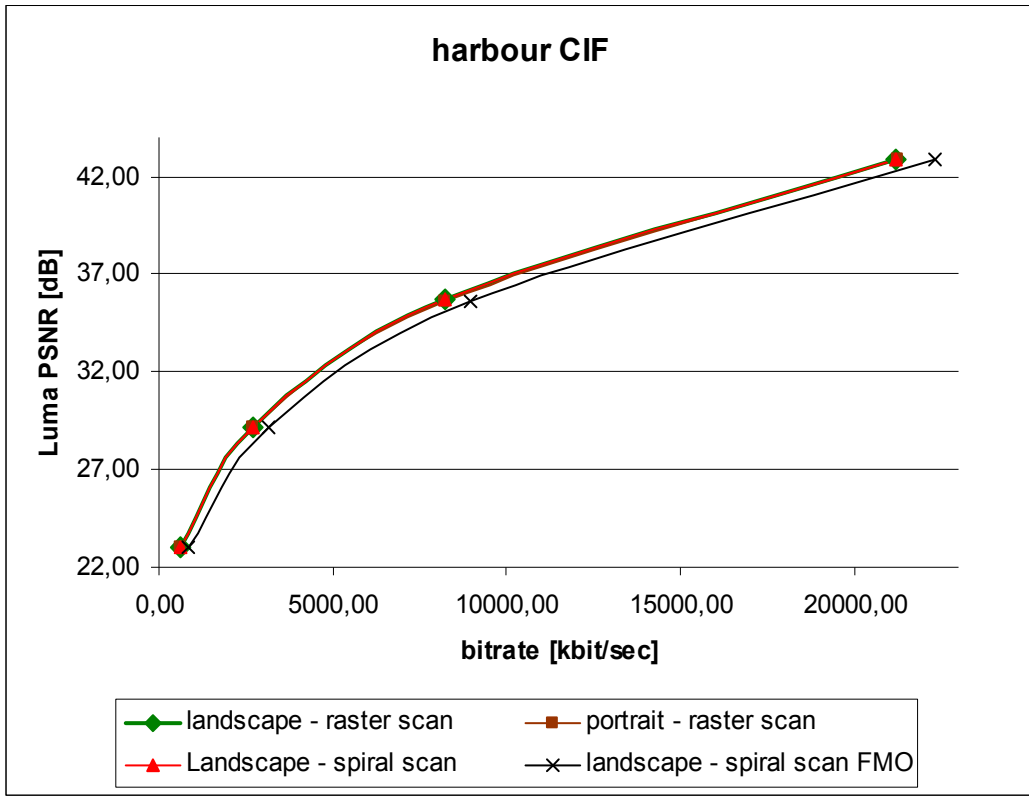
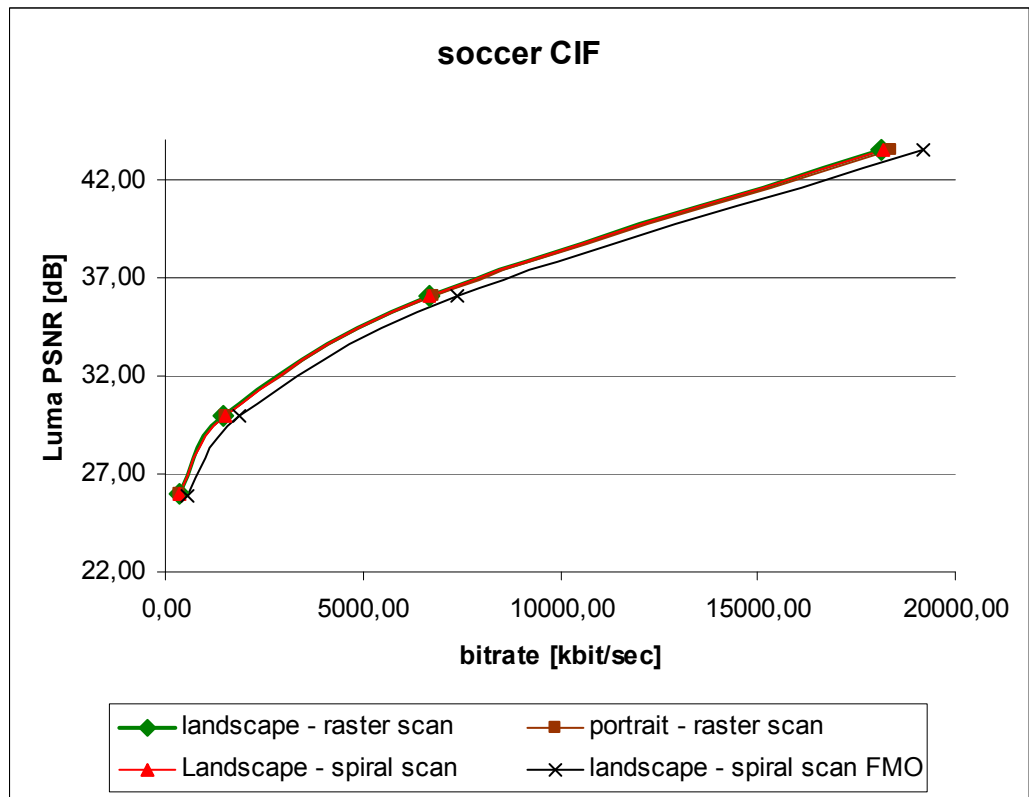


Fig. 7.13. Comparison of various macroblock scans for CIF test sequences A CITY, B CREW, C HARBOUR, D SOCCER.



C



D

Fig. 7.13. (cont.) Comparison of various macroblock scans for CIF test sequences A CITY, B CREW, C HARBOUR, D SOCCER.

The first two columns in Table 7.5 and 7.6 show the differences in bitrate overhead for various directions of macroblocks processing. The implication of those results is that the coding efficiency slightly depends on the direction of image processing for AVC/H.264 encoder. The average difference is very small, but it exists. The reason for that fact is that the intra prediction of 4×4 pixels blocks may use for the prediction process more pixels in horizontal direction than in vertical direction. It means that the horizontal direction prediction is preferred. So the coding efficiency for various scan orders of macroblocks depends on the image properties.

The results show that the modifications proposed by the author enable to encode the input video sequence by the use of the spiral scan order of macroblocks with no bitrate overhead in comparison with the raster scan of macroblocks. Moreover, it should be noticed that usage of FMO with the spiral scan drastically increases bitrate overhead. So, the conclusion for that fact is that spiral scan for an intra mode without coding algorithm modification is not acceptable for usage.

In the case of proposed modified encoder the results prove that it is possible to use the spiral scan in place of the raster scan without any bitrate overhead. So the conclusion is that the spiral order of macroblocks processing may be efficiently used in scalable coding obtained by data partitioning providing the bitstream with no bitrate overhead, but with new functionality.

7.2.3.2. Inter-frame coding test results

The aim of this experiment was to check what the influence on the encoding efficiency is when different macroblock order scans are used. Moreover, what is the influence on the coding efficiency when modifications to encoding algorithms are introduced for the spiral scan. In order to perform this experiment the single layer modified JSVM 1.0 verification model was used. Each test sequence was encoded at 6 different quality levels and the bitrate was compared. The conditions of the experiment were the following:

- input sequences: 4CIF and CIF,
- frame rate: 30Hz,
- GOP (open GOP): IPPPP,
- entropy coding: CABAC,

- quantization parameter: 20, 30, 35, 40, 45, 50,
- encoded frames: 100,
- motion search algorithm: diamond search,
- motion estimation accuracy: $\frac{1}{4}$ pixel,
- the spiral scan starting point: a centre of the picture,

Each test sequence was encoded by use of this verification model for three macroblock scan modes:

- raster scan,
- spiral scan obtained by FMO (no modified prediction of data elements),
- spiral scan used with modified prediction of data elements.

For all test sequences the first frame was encoded by the use of the same macroblock scan mode: raster scan. It was set so, because of the need to have the same reference for all experiments. The first frame should not influence the results when different modes were used.

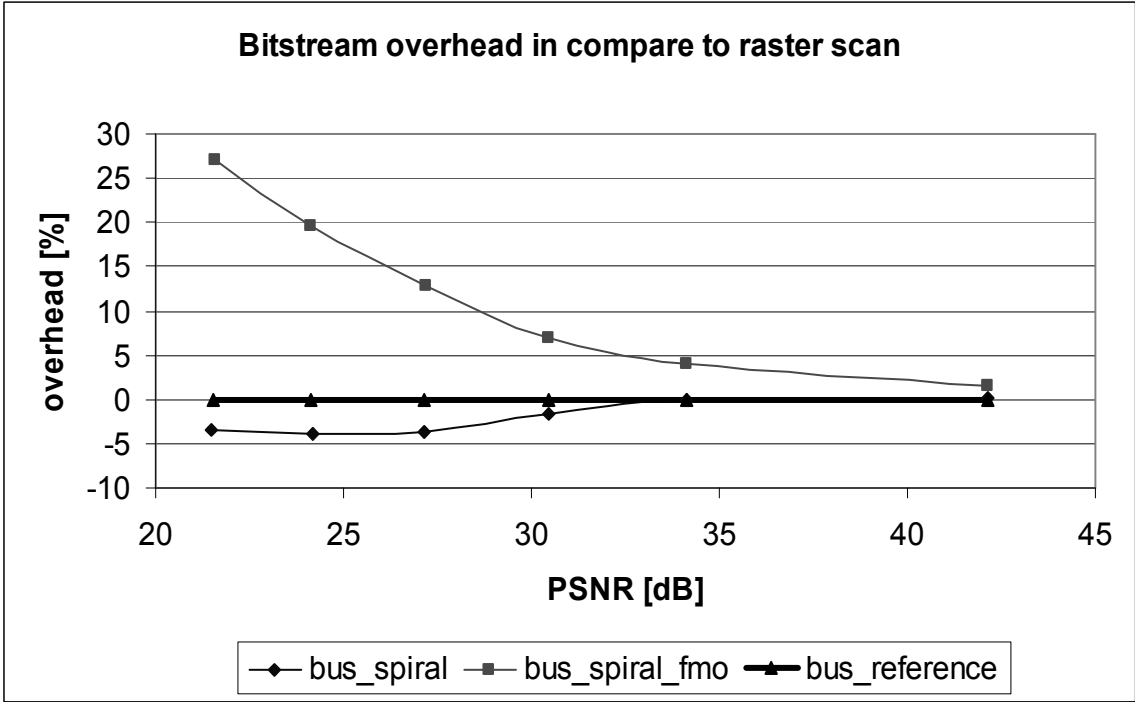


Fig. 7.14. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for BUS test sequence (CIF).

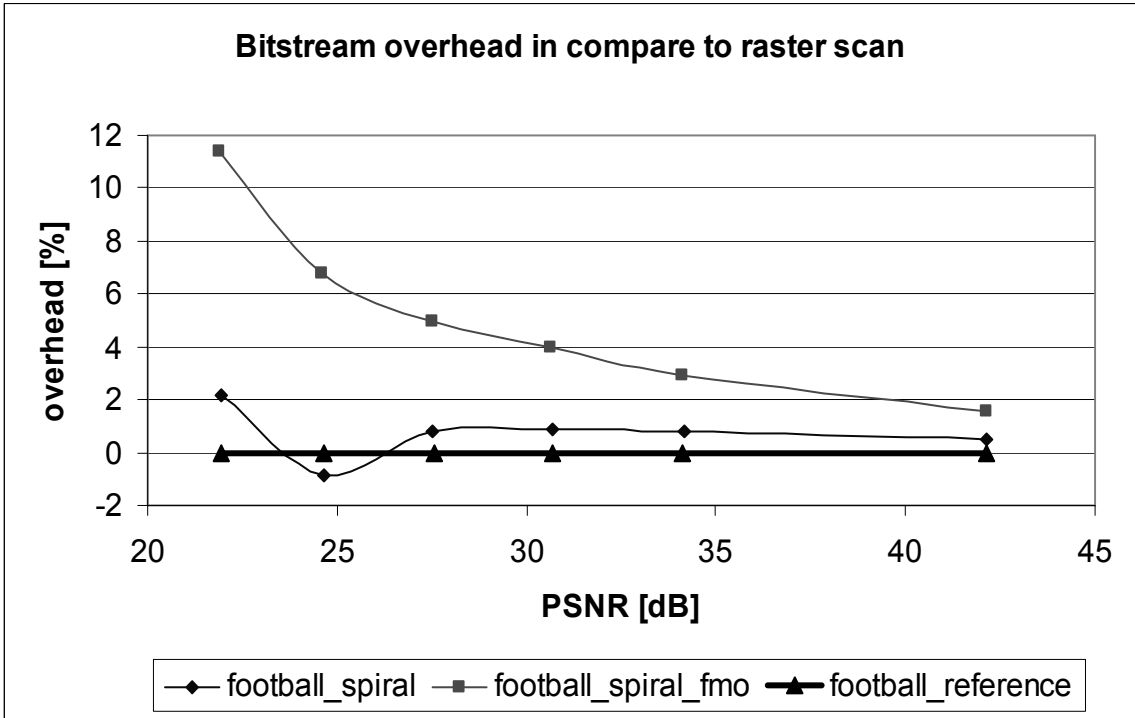


Fig. 7.15. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for FOOTBALL test sequence (CIF).

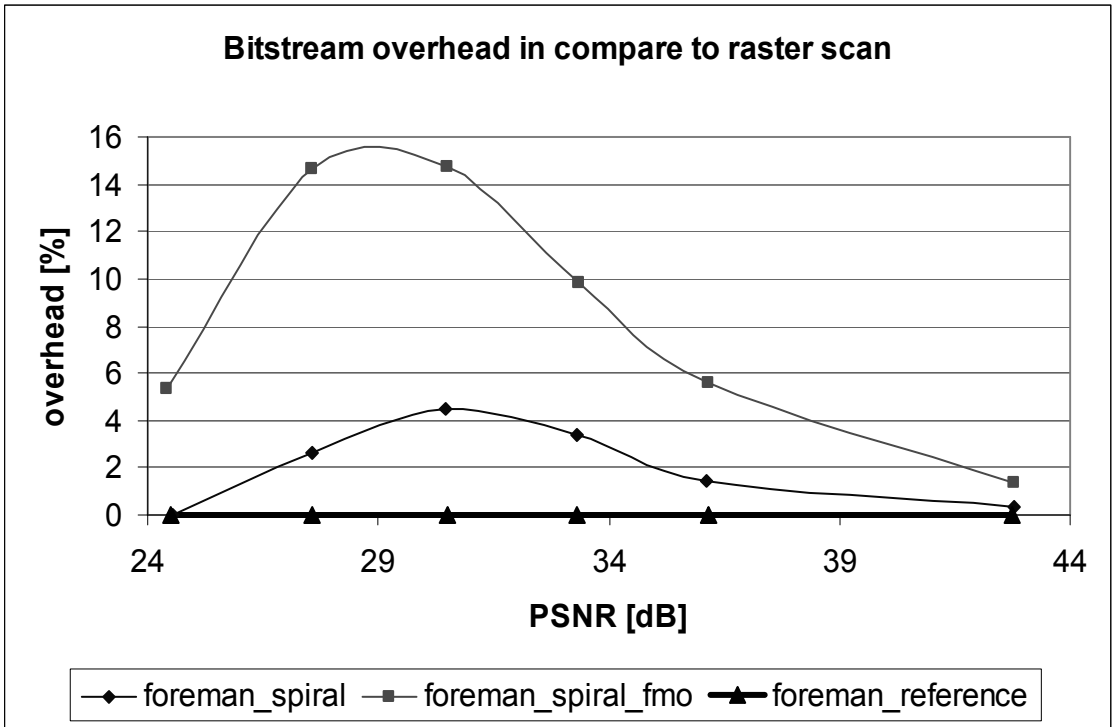


Fig. 7.16. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for FOREMAN test sequence (CIF).

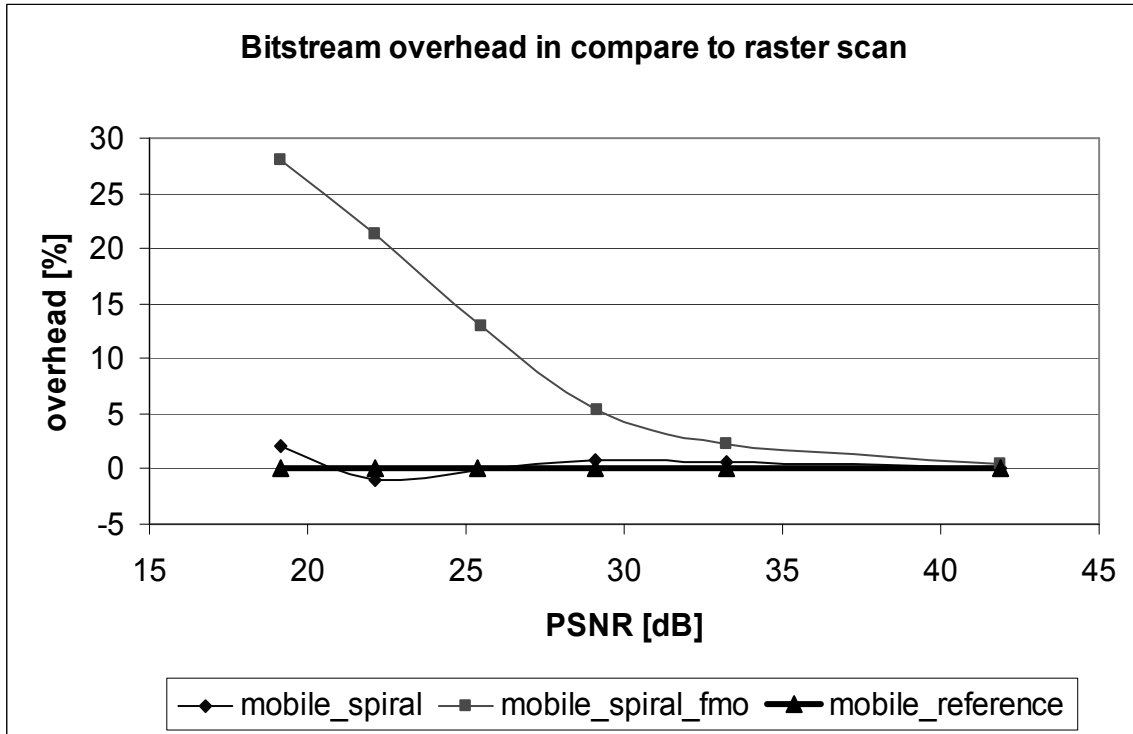


Fig. 7.17. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for MOBILE test sequence (CIF).

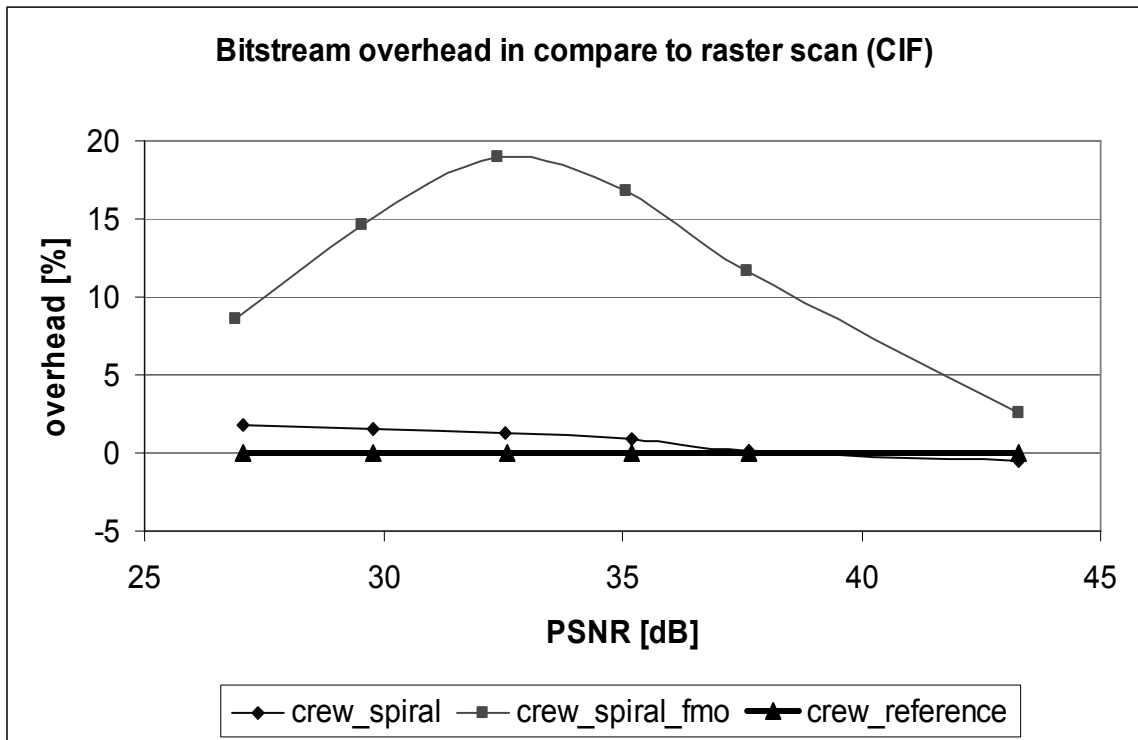


Fig. 7.18. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for CREW test sequence (CIF).

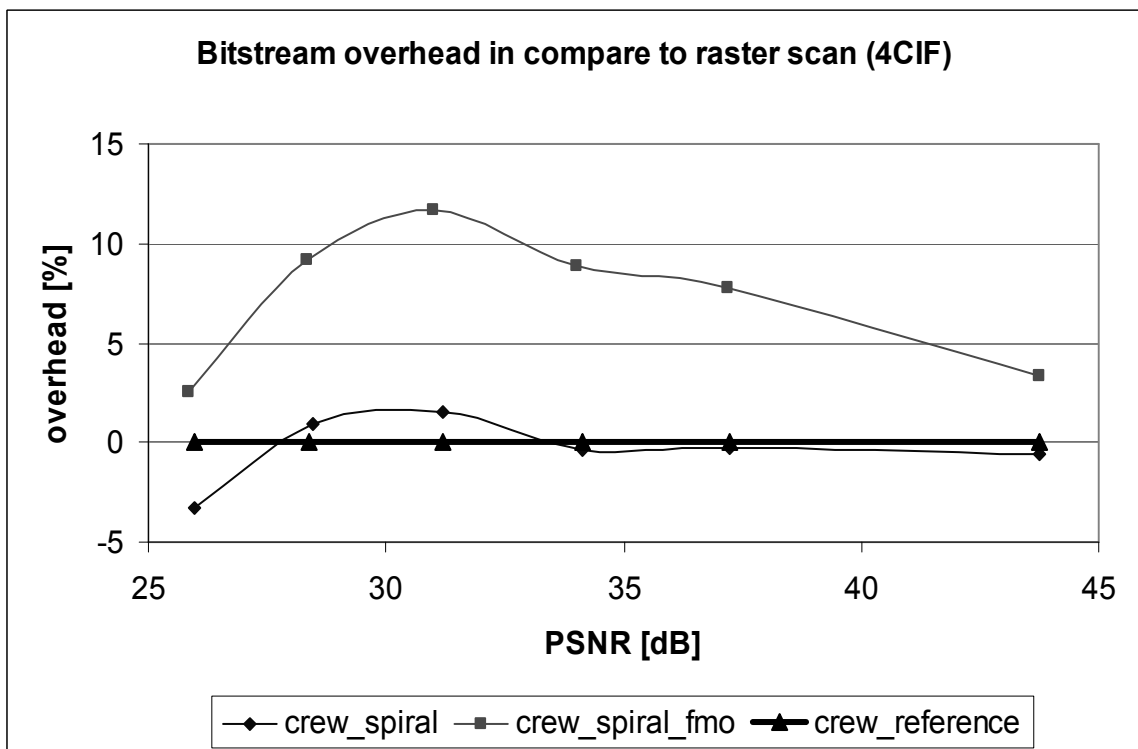


Fig. 7.19. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for CREW test sequence (4CIF).

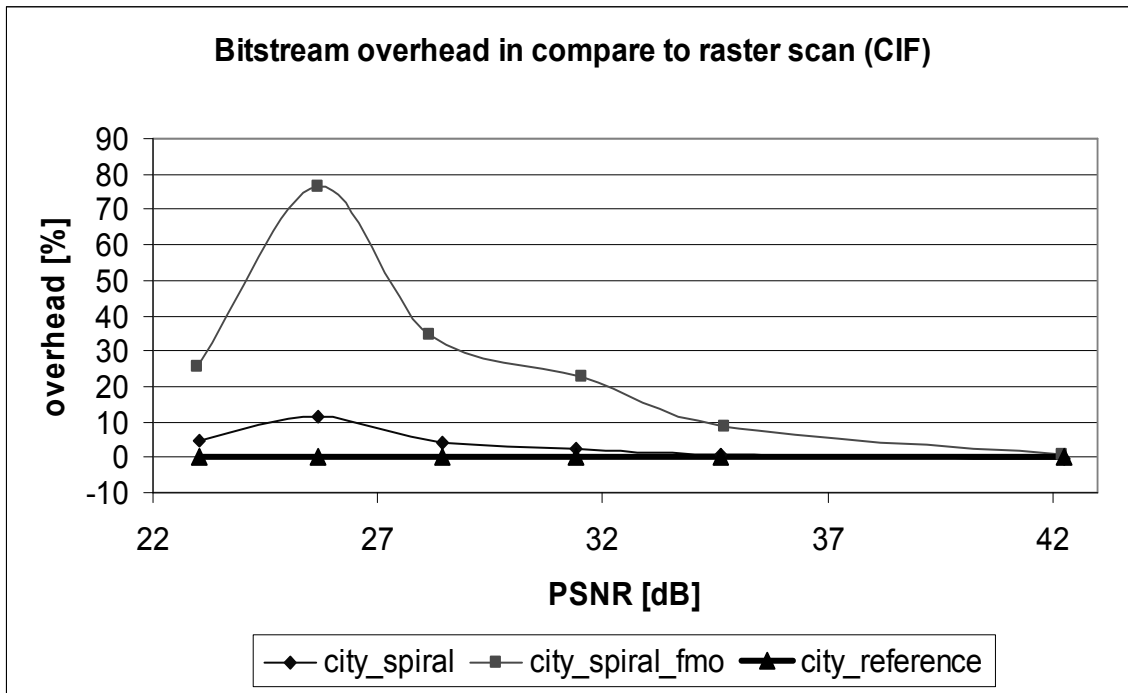


Fig. 7.20. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for CITY test sequence (CIF).

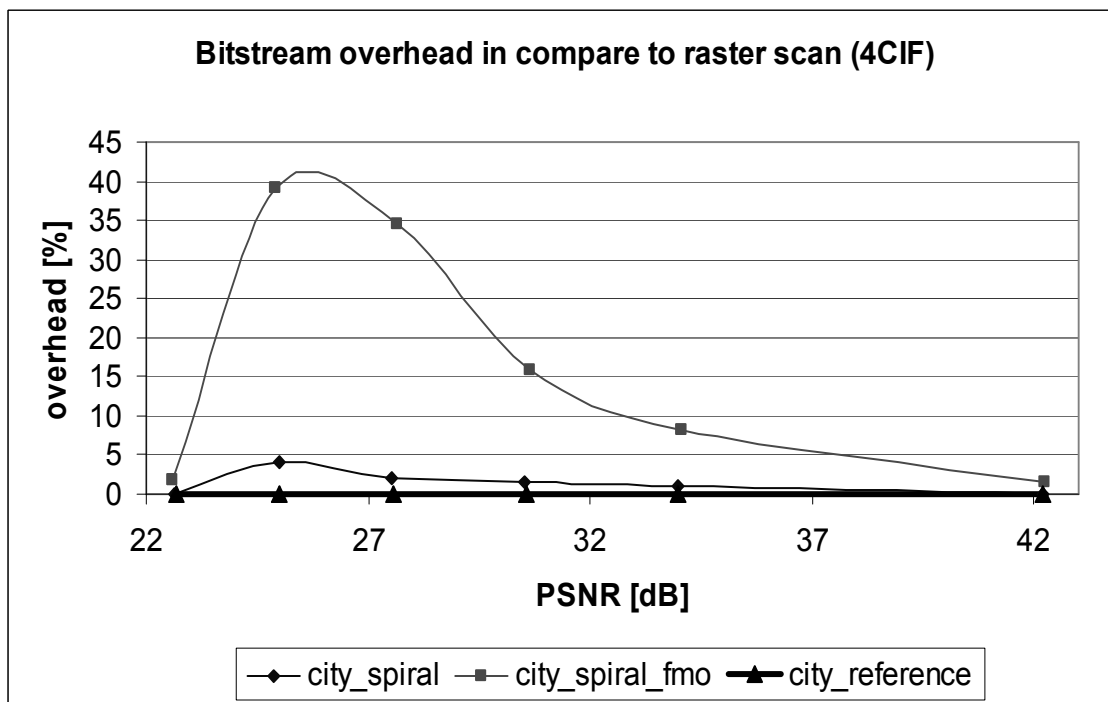


Fig. 7.21. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for CITY test sequence (4CIF).

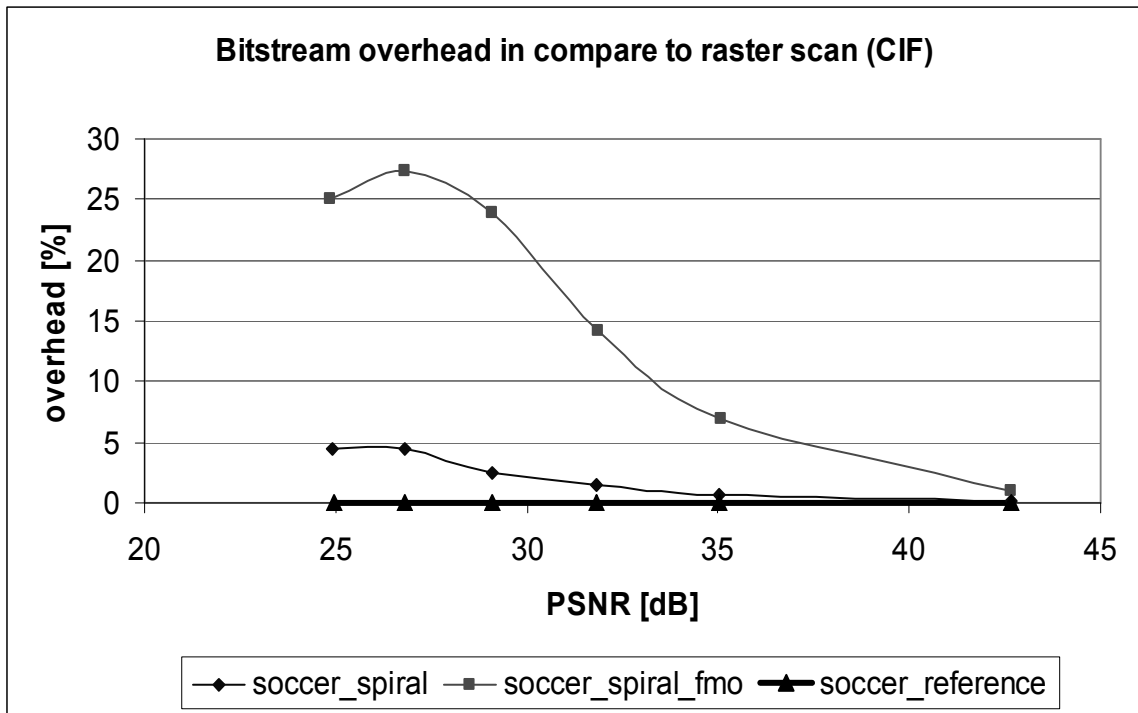


Fig. 7.22. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for SOCCER test sequence (CIF).

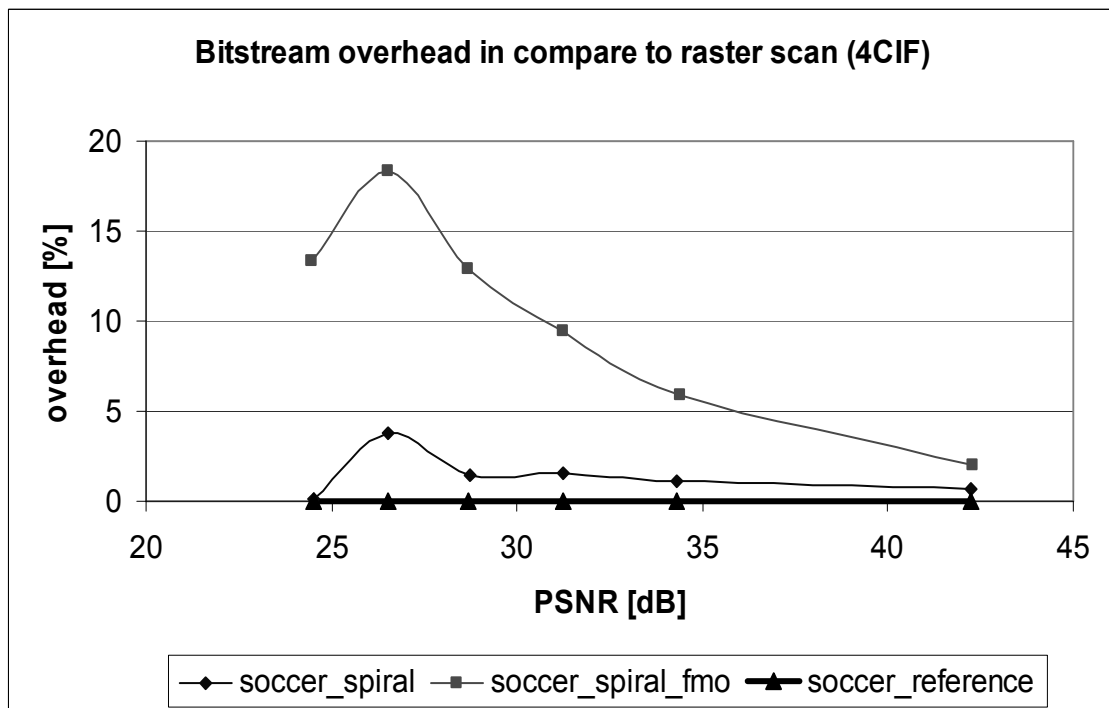


Fig. 7.23. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for SOCCER test sequence (4CIF).

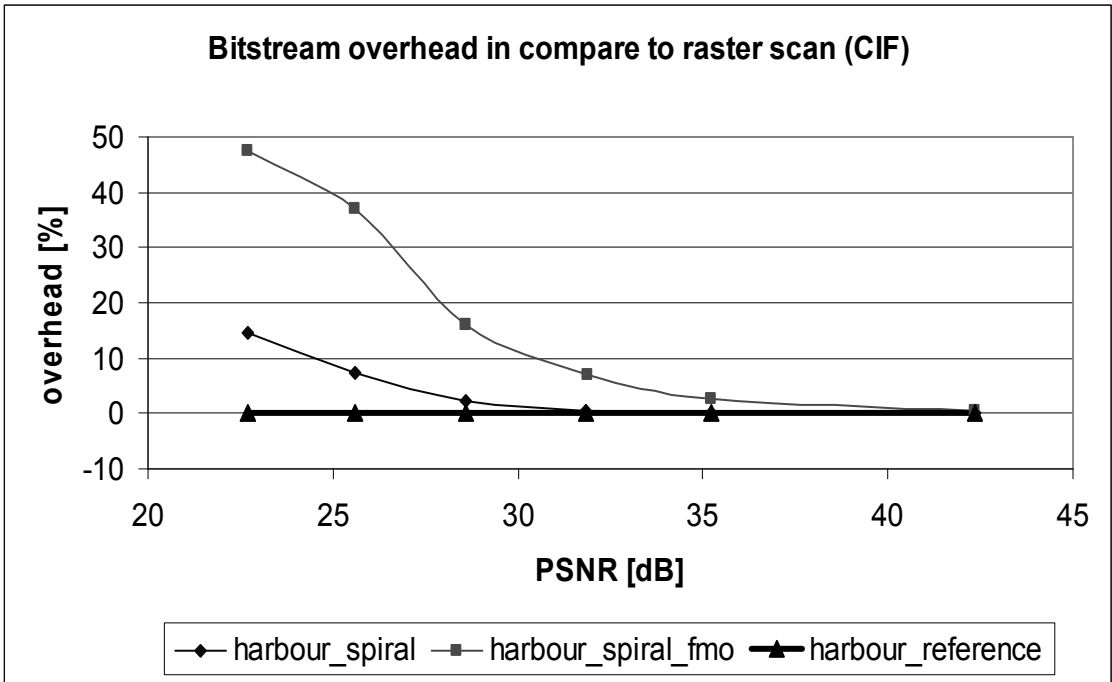


Fig. 7.24. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for HARBOUR test sequence (CIF).

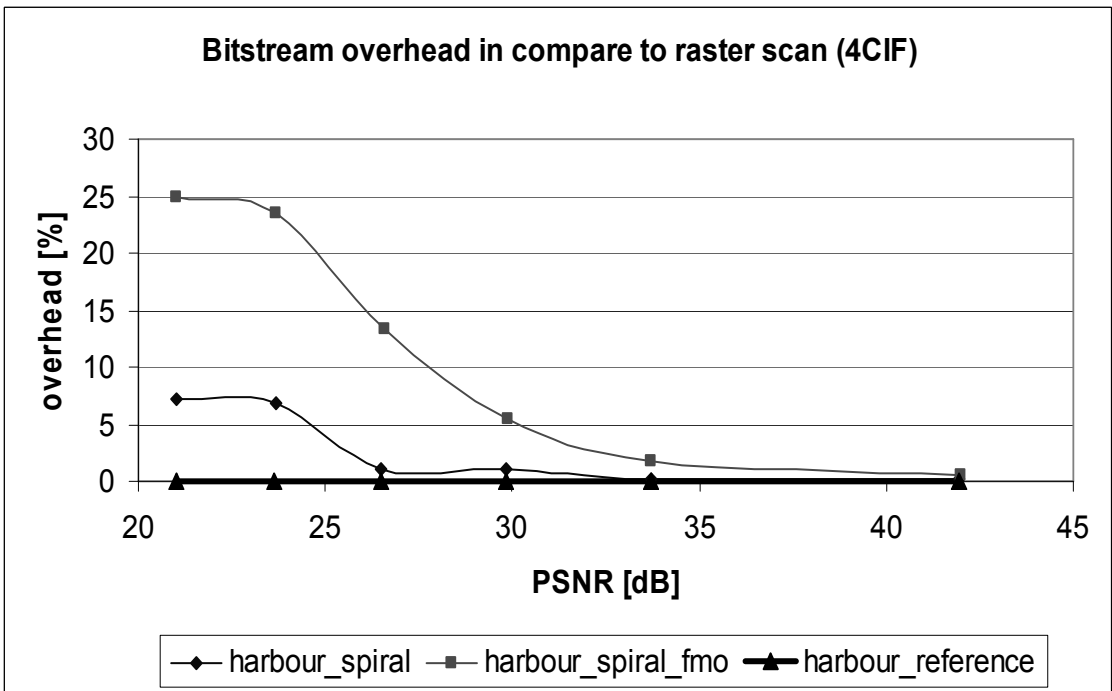


Fig. 7.25. Comparison of coding efficiency for raster scan, spiral scan (proposed) and spiral scan (FMO) for HARBOUR test sequence (4CIF).

While the macroblock is coded by the use of encoder with the raster scan of macroblocks the encoded data are predicted from left and upper neighborhood. In the case of the spiral scan often this neighborhood is unavailable, thus causing the encoder not to be able to predict this data, which at last leads to the loss of encoding efficiency. It can be noticed on all Figs. 7.14-25 that the spiral scan obtained by FMO causes huge bitrate overhead. When the modified H.264, proposed by the author, is used with spiral scan the gain in coding efficiency may be noticed (see Fig. 7.14, 7.15, 7.17-19) or small bitrate overhead (see Fig. 7.16, 7.20-25).

7.2.4. Testing the H.264 with raster scan and spatial, temporal and quality scalability

The aim of this experiment was to compare the coding efficiency for scalable video model (SVM) when the spiral scan and raster scan is used. When the spiral scan is used, the encoder gains the quality scalability feature, when the raster scan is used the encoder does not support quality scalability.

The test conditions of this experiment were the following:

- Search range (Full Pixel) 96
- Maximum number of iterations for bi-prediction search 4
- Maximum frame rate [Hz] 30
- IDR period of low-pass Picture 4
- Number of decomposition stages (Layer 0) 1
- Number of decomposition stages (Layer 1) 1
- Number of decomposition stages (Layer 2) 2
- Temporal filtering macroblock adaptive (Haar, 5/3)
- Maximum number of reference frame for filtering 1
- Update Step Yes
- Number of reference frames for DPCM 1
- Residual Prediction (Layer 0) never
- Base Layer Motion (Layer 1 and 2) independent estimation
- Residual Prediction (Layer 1 and 2) always
- Base Layer Motion (Layer 1 and 2) use base layer

The encoded bitstreams consist of three or four layers. The lowest layer (base layer) is a H.264 standard compatible with raster scan, QCIF spatial resolution and 15Hz temporal resolution. For all the remaining layers the spiral scan was applied for the author proposal and raster scan for reference. The layer 1 is QCIF spatial resolution and 15Hz temporal resolution, layer 2 is a 30Hz CIF and layer 3 is 30Hz 4CIF sequence. The results are shown in Table 7.7.

Table 7.7. Comparison of the SVM with spiral scan and raster scan.

		Standard		spiral	
		Bitrate [kbit/sec]	Luminance PSNR [dB]	Bitrate [kbit/sec]	Luminance PSNR [dB]
Bus	Layer 0 (176x144 @ 15.0Hz)	95.1824	29.242	95.1824	29.242
	Layer 1 (176x144 @ 15.0Hz)	190.4832	32.259	191.0384	32.246
	Layer 2 (352x288 @ 30.0Hz)	509.6736	29.108	510.7616	29.074
Football	Layer 0 (176x144 @ 15.0Hz)	192.2742	32.157	192.2742	32.157
	Layer 1 (176x144 @ 15.0Hz)	383.5163	36.144	383.6086	36.099
	Layer 2 (352x288 @ 30.0Hz)	1024.4335	33.463	1023.7246	33.444
Foreman	Layer 0 (176x144 @ 15.0Hz)	48.0224	31.760	48.0224	31.760
	Layer 1 (176x144 @ 15.0Hz)	95.8664	34.776	95.4312	34.633
	Layer 2 (352x288 @ 30.0Hz)	255.1768	32.971	255.5728	32.921
Mobile	Layer 0 (176x144 @ 15.0Hz)	64.5040	25.204	64.5040	25.204
	Layer 1 (176x144 @ 15.0Hz)	129.0464	28.161	129.0520	28.066
	Layer 2 (352x288 @ 30.0Hz)	387.0608	26.201	385.4768	26.179
City	Layer 0 (176x144 @ 15.0Hz)	63.5288	33.537	63.5288	33.537
	Layer 1 (176x144 @ 15.0Hz)	126.9720	36.329	128.3224	36.330
	Layer 2 (352x288 @ 30.0Hz)	511.3744	35.254	511.5984	35.218
	Layer 3 (704x576 @ 60.0Hz)	2045.0544	33.882	2029.9720	33.843

Crew	Layer 0 (176x144 @ 15.0Hz)	95.2024	33.160	95.2024	33.160
	Layer 1 (176x144 @ 15.0Hz)	192.0896	36.091	193.4096	36.079
	Layer 2 (352x288 @ 30.0Hz)	763.0848	36.553	766.2248	36.542
	Layer 3 (704x576 @ 60.0Hz)	3059.0768	36.966	3065.7776	36.964
Harbour	Layer 0 (176x144 @ 15.0Hz)	96.6696	30.466	96.6696	30.466
	Layer 1 (176x144 @ 15.0Hz)	192.9712	33.202	193.2200	33.162
	Layer 2 (352x288 @ 30.0Hz)	763.9552	31.754	762.2192	31.725
	Layer 3 (704x576 @ 60.0Hz)	3082.5256	33.332	3083.5352	33.325
Soccer	Layer 0 (176x144 @ 15.0Hz)	95.5280	33.887	95.5280	33.887
	Layer 1 (176x144 @ 15.0Hz)	192.2168	37.151	192.3424	37.119
	Layer 2 (352x288 @ 30.0Hz)	762.2752	36.189	762.1208	36.158
	Layer 3 (704x576 @ 60.0Hz)	3065.3912	36.115	3065.7784	36.097

As it can be noticed the compression efficiency for scalable codec with raster and spiral scans is similar. The differences in PSNR values are in the range of ± 0.05 dB and the bitrate mismatch is in range of $\pm 0.5\%$. Thus, the results prove that usage of spiral scan in scalable coding does not change the coding efficiency.

7.3. Assessment using subjective measure

7.3.1. Introduction

To evaluate a subjective quality of coded video the Single Stimulus MultiMedia (SSMM) test method has been performed. The SSMM test method has been derived from the Single Stimulus method, as described in ITU-R rec. BT 500-11, and the Single Stimulus with two repetitions. The SSMM method requires the use of progressively scanned display, such as computer monitors, LCD displays or DLP projectors.

For the tests, the distance between the screen and the viewer has been defined. There are three distances defined according to the dimensions of the image which have to be assessed:

- 1. HI=704x576
- 2. MID=352x288
- 3. LOW=176x144

To ensure that the proper relationship between display height and viewing distance is preserved the DLP projector was used. A viewing distance of 3H (3 times of the display height) was used for HI case, 4H for the MID case, and 6H for LOW case.

In order to avoid the well known effect of subjective test, which is called “Contextual effect”, the test sequences are presented twice in a different order. The “Contextual effect” means that when two subsequent conditions have highly different quality the judgment is not as fair as when two subsequent conditions have quite the same quality. This effect is particularly strong in the Single Simulcast category test method where no reference is presented.

The SSMM test method protocol is shown in Fig. 7.26. Each sequence is presented for 10 second, then for 5 second the invitation for voting is displayed.

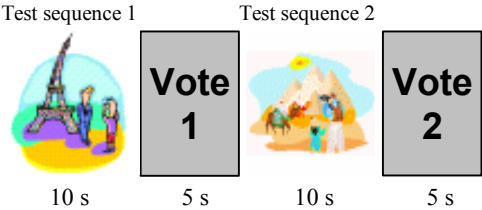


Fig. 7.26. SSMM test method presentation protocol [ISO04g].

11 grade voting scale (from 0 to 10) is given in Fig. 7.27.

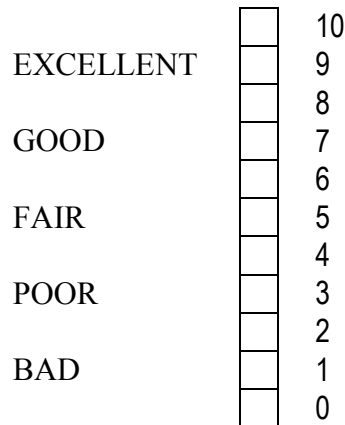


Fig. 7.27. SSMM test method voting scale [ISO04g].

Each displayed test sequence received two votes. The final score was obtained by making the mean of these two values. The tests were performed by means of at least 20 subjects.

7.3.2. Testing the scalable model with quality scalability based on JM 7.3 reference software

The tests have been prepared and executed by MPEG testing group in Germany and in Italy. There were two scenarios of experiments: one for bitstreams containing three spatial scalable levels, three temporal scalable levels and SNR scalability; the other scenario was for bitstreams containing two spatial scalable levels, three temporal scalable levels and SNR scalability. The results of those scenarios are shown in Fig. 7.28 and Fig. 7.29. Figures show results of assessment of several codecs which took part in the competition for the best scalable video coding solution. The codec proposed by the group, the author has been part of, has been given one of three highest notes as compared to other codecs. The codec that has been chosen as the best one was one proposed by Fraunhofer Institute for Telecommunications – Henrich Hertz Institute.

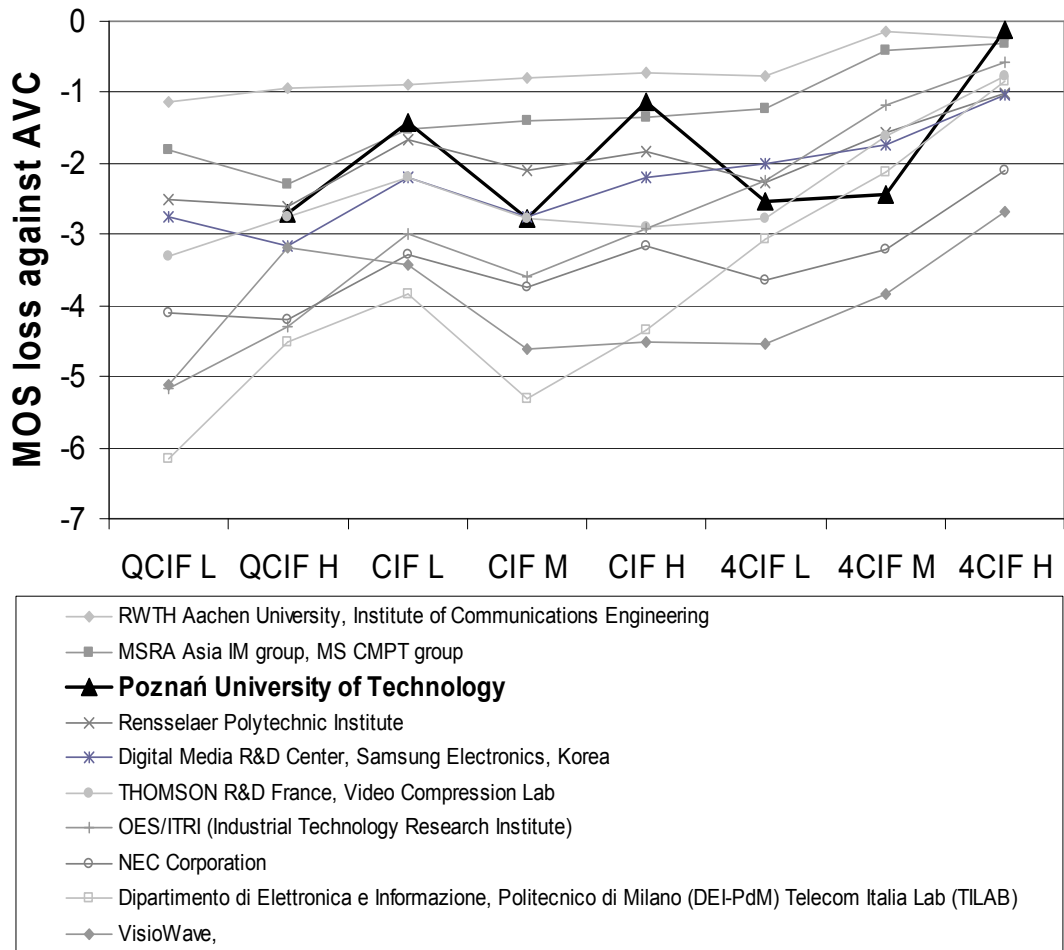


Fig. 7.28. Test scenario 1 (3 spatial levels, 3 temporal levels and SNR scalability).

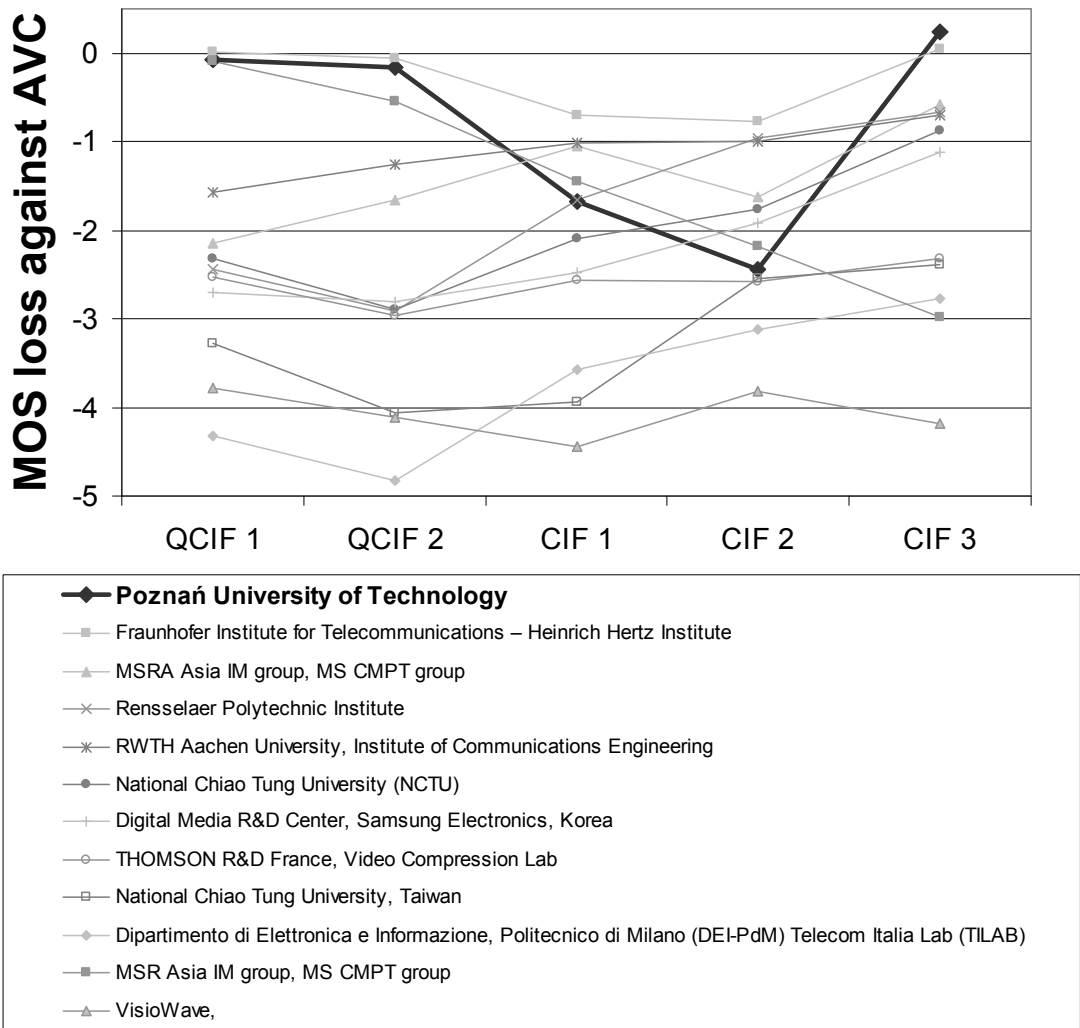


Fig. 7.29. Test scenario 2 (2 spatial levels, 3 temporal levels and SNR scalability).

The figures show the mean opinion score (MOS) of codecs against the non-scalable reference H.264 codec. Value 0 indicated no subjective quality loss compared to non-scalable H.264 codec.

As it can be noticed, the Poznań University of Technology codec, which is mostly the author's proposal, is one of the best scalable solutions. The significant decrease of MOS for scenario 2 is caused by one broken test sequence, which received very poor notes. The fluctuations of the subjective quality in scenario 1 may be explained by improper RoI setting for some test sequences. The encoder expected, for all test sequences, the RoI to be in the centre of the picture, what was false for some sequences. In the case of proper RoI setting the expected MOS would be higher.

The author's codec has been only a little bit worse than the winner codec. But, the most efficient scalable codec has additional (quite significant) cost of complexity

which is MCTF (*Motion Compensated Temporal Filtering*). The author’s solution is much less computationally complex, which is quite important.

7.3.3. Testing the scalable model with quality scalability based on JSVM 1.0 software

The subjective tests have been performed by the use of 20-person test group. To ensure that the proper relationship between display height and viewing distance is preserved the DLP projector was used. A viewing distance of 3H (3 times of the display height) was used. The results on Figs. 7.30 – 7.32 show the quality decrease when spiral scan is used and bitstream cut is applied.

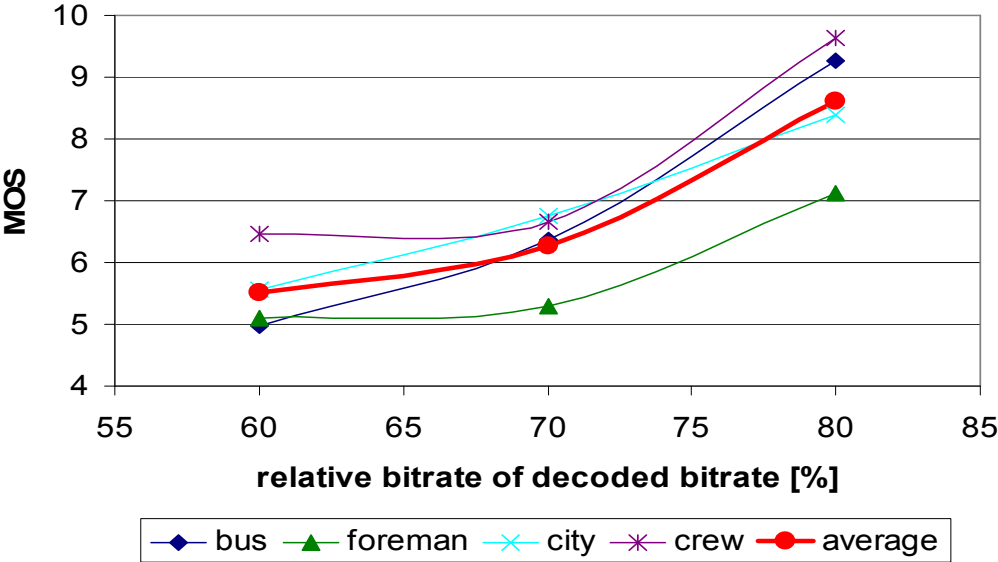


Fig. 7.30. Subjective results in mean opinion score measure.

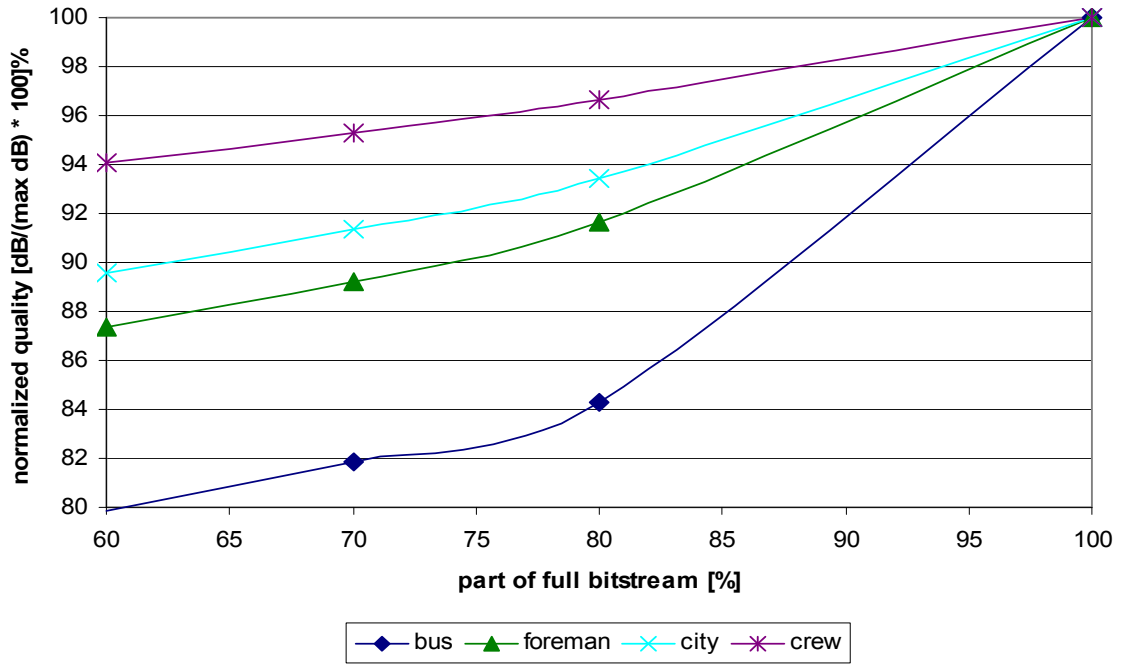


Fig. 7.31. Normalized results of objective quality measure for given subjective tests.

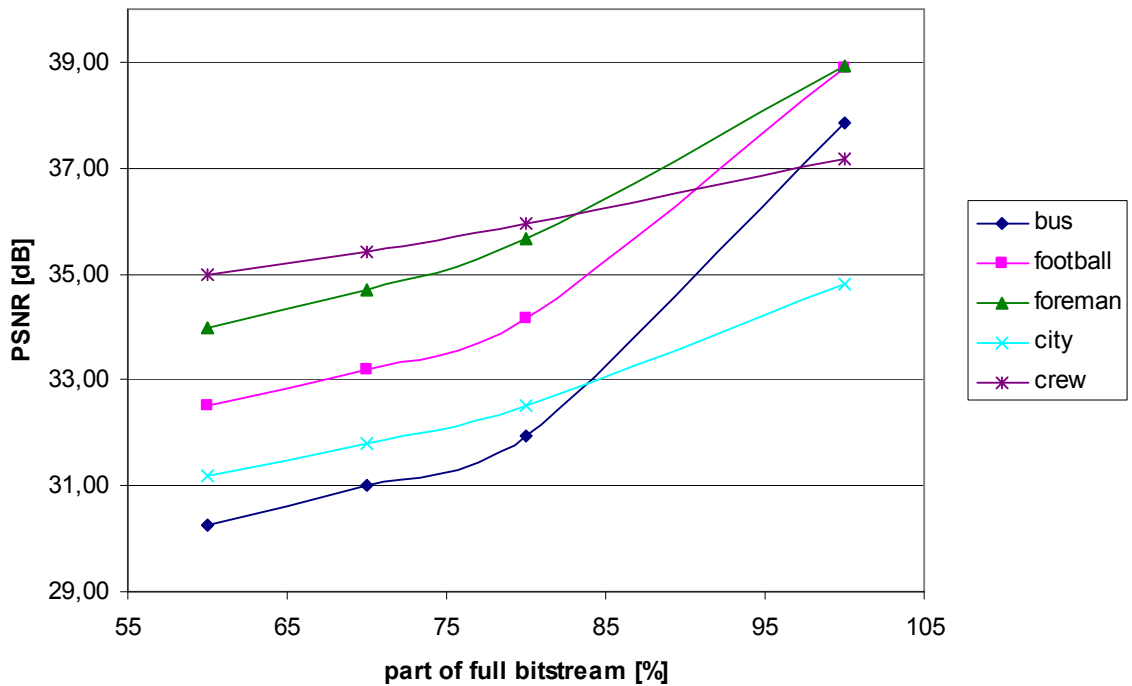


Fig. 7.32. Objective results for given subjective tests.

As it may be noticed on the Fig. 7.30 the subjective quality loss is proportional to the bitrate decrease. Moreover, the average subjective quality for 80% of the bitstream is higher than 80% of the normalized quality as compared to original. It means that observers did not notice the objective quality decrease. The real quality loss was

hidden for them. As the reference the objective quality loss is shown in the Fig. 7.32 for all test sequences.

The results show that the proposed technique of quality scalability obtained by the spiral scan of macroblocks is an efficient method and may be applied into any advanced video coding technique without almost any cost.

Chapter 8

Conclusions

8.1. *Summary*

In this work, the author has presented descriptions and results of experiments which prove the thesis of this doctoral dissertation, formed in the introduction. Against the historical background of video codec development, in Chapter 2, and the description of current scalable video coding technology, in Chapter 3, the author has presented the structure of the scalable video codec which became the basis for designing a verification model of new coding techniques presented in this work. The structure of scalable video codec has been presented in Chapter 4.

Next, in Chapter 5, the modified algorithm for spiral scan of macroblocks has been presented. In this chapter the spiral scan of macroblock has been proposed as a tool for fine granularity scalability.

In Chapter 6 the author has presented the description and the results of experiments which were done in order to properly set several parameters of the tools used for proposed scalable coding. Parameter setting was the integral part of proposed scalable coding designing.

The verification model of scalable codec with properly set parameters has been used to perform experiments for verification of several features of the designed technique. In Chapter 7, the results of these experiments are presented. The results prove that the scalable technique, proposed by the author for advanced video coding is very efficient.

For the proposed scalable encoder the lowering of encoding performance in comparison with non scalable techniques, which is a common drawback for all scalable encoding techniques, is between 0% up to 15%. The encoding efficiency of less advanced techniques but based on the same structure as the proposed encoder is similar. A good example of such an encoder is the one proposed in the doctoral dissertation [Mac02].

In Chapter 7 there are the results of the comparison of several scalable coding techniques, including the author's one. As it can be noticed, the author's technique is, in general, one of the most efficient techniques (see Fig 7.28 and 7.29 on pages 7-32 to 7-33).

The scalable encoder proposed in this doctoral dissertation consists of some independent tools. Each of the tools may be used separately as well as jointly. All of them have been analyzed and described in this work. Several experiments have been performed and results are placed in Chapter 7. The results on pages 7-13 to 7-29 show that using the modified spiral scan of macroblocks for both scalable and non-scalable coding does not make the coding efficiency worse in comparison with the standard raster scan of macroblocks. As a result, a new functionality is achieved without any cost (such as encoding performance). This feature has been used to provide fine grain scalability into advanced video codec. The quality scalability, achieved in that way, has also been investigated. The objective tests (see Fig. 7-32) and subjective tests (see Figs 7-28 to 7-31) have been done in order to estimate their usability in real environment. The fact that the coding-decoding delay is the same as for non-scalable AVC is quite an important feature of the proposed codec.

8.2. *Original achievements*

The author's work on the codec in this doctoral dissertation was done parallel with the work of other scientists on similar codecs, among which the best scalable advanced video codec was chosen as the base for the development of a new international scalable video coding standard. This fact gave the author an opportunity to participate in the development of the new international standard. The author worked together with world leading groups of scientists. The group that author cooperated with had a significant influence on the form of the new scalable advanced video coding

standard. The technique proposed by the author was assessed as one of the best solutions for scalable coding.

In this work the author has described his original achievements. These are designing and implementing scalable codecs based on the advanced video coding technology. The author has built the following verification models of scalable codecs:

- scalable codec based on H.263 non-scalable codec,
- scalable codecs based on H.264 non-scalable codec,
- scalable codec based on JSVM (Join Scalable Video Model) with extensions proposed by the author.

For the verification models based on non-scalable H.263 and H.264, the author has used existing techniques for scalable coding, as proposed in [Mać02], and adapted them to the new advanced video coding technology. The proposed scalable advanced video codec with multilayer structure consists of several extensions made by the author. Pixel data prediction modes, such as prediction from the lower layer and prediction from the average of lower layer and one of the reference frames, have been modified to fit the new advanced video coding with scalable extension. Those modes have been extended (see Chapter 4) to take advantage of the new capabilities of advanced video coding technology.

A very important achievement is the spiral scan as a tool for scalability of fine granularity. A very promising result of scalable coding when spiral scan is used has been presented in Chapter 7. The quality and efficiency of the proposed technique have been estimated by use of subjective tests. These tests have been performed by a special group from MPEG as well as by students from Poznań University of Technology. The description and results of the tests have been presented in Chapter 7.

In order to make the spiral scan a very efficient tool, the author has made several modifications in the verification model with scalable extension of H.264 codec. The motion vector prediction algorithm has been modified to use efficiently all motion vectors existing in the neighborhood of the current macroblock for prediction of current motion vectors. The standard algorithm uses for prediction the motion vectors from only some of neighboring macroblocks. Also the spatial prediction when intra coding is used was modified to predict samples from all spatial directions. This provides the ability to code intra macroblocks efficiently at any position in a picture when the spiral scan is used. The author has also modified data prediction in the CABAC entropy codec,

allowing prediction from any possible direction, and not only left and up as it was before. The detailed description of these modifications has been presented in Chapter 5.

The author's achievement is also the parameters setting for all verification models of scalable codecs. The proper values of various parameters of scalable encoders have a very strong influence on coding efficiency. The description of parameters, experiments and their results have been presented in Chapter 6.

The most important achievements of this doctoral dissertation are:

- Designing of a scalable advanced video codec (adaptation of classic structure to the advanced coding technique);
- Inter-layer prediction (proposal and experimental tests);
- Assessments of the spiral scan as a tool for fine granularity scalability (assessment of the coding efficiency of spiral scan in comparison with raster scan, experimental tests).

The research done for this doctoral dissertation may be continued. The technique for achieving quality scalability by the use of spiral scan may be further improved. It could be modified in such a way that it could be used in temporal high-pass layers of a scalable codec proposed by MPEG, which is a H.264-based scalable encoder with motion compensated temporal filtering.

References

Author's contributions

- [Dom02a] M. Domański, S. Maćkowiak, Ł. Błaszak, "Scalable Video Compression for Wireless Systems", URSI X Nat. Symposium of Radio Sciences, pp. 336-340, Poznań, March 2002.
- [Dom02b] M. Domański, Ł. Błaszak, S. Maćkowiak, A. Łuczak, „Spatio-Temporal Scalability in DCT-based Hybrid Video Coders”, ISO/IEC JTC1/SC29/WG11 MPEG2002/M8672, Klagenfurt, July 2002.
- [Dom02c] M. Domański, S. Maćkowiak, Ł. Błaszak, "Efficient Structure of Video Coders with Motion-Compensated Fine-Granularity Scalability", Proceedings of XI European Signal Processing Conference EUSIPCO'2002, September 3-6, 2002 Toulouse, France.
- [Dom02d] M. Domański, S. Maćkowiak, Ł. Błaszak, "Fine Granularity In Multi-Loop Hybrid Coders With Multi-Layer Scalability", Proceedings of IEEE International Conference on Image Processing 2002, Rochester, NY, USA.
- [Dom02e] M. Domański, S. Maćkowiak, Ł. Błaszak, "Efficient Hybrid Video Coders with Spatial and Temporal Scalability", Proceedings IEEE International Conference on Multimedia and Expo ICME'2002, August 26-29, 2002 Lausanne, Switzerland.
- [Dom02f] M. Domański, S. Maćkowiak, Ł. Błaszak, "Scalable Hybrid Video Coders with Double Motion Compensation", Proceedings of International Symposium on Consumer Electronics, September 23-26, 2002 Erfurt, Germany.
- [Bla02] Ł. Błaszak, Z. Korus, M. Domański, "Performance of H.26L / JVT Coding Tools", International Conference on Computer Vision and Graphics (ICCVG 2002), September 25-29, 2002, Zakopane, Poland.
- [Dom03] M. Domański, Ł. Błaszak, S. Maćkowiak, "AVC Video Coders with Spatial and Temporal Scalability," Picture Coding Symposium, Saint Malo, pp. 41-47, April 2003.
- [Bla03a] Ł. Błaszak, M. Domański, S. Maćkowiak, "Spatio-Temporal Scalability in AVC codecs", Doc. ISO/IEC JTC1/SC29/WG11 M9469, March 2003.
- [Bla03b] Ł. Błaszak, S. Maćkowiak, "Koder Wizyjny AVC ze Skalowalnością Przestrzenno-Czasową", Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2003, Wrocław, Poland, 25-27 czerwca 2003, pp. 345-348.

- [Bla03c] Ł. Błaszak, M. Domański, "Modified AVC Codecs with Spatial and Temporal Scalability", ISO/IEC JTC1/SC29/WG11 MPEG2003/M9895 Trondheim, July 2003.
- [Bla03d] Ł. Błaszak, M. Domański, S. Maćkowiak, "Multi-Loop Spatio-Temporal Scalable Video Coders", ECCTD '03, European Conference on Circuit Theory and Design, Kraków, Poland, September 1-4, 2003.
- [Bla03e] Ł. Błaszak, M. Domański, S. Maćkowiak, "Extended AVC/H.264 Video Codecs with Mixed Scalability", Proceedings of 10th International Workshop on Systems, Signals and Image Processing, IWSSIP'2003, Prague, Sept. 10-11, 2003, Recent Trends in Multimedia Information Processing (ed. B. Šimák, P. Zahradník), Czech Technical University in Prague, 2003, pp. 73-76.
- [Bla04a] Ł. Błaszak, M. Domański, R. Lange, S. Maćkowiak, "Scalability of Modified AVC/H.264 Video Codecs", 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, Lisboa, Portugal.
- [Bla04b] Ł. Błaszak, M. Domański, R. Lange, A. Łuczak, "Scalable AVC Codec", ISO/IEC JTC1/SC29/WG11 MPEG04/M10626, 2004, Munich, Germany.
- [Bla04c] Ł. Błaszak, R. Lange, "Skalowalny Koder AVC z Ulepszoną Predykcją Wektorów Ruchu", Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2004, 2004, Warsaw.
- [Bla04d] Ł. Błaszak, M. Domański, R. Lange, "Modified AVC Codec with SNR Scalability Based on Macroblock Hierarchy", International Workshop on Systems, Signals and Image Processing, IWSSIP'04, pp. 15-18, Poznań.
- [Dom04a] M. Domański, Ł. Błaszak, R. Lange, A. Łuczak, "Propozycje Skalowalnego Kodera Obrazu MPEG", Przegląd Telekomunikacyjny, 2004.
- [Dom04b] M. Domański, Ł. Błaszak, R. Lange, A. Łuczak, "Propozycje Skalowalnego Kodera Obrazu MPEG", Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2004, 2004, Warsaw.
- [Bla04e] Ł. Błaszak, M. Domański, R. Lange, A. Łuczak, "Response to SVC CE2 Tasks: Testing of SNR Scalability Technologies", ISO/IEC JTC1/SC29/WG11 MPEG04/M11101, 2004, Redmond, USA.
- [Bla04f] Ł. Błaszak, M. Domański, R. Lange, "SVC CE6 Results at Poznań University of Technology", ISO/IEC JTC1/SC29/WG11 MPEG04/M11418, 2004, Palma, Spain.

- [Lan04] R. Lange, Ł. Błaszak, M. Domański, "Simple AVC-Based Codecs with Spatial Scalability", International Conference on Image Processing 2004, Singapore.
- [Bla05a] Ł. Błaszak, "Implementation of SNR Scalability Using Spiral Scan", ISO/IEC JTC1/SC29/WG11 MPEG04/M11735, 2005, Hong Kong, China.
- [Bla05b] Ł. Błaszak, M. Domański, "Results of CE5 in Scalable Video Coding", ISO/IEC JTC1/SC29/WG11 MPEG04/M11734, 2005, Hong Kong, China.
- [Bla05c] Ł. Błaszak, "Verification of the CE Results of Nokia in SVC", ISO/IEC JTC1/SC29/WG11 MPEG04/M11733, 2005, Hong Kong, China.
- [Bla05d] Ł. Błaszak, M. Domański, "A Simple Technique for SNR-Scalable Video Coding Using Spiral Re-ordering", IEEE International Conference on Multimedia & Expo ICME 2005, Amsterdam, The Netherlands, 2005.
- [Bla05e] Ł. Błaszak, M. Domański, "Spiral Coding Order of Macroblocs with Applications SNR-Scalable Video Compression", International Conference on Image Processing, Genova, Switzerland, 2005.
- [Bla05f] Ł. Błaszak, M. Domański, "Results of SVC CE 8 on Spiral Scan", ISO/IEC JTC1/SC 29/WG11, MPEG/M11998, Busan, South Korea, 2005.
- [Bla05g] Ł. Błaszak, M. Domański, "Comparison of Spiral Scan and Raster Scan", ISO/IEC JTC1/SC 29/WG11, MPEG/M11998, Busan, South Korea, 2005.

References used in mention in the dissertation

- [And04] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Schaar, J. Cornelis and P. Schelkens, "In-Band Motion Compensated Temporal Filtering," *Signal Processing: Image Communication*, vol. 19, no. 7, pp. 653-673, August 2004.
- [And97] J. Andrew, "A Simple and Efficient Hierarchical Image Coder", *ICIP'97*, Vol. 3, pp. 658-661, 1997.
- [Ben98] M. Bénétière and C. Dufour, "Matching Pursuits Residual Coding for Video Fine Granular Scalability", *ISO/IEC JTC1/SC29/WG11, MPEG98/M4008*, Oct. 1998.
- [Cal98] R. Calderbank, I. Daubechies, W. Sweldens, B.-L. Yeo, "Wavelet Transforms that Map Integer to Integer", *Applied and Computational Harmonic Analysis*, Vol. 5, pp. 332-369, July 1998.
- [Che98] S.-C. S. Cheung and A. Zakhor, "Matching Pursuit Coding for Fine Granular Video Scalability", *ISO/IEC JTC1/SC29/WG11, MPEG98/M3991*, Oct. 1998.
- [Che04] S-R Chen, C-P Chang, and C-W Lin, "MPEG-4 FGS Performance Improvement Using Adaptive Inter-Layer Prediction", *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing*, May 2004, Montreal, Canada.
- [Cho96] S. J. Choi, "Three-Dimensional Subband/Wavelet Coding of Video with Motion Compensation", PhD thesis, Rensselaer Polytechnic Institute, 1996.
- [Cho99] S.-J. Choi and J. W. Woods, "Motion-Compensated 3-D Subband Coding of Video", *IEEE Transactions on Image Processing*, 8(2):155-167, February, 1999.
- [Dom00a] M. Domański, A. Łuczak, S. Maćkowiak, "Spatio-Temporal Scalability for MPEG", *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 10 No. 7 October 2000, pp. 1088-1093.
- [Dom00b] M. Domański, A. Łuczak, S. Maćkowiak, "On Improving MPEG Spatial Scalability", *IEEE International Conference on Image Processing*, Vancouver, BC, Canada, 10-13 IX 2000 r., vol. II, pp. 848-851.
- [Dom00c] M. Domański, A. Łuczak, S. Maćkowiak, "Spatio-Temporal Scalability for MPEG-2 Video Coding", *Proceedings of Signal Processing X: Theories and Applications*, Eusipco 2000, Tampere 2000.
- [Dom00d] M. Domański, A. Dobrogowski, W. Kabaciński, M. Bartkowiak, J. Kleban, A. Łuczak, S. Maćkowiak, "Multimedialne Systemy

Telekomunikacyjne”, Instytut Elektroniki I Telekomunikacji, Politechnika Poznańska, Poznań, Marzec 2000.

- [Dom01] M. Domański, S. Maćkowiak, "Modified MPEG-2 Video Coders with Efficient Multi-Layer Scalability", Proceedings of 2001 International Conference of Image Processing ICIP, vol. II pp. 1033-1036, Thessaloniki 2001, October 2001.
- [Fli03] M. Flierl, B. Girod, "Generalized B Pictures and the Draft H.264/ACV Video-Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 587-598, July 2003.
- [He01] Y. He, R. Yan, F. Wu, S. Li, "H.26L-Based Fine Granularity Scalable Video Coding", ISO/IEC JTC1/SC 29/WG11, MPEG2001/ M7788, December 2001.
- [Hor99] U. Horn, K. Stuhlmüller, M. Link, B. Girod, "Robust Internet Video Transmission Based on Scalable Coding and Unequal Error Protection", Image Communication, Special Issue on Real-time Video over the Internet, pp. 77-94, 15(1-2), September, 1999.
- [Hor03] M. Horwitz, A. Jocht, F. Kossentini, A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 704-717, July 2003.
- [Hsi00] S.-T. Hsiang, J. T. Woods, "Embedded Image Coding Using ZeroBlocks at Subband/Wavelet Coefficients and Context Modeling", ISCAS 2000, vol. 3, pp. 662-665, 2000.
- [Hsi01] S. T. Hsiang, J. W. Woods, "Embedded Video Coding Using Invertible Motion Compensated 3-D Subband/Wavelet Filter Bank", Signal Processing: Image Communication 16 (2001), pp. 705-724, 2001.
- [Hua04] H-C Huang, W-H Peng, Y-C Lin, C-N Wang, T. Chiang, H-M Hang, "Response to CfP on Scalable Video Coding Technology: Proposal S07 – A Robust Scalable Video Coding Technique", ISO/IEC JTC1/SC 29/WG11, MPEG2004/M10569/S07, Munich, March 2004.
- [Huf52] D. Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proceedings of the IRE 40, pp. 1098-1101, 1952.
- [ISO90] ITU-T, "Video Codec for Audiovisual Services at <64 kbits/s.", ITU-T Rec. H.261 v1: Nov. 1990, v2: Mar. 1993.
- [ISO93] ISO/IEC JTC 1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s – Part 2: Video.", ISO/IEC 11172 (MPEG-1), Nov. 1993.

- [ISO94] “Generic Coding of Moving Pictures and Associated Audio Information, Part-2 Video”, ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and 13818-2 (MPEG-2), Nov. 1994.
- [ISO96] ITU-T, “Video Coding for Narrow Telecommunication Channels at < 64kbps/s”, Recommendation H.263, 1996.
- [ISO99] ISO/IEC, “Coding of Audio-Visual Objects – Part-2: Visual”, ISO/IEC 14496-2 (MPEG-4 Part 2), Jan. 1999 (with several subsequent amendments and corrigenda).
- [ISO00] “JPEG2000 Image Coding System”, ISO/IEC IS 15444-1, 2000.
- [ISO02] ITU-T Recommendation BT.500-11, “Methodology for the Subjective Assessment of the Quality of Television Pictures”, Geneva, 2002.
- [ISO04a] C. Tillier, B. Pesquet-Popescu, “Motion-Compensated Temporal Codec – A Standalone Tool in Response to the CfP on SVC Technology”, ISO/IEC JTC1/SC29/WG11 M10538, March 2004, Munchen.
- [ISO04b] M. Wien, T. Rusert, K. Hanke, “RWTH Proposal for Scalable Video Coding Technology”, ISO/IEC JTC1/SC29/WG11 M10569/S16, March 2004, Munchen.
- [ISO04c] H. Schwarz, D. Marpe, T. Wiegand, “Scalable Extension of H.264/AVC”, ISO/IEC JTC1/SC29/WG11 MPEG04/M10569/S03, March 2004, Munchen.
- [ISO04d] J. Reichel, M. Wien, “Scalable Video Model Version 2.0”, ISO/IEC JTC1/SC29/WG11 MPEG2004/N6520, Redmond, July 2004.
- [ISO04e] T. Kimoto and Y. Miyamoto, “Multi-Resolution Motion-Compensated Temporal Filtering for 3-D Wavelet Coding”, ISO/IEC JTC1/SC29/WG11 MPEG2004/M10569/S09, Munich, March 2004.
- [ISO04f] Nicola Adami, Michele Brescianini, Riccardo Leonardi, Alberto Signoroni, “SVC CE1: Tool - a Native Spatially Scalable Approach to SVC”, ISO/IEC JTC1/SC29/WG11 MPEG2004/M11368, Palma, October 2004.
- [ISO04g] “Subjective test results for the CfP on Scalable Video Coding Technology”, ISO/IEC JTC1/SC29/WG11 MPEG2004/M110737, Munich, March 2004.
- [ISO-AVC] “Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding”, ITU-T and ISO/IEC FDIS, ITU-T Recommendation H.264 and 14496-10/AVC, ISO/IEC JTC1/SC29/WG11 MPEG05/N7081, Busan, April 2005.

- [ISO-JSVM] “Joint Scalable Video Model JSVM-6”, ISO/IEC JTC 1/SC 29/WG11 N8015, Montreux, Switzerland, April 2006.
- [Kar03] M. Karczewicz, R. Kurceren, “The SP- and SI-Frames Design for H.264/AVC”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 637-645, July 2003.
- [Kar88] G. Karlsson, M. Vetterli, “Three Dimensional Subband Coding of Video”, Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1100-1103, April 1988.
- [Kim03] T. Kim, M. H. Ammar, “Optimal Quality for MPEG-4 Fine-Grained Scalable Video”, IEEE INFOCOM, 2003.
- [Kro90] T. Kronander, “New Results on 3-Dimensional Motion Compensated Subband Coding”, Proceedings Picture Coding Symposium, March 1990.
- [Lap03] V. Lappalainen, A. Hallapuro, T. D. Hämäläinen, “Complexity of Optimized H.26L Video Decoder Implementaion”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 717-726, July 2003.
- [Li98] J. Liang, J. Yu, Y. Wang, M. Srinath, and M. Zhou, “Fine Granularity Scalable Video Coding Using Combination of MPEG4 Video Objects and Still Texture Objects”, ISO/IEC JTC1/SC29/WG11, MPEG98/M4025, Oct. 1998.
- [Li01] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, IEEE Transactions on Circuits and Systems for Video Technology Vol. 11, no. 3, March 2001.
- [Li04] X. Li, “Scalable Video Compression via Overcomplete Motion Compensated Wavelet Coding”, Signal Processing: Image Communication, vol. 19, no. 7, pp. 637-651, 2004.
- [Lin05] J-L Lin, W-L Hwang, S-C Pei, “SNR Scalability Based on Bitplane Coding of Matching Pursuit Atoms at Low Bit Rates: Fine-Grained and Two-Layer”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 1, January 2005.
- [Lis03] P. List, A. Joch, J. Lainema, G. Bjøntegaard, M. Karczewicz, “Adaptive Deblocking Filter”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 614-620, July 2003.
- [Mac02] S. Maćkowiak, “Scalable Coding of Digital Video”, Doctoral Dissertation, Poznań, 2002.

- [Mal03] H. S. Malvar, A. Hallapuro, M. Karczewicz, L. Kerofsky, "Low-Complexity Transform and Quantization in H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 598-604, July 2003.
- [Mar03] D. Marpe, H. Schwarz, T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 620-637, July 2003.
- [Ohm93] J.-R. Ohm, "Advanced Packet Video Coding Based on Layered VQ and SBC Techniques", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, pp. 208-221, June 1993.
- [Ohm94] J.-R. Ohm, "Three Dimensional Subband Coding with Motion Compensation", IEEE Transactions on Image Processing, Vol. 3, no. 5, pp. 559-571, September 1994.
- [Ohm01] J.-R. Ohm, M. Beermann, "Status of Scalable Technology in Video Coding", Doc. ISO/IEC JTC1/SC29/WG11 MPEG01/M7483, Sydney, July 2001.
- [Ohm02] J.-R. Ohm, T. Ebrahimi, "Report of Ad hoc Group on Exploration of Interframe Wavelet Technology in Video", ISO/IEC JTC1/SC29/WG11 MPEG02/M8359, Fairfax, May 2002.
- [Ost04] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity", IEEE Circuits and Systems Magazine, pp. 7-28, 2004.
- [Par00] H. W. Park and H. S. Kim, "Motion Estimation Using Low-Band-Shift Method for Wavelet-Based Moving-Picture Coding," IEEE Trans. Image Processing, vol. 9, No. 4, pp. 577-587, April 2000.
- [Par02] G. H. Park, Y. L. Lee, "Water Ring Scan Method for H.26L Based FGS", JVT-B094, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), Geneva, Jan.29 – Feb. 1, 2002.
- [Pea98] W. A. Pearlman, B.-J. Kim, Z. Xiong, "Embedded Video Subband Coding with 3D SPIHT", Wavelet Image and Video Compression, Kluwer Academic Publishers, pp. 397-432, 1998.
- [Rad99a] H. Radha, Y. Chen, "Fine-Granular-Scalable Video for Packet Networks", Packet Video '99, New York, April 1999.

- [Rad99b] H. Radha, Y. Chen, K. Parthasarathy, R. Cohen, "Scalable Internet Video Using MPEG-4", *Signal Processing: Image Communication*, vol. 15, pp. 95-126, September 1999.
- [Rad01] H. M. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming Over IP", *IEEE Transactions on Multimedia*, Vol. 3, No. 1, pp. 53-68, March 2001.
- [Ram99] G. Ramponi, "Warped Distance for Space-Variant Linear Image Interpolation", *IEEE Transactions on Image Processing*, Vol. 8, pp. 629-639, May 1999.
- [Rib03] J. Ribas-Corbera, P. A. Chou, G. J. Regunathan, "A Generalized Hypothetical Reference Decoder for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard*, Vol. 13, No. 7, pp. 674-688, July 2003.
- [Ric03] I. E. G. Richardson, "H .264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia", John Wiley & Sons, Ltd. 2003.
- [Rid04] J. Ridge, Y. Bao, M. Karczewicz, "Scalable Video Coder with AVC-like Properties", ISO/IEC JTC1/SC 29/WG11, MPEG2004/M10732, Munich, March 2004.
- [Rid05] J. Ridge, Y. Bao, M. Karczewicz, X. Wang, "Cyclical Block Coding for FGS", ISO/IEC JTC1/SC 29/WG11, MPEG2005/ M11509, Hong Kong, January 2005.
- [Sai96] A. Said. W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243-250, June 1996.
- [Sch01] M. van der Schaar, H. Radha, "A Hybrid Temporal-SNR Fine-Granular Scalability for Internet Video", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, March 2001.
- [Sch02] M. van der Schaar, H. Radha, "Adaptive Motion-Compensated Fine-Granular-Scalability (AMC-FGS) for Wireless Video", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 6, June 2002.
- [Sch03] H. Schwarz, D. Marpe, T. Wiegand, „SNR-scalable Extension of H.264/AVC", JVT-J035, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), Waikoloa, Hawaii, USA, 8-12 December, 2003.
- [Sch04] H. Schwarz, D. Marpe, T. Wiegand, „Scalable Extension of H.264/AVC", ISO/IEC JTC1/SC 29/WG11, MPEG2004/M10569/S03, Munich, March 2004.

- [Schu98] B. Schuster, "Fine Granular Scalability with Wavelet Coding.", ISO/IEC JTC1/SC29/WG11, MPEG98/M4021, Oct. 1998.
- [Sha93] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE Transactions on Signal Processing, Vol. 41, no. 12, pp. 2445-3462, December, 1993.
- [SMPTE05] "Proposed SMPTE Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process", SMPTE Draft Standard for Television, SMPTE 421M, August 2005.
- [Sri04] S. Srinivasan, P. J. Hsu, T. Holcomb, K. Mukerjee, S. L. Regunathan, B. Lin, J. Liang, M-C Lee, J. Ribas-Corbera, "Windows Media Video 9: Overview and Applications", Signal Processing: Image Communication 19, pp. 851-875, 2004.
- [Sto03] T. Stockhammer, M. M. Hannuksela, T. Wiegand, "H.264/AVC in Wireless Environments", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 657-674, July 2003.
- [Sul04] G. J. Sullivan, P. Topiwala, A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions", SPIE Conference on Applications of Digital Image Processing XXVII Special Session on Advances in the New Emerging Standard: H.264/AVC, August, 2004.
- [Sun02] X. Sun, F. Wu, S. Li, "The Description and Proposed Syntax of JVT-based FGS", ISO/IEC JTC1/SC 29/WG11, MPEG2002/ M8204, March 2002.
- [Sun04] X. Sun, Y. Zhou, Y. Wang, G. Sullivan, M-C Lee, F. Wu, S. Li, "Progressive Fine Granularity Scalable (PFGS) video coding", ISO/IEC JTC1/SC 29/WG11, MPEG2004/M10569/S06, Munich, March 2004.
- [Tau94] D. Taubman, A. Zakhor, "Multirate 3-D Subband Coding of Video", IEEE Transactions on Image Processing, Vol. 8, no. 7, July 1999.
- [Tau02] D. Taubman, M. Marcellin, "JPEG2000 Image Compression Fundamentals, Standards and Practice", Kluwer 2002
- [Tra03] T. D. Tran, J. Liang, C. Tu, "Lapped Transform via Time-Domain Pre- and Post-Filtering", IEEE Trans. Signal Processing 51(6), pp. 1557-1571, June 2003.
- [Ugu03] K. Ugur, P. Nasiopoulos, "Design Issues and Proposal for H.264 Based FGS", ISO/IEC JTC1/SC 29/WG11, MPEG2003/ M9505, Pattaya, Thai, March 2003.

- [VCEG-L13] D. Marpe, G. Blättermann, T. Wiegand, “Adaptive Codes for H.26L“, document VCEG-L13, ITU-T Video Coding Experts Group, Eibsee, Germany, 09-12 January 2001.
- [VCEG-M20] S. Sun and S. Lei, “Loop Filter with Skip Mode,” document VCEG-M20, ITU-T Video Coding Experts Group (VCEG), Austin, USA, 2-4 April 2001.
- [VCEG-M59] D. Marpe, G. Blättermann, G. Heising, T. Wiegand, “Further Results for CABAC Entropy Coding Scheme”, document VCEG-M59, ITU-T Video Coding Experts Group, Austin, Texas, USA, 2-4 April, 2001.
- [VCEG-N54] G. Conklin, “New Intra Prediction Modes”, Video Coding Experts Group (VCEG), VCEG-N054, Santa Barbara, 24-27 Sep 2001.
- [VCEG-O31] M. Wien, “Intra Coding using Variable Block Sizes”, Video Coding Experts Group, VCEG-O31, Pattaya, Thailand, 4-6 Dec., 2001.
- [VCEG-O48] Z. Xue, S.-M. Shen, T.-W. Foo, CJ Lee, S. Kadono, “An Automatic Mode Decision Method for Intra Frame Coding and Decoding”, Video Coding Experts Group, VCEG-O48, Pattaya, Thailand, 4-6 Dec., 2001.
- [JVT-B039] J. Jung, E. Lesellier, Y. Le Maguet, C. Miró, J. Gobert, “PDS, a Low Complexity deblocking for JVT”, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-B037, Geneva, Jan. 29 - Feb. 1, 2002.
- [Wan99] A. Wang, Z. Xiong, P. A. Chou, and S. Mehrotra, “Three-Dimensional Wavelet Coding of Video with Global Motion Compensation”, Data Compression Conference, pp. 404-413, Snowbird, UT, 1999.
- [Wed03] T. Wedi, H. G. Musmann, “Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 577-587, July 2003.
- [Wen03] S. Wenger, “H.264/AVC Over IP”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 645-657, July 2003.
- [Wie03a] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Lutra, „Overview of the H.264/AVC Video Coding Standard”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 560-577, July 2003.
- [Wie03b] M. Wien, “Variable Block-Size Transform for H.264/AVC”, IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 604-614, July 2003.

- [Wie03c] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards", IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on the H.264/AVC Video Coding Standard, Vol. 13, No. 7, pp. 688-704, July 2003.
- [Win99] S. Winkler, "Issue in Vision Modeling for Perceptual Video Quality Assessment", Signal Processing 78, pp. 231-252, 1999.
- [Wit87] I.H Witten, R.M. Neal, J.G. Cleary, "Arithmetic Coding for Data Compression", Comm. ACM 30, pp. 520-540, 1987.
- [Woo02] J. W. Woods, P. Chen, "Improved MC-EZBC with Quarter-pixel Motion Vectors", ISO/IEC JTC1/SC29/WG11 MPEG02/M8366, Fairfax, VA, May 2002.
- [Wu00] F. Wu, S. Li and Y.-Q. Zhang, "DCT-Prediction Based Progressive Fine Granularity Scalability Coding", ICIP 2000, Vancouver, Canada, Vol. 3, pp. 556-559, Sep 10-13, 2000.
- [Wu01] F. Wu, S. Li, X. Sun, R. Yan, Y.-Q. Zhang, "Macroblock-Based Progressive Fine Granularity Scalable Coding", ISO/IEC JTC1/SC29/WG11, MPEG2001/N6779, Pisa, January, 2001.
- [Xio04] R. Xiong, J. Xu, F. Wu, S. Li, Y.-Q. Zhang, "Spatial Scalability in 3D Wavelet Coding with Spatial Domain MCTF Encoder", Proceedings of Picture Coding Symposium 2004, San Francisco, CA, USA, Dec 2004.
- [Xio05a] R. Xiong, J. Xu, F. Wu, S. Li, "Studies on Spatial Scalable Frameworks for Motion Aligned 3D Wavelet Video Coding", Visual Communication and Image Processing 2005, Proceedings of SPIE, vol. 5960, pp. 189 – 200, Bellingham, WA, 2005.
- [Xio05b] R. Xiong, J. Xu, F. Wu, S. Li, Y.-Q. Zhang, "Optimal Subband Rate Allocation for Spatial Scalability in 3D Wavelet Video Coding with Motion Aligned Temporal Filtering", SPIE VCIP 2005, Beijing, July 2005.
- [Xu02] J. Xu, Z. Xiong, S. Li, Y. Q. Zhang, "Memory-Constrained 3D Wavelet Transform for Video Coding Without Boundary Effect", IEEE Transactions on Circuits and Systems for Video Technology 12(10):850-856, October, 2002.
- [Yan01] R. Yan, F. Wu, S. Li, Y-Q Zhang, "Macroblock-based Progressive Fine Granularity Spatial Scalability (mb-PFGSS)", ISO/IEC JTC1/SC 29/WG11, MPEG2001/ M7112, Munich, March 2001.

- [Ye03] J. C. Ye and M. van der Schaar, “Fully Scalable 3D Overcomplete Wavelet Video Coding Using Adaptive Motion Compensated Temporal Filtering”, Proc. VCIP2003, Vol. 5150, pp. 1169-1180, Lugano, Switzerland, 2003.
- [Zho03] J. Zhou, H-R. Shao, C. Shen, M-T. Sun, “FGS Enhancement Layer Truncation with Minimized Intra-Frame Quality Variation”, IEEE International Conference on Multimedia and Expo (ICME), Vol. 2, pp. 361-364, July 2003.
- [Ziv77] J. Ziv, A. Lempel, “A Universal Algorithm for Sequential Data Compression”, IEEE Transactions on Information Theory, IT-23, pp. 337-343, 1977.
- [Ziv78] J. Ziv, A. Lempel, “Compression of Individual Sequences via Variable-Rate Coding”, IEEE Transactions on Information Theory, IT-24, pp. 530-536, 1978.
- [http01] Official Windows Media Web site,
<http://www.microsoft.com/windows/windowsmedia/default.asp>.

Annex A

Intra prediction

A.1. Intra prediction for chrominance blocks

Here, in this Section the algorithm of spatial prediction of chrominance samples for spiral scan of macroblocks is proposed. The values of prediction samples for standard chrominance prediction modes are derived as follows:

- horizontal: $pred_c(x, y) = p(-1, y)$, with $x, y = 0 .. 7$
- vertical: $pred_c(x, y) = p(x, -1)$, with $x, y = 0 .. 7$
- plane: $pred_c(x, y) = clip((a + b * (x - 3) + c * (y - 3) + 16) \gg 5)$, with $x, y = 0 .. 7$,

$$\text{where } clip(x) = \begin{cases} 0 & x < 0 \\ 255; & x > 255 \\ x; & \text{otherwise} \end{cases},$$

$$a = 16 * (p(-1, 7) + p(7, -1))$$

$$\text{and } b = (17 * H + 16) \gg 5,$$

$$c = (17 * V + 16) \gg 5$$

$$\text{where } H = \sum_{\alpha=0}^3 (\alpha + 1) * (p(4 + \alpha, -1) - p(2 - \alpha, -1))$$

$$V = \sum_{\beta=0}^3 (\beta + 1) * (p(-1, 4 + \beta) - p(-1, 2 - \beta))$$

- DC:

- for x=0..3 and y=0..3

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(\alpha, -1) + \sum_{\beta=0}^3 p(-1, \beta) + 4) \gg 3; & p(0..3, -1) \wedge p(-1, 0..3) \rightarrow available \\ (\sum_{\alpha=0}^3 p(\alpha, -1) + 2) \gg 2; & p(0..3, -1) \rightarrow available \wedge p(-1, 0..3) \rightarrow unavailable \\ (\sum_{\beta=0}^3 p(-1, \beta) + 2) \gg 2; & p(0..3, -1) \rightarrow unavailable \wedge p(-1, 0..3) \rightarrow available \\ 128; & p(0..3, -1) \rightarrow unavailable \wedge p(-1, 0..3) \rightarrow unavailable \end{cases}$$

- for x=4..7 and y=0..3

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(\alpha, -1) + 2) \gg 2; & p(4..7, -1) \rightarrow available \\ (\sum_{\beta=0}^3 p(-1, \beta) + 2) \gg 2; & p(-1, 0..3) \rightarrow available \\ 128; & p(4..7, -1) \rightarrow unavailable \wedge p(-1, 0..3) \rightarrow unavailable \end{cases}$$

- for x=0..3 and y=4..7

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(\alpha, -1) + 2) \gg 2; & p(0..3, -1) \rightarrow available \\ (\sum_{\beta=4}^7 p(-1, \beta) + 2) \gg 2; & p(-1, 4..7) \rightarrow available \\ 128; & p(0..3, -1) \rightarrow unavailable \wedge p(-1, 4..7) \rightarrow unavailable \end{cases}$$

- for x=4..7 and y=4..7

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(\alpha, -1) + \sum_{\beta=4}^7 p(-1, \beta) + 4) \gg 3; & p(4..7, -1) \wedge p(-1, 4..7) \rightarrow available \\ (\sum_{\alpha=4}^7 p(\alpha, -1) + 2) \gg 2; & p(4..7, -1) \rightarrow available \wedge p(-1, 4..7) \rightarrow unavailable \\ (\sum_{\beta=4}^7 p(-1, \beta) + 2) \gg 2; & p(4..7, -1) \rightarrow unavailable \wedge p(-1, 4..7) \rightarrow available \\ 128; & p(4..7, -1) \rightarrow unavailable \wedge p(-1, 4..7) \rightarrow unavailable \end{cases}$$

And the transformed equations for modified neighborhood:

1. horizontal: $pred_c(x, 7 - y) = p(-1, 7 - y)$

vertical: $pred_c(x, 7 - y) = p(x, 8)$

then back to x, y coordinates by substitution $x = x$ and $y = 7 - y$

horizontal: $pred_c(x, y) = p(-1, y)$

vertical: $pred_c(x, y) = p(x, 8)$

plane: $pred_c(x, y) = clip((a + b * (x - 3) + c * (4 - y) + 16) >> 5)$

$$a = 16 * (p(-1, 0) + p(7, 8))$$

and $b = (17 * H + 16) >> 5$,

$$c = (17 * V + 16) >> 5$$

$$H = \sum_{\alpha=0}^3 (\alpha + 1) * (p(4 + \alpha, 8) - p(2 - \alpha, 8))$$

where

$$V = \sum_{\beta=0}^3 (\beta + 1) * (p(-1, 3 - \beta) - p(-1, 5 + \beta))$$

DC:

a. for $x=0..3$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(\alpha, 8) + \sum_{\beta=0}^3 p(-1, 7 - \beta) + 4) >> 3, p(0..3, 8) \wedge p(-1, 4..7) \rightarrow available \\ (\sum_{\alpha=0}^3 p(\alpha, 8) + 2) >> 2, p(0..3, 8) \rightarrow available, p(-1, 4..7) \rightarrow unavailable \\ (\sum_{\beta=0}^3 p(-1, 7 - \beta) + 2) >> 2, p(0..3, 8) \rightarrow unavailable, p(-1, 4..7) \rightarrow available \\ 128, p(0..3, 8) \rightarrow unavailable, p(-1, 4..7) \rightarrow unavailable \end{cases}$$

b. for $x=4..7$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(\alpha, 8) + 2) >> 2, p(4..7, 8) \rightarrow available \\ (\sum_{\beta=0}^3 p(-1, 7 - \beta) + 2) >> 2, p(-1, 4..7) \rightarrow available \\ 128, p(4..7, 8) \rightarrow unavailable, p(-1, 4..7) \rightarrow unavailable \end{cases}$$

c. for $x=0..3$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(\alpha, 8) + 2) >> 2, p(0..3, 8) \rightarrow available \\ (\sum_{\beta=4}^7 p(-1, 7 - \beta) + 2) >> 2, p(-1, 0..3) \rightarrow available \\ 128, p(0..3, 8) \rightarrow unavailable, p(-1, 0..3) \rightarrow unavailable \end{cases}$$

d. for $x=4..7$ and $y=4..7$

$$pred_c(x,y) = \begin{cases} (\sum_{\alpha=4}^7 p(\alpha,8) + \sum_{\beta=4}^7 p(-1,7-\beta) + 4) \gg 3, p(4..7,8) \wedge p(-1,0..3) \rightarrow available \\ (\sum_{\alpha=4}^7 p(\alpha,8) + 2) \gg 2, p(4..7,8) \rightarrow available, p(-1,0..3) \rightarrow unavailable \\ (\sum_{\beta=4}^7 p(-1,7-\beta) + 2) \gg 2, p(4..7,8) \rightarrow unavailable, p(-1,0..3) \rightarrow available \\ 128, p(4..7,8) \rightarrow unavailable, p(-1,0..3) \rightarrow unavailable \end{cases}$$

2. horizontal: $pred_c(y,x) = p(y,-1)$

vertical: $pred_c(y,x) = p(-1,x)$

then back to x, y coordinates by substitution $x = y$ and $y = x$

horizontal: $pred_c(x,y) = p(x,-1)$

vertical: $pred_c(x,y) = p(-1,y)$

plane: $pred_c(x,y) = clip((a + b * (y - 3) + c * (x - 3) + 16) \gg 5)$

$$a = 16 * (p(7,-1) + p(-1,7))$$

and $b = (17 * H + 16) \gg 5$,

$$c = (17 * V + 16) \gg 5$$

$$H = \sum_{\alpha=0}^3 (\alpha + 1) * (p(-1,4 + \alpha) - p(-1,2 - \alpha))$$

where

$$V = \sum_{\beta=0}^3 (\beta + 1) * (p(4 + \beta, -1) - p(2 - \beta, -1))$$

DC:

a. for $x=0..3$ and $y=0..3$

$$pred_c(x,y) = \begin{cases} (\sum_{\alpha=0}^3 p(-1,\alpha) + \sum_{\beta=0}^3 p(\beta,-1) + 4) \gg 3, p(-1,0..3) \wedge p(0..3,-1) \rightarrow available \\ (\sum_{\alpha=0}^3 p(-1,\alpha) + 2) \gg 2, p(-1,0..3) \rightarrow available, p(0..3,-1) \rightarrow unavailable \\ (\sum_{\beta=0}^3 p(\beta,-1) + 2) \gg 2, p(-1,0..3) \rightarrow unavailable, p(0..3,-1) \rightarrow available \\ 128, p(-1,0..3) \rightarrow unavailable, p(0..3,-1) \rightarrow unavailable \end{cases}$$

b. for $x=4..7$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(-1, \alpha) + 2) \gg 2, p(-1, 4..7) \rightarrow available \\ (\sum_{\beta=0}^3 p(\beta, -1) + 2) \gg 2, p(0..3, -1) \rightarrow available \\ 128, p(-1, 4..7) \rightarrow unavailable, p(0..3, -1) \rightarrow unavailable \end{cases}$$

c. for $x=0..3$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(-1, \alpha) + 2) \gg 2, p(-1, 0..3) \rightarrow available \\ (\sum_{\beta=4}^7 p(\beta, -1) + 2) \gg 2, p(4..7, -1) \rightarrow available \\ 128, p(-1, 0..3) \rightarrow unavailable, p(4..7, -1) \rightarrow unavailable \end{cases}$$

d. for $x=4..7$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(-1, \alpha) + \sum_{\beta=4}^7 p(\beta, -1) + 4) \gg 3, p(-1, 4..7) \wedge p(4..7, -1) \rightarrow available \\ (\sum_{\alpha=4}^7 p(-1, \alpha) + 2) \gg 2, p(-1, 4..7) \rightarrow available, p(4..7, -1) \rightarrow unavailable \\ (\sum_{\beta=4}^7 p(\beta, -1) + 2) \gg 2, p(-1, 4..7) \rightarrow unavailable, p(4..7, -1) \rightarrow available \\ 128, p(-1, 4..7) \rightarrow unavailable, p(4..7, -1) \rightarrow unavailable \end{cases}$$

3. horizontal: $pred_c(7-x, y) = p(8, y)$

vertical: $pred_c(7-x, y) = p(7-x, -1)$

then back to x, y coordinates by substitution $x = 7-x$ and $y = y$

horizontal: $pred_c(x, y) = p(8, y)$

vertical: $pred_c(x, y) = p(x, -1)$

plane: $pred_c(x, y) = clip((a + b * (4 - x) + c * (y - 3) + 16) \gg 5)$

$$a = 16 * (p(8, 7) + p(0, -1))$$

and $b = (17 * H + 16) \gg 5$,

$$c = (17 * V + 16) \gg 5$$

$$H = \sum_{\alpha=0}^3 (\alpha + 1) * (p(3 - \alpha, -1) - p(5 + \alpha, -1))$$

where

$$V = \sum_{\beta=0}^3 (\beta + 1) * (p(8, 4 + \beta) - p(8, 2 - \beta))$$

DC:

a. for $x=0..3$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(7-\alpha, -1) + \sum_{\beta=0}^3 p(8, \beta) + 4) \gg 3; & p(7..4, -1) \wedge p(8, 0..3) \rightarrow available \\ (\sum_{\alpha=0}^3 p(7-\alpha, -1) + 2) \gg 2; & p(7..4, -1) \rightarrow available \wedge p(8, 0..3) \rightarrow unavailable \\ (\sum_{\beta=0}^3 p(8, \beta) + 2) \gg 2; & p(7..4, -1) \rightarrow unavailable \wedge p(8, 0..3) \rightarrow available \\ 128 & p(7..4, -1) \rightarrow unavailable \wedge p(8, 0..3) \rightarrow unavailable \end{cases}$$

b. for $x=4..7$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(7-\alpha, -1) + 2) \gg 2, & p(3..0, -1) \rightarrow available \\ (\sum_{\beta=0}^3 p(8, \beta) + 2) \gg 2, & p(8, 0..3) \rightarrow available \\ 128, & p(3..0, -1) \rightarrow unavailable, p(8, 0..3) \rightarrow unavailable \end{cases}$$

c. for $x=0..3$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(7-\alpha, -1) + 2) \gg 2, & p(7..4, -1) \rightarrow available \\ (\sum_{\beta=4}^7 p(8, \beta) + 2) \gg 2, & p(8, 4..7) \rightarrow available \\ 128, & p(7..4, -1) \rightarrow unavailable, p(8, 4..7) \rightarrow unavailable \end{cases}$$

d. for $x=4..7$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(7-\alpha, -1) + \sum_{\beta=4}^7 p(8, \beta) + 4) \gg 3, & p(3..0, -1) \wedge p(8, 4..7) \rightarrow available \\ (\sum_{\alpha=4}^7 p(7-\alpha, -1) + 2) \gg 2, & p(3..0, -1) \rightarrow available \wedge p(8, 4..7) \rightarrow unavailable \\ (\sum_{\beta=4}^7 p(8, \beta) + 2) \gg 2, & p(3..0, -1) \rightarrow unavailable \wedge p(8, 4..7) \rightarrow available \\ 128, & p(3..0, -1) \rightarrow unavailable \wedge p(8, 4..7) \rightarrow unavailable \end{cases}$$

4. horizontal: $pred_c(7-y, 7-x) = p(7-y, 8)$

vertical: $pred_c(7-y, 7-x) = p(8, 7-x)$

then back to x, y coordinates by substitution $x = 7-y$ and $y = 7-x$

horizontal: $pred_c(x, y) = p(x, 8)$

vertical: $pred_c(x, y) = p(8, y)$

plane: $pred_c(x, y) = clip((a + b * (4 - y) + c * (4 - x) + 16) \gg 5)$

$$a = 16 * (p(0, 8) + p(8, 0))$$

and $b = (17 * H + 16) \gg 5$,

$$c = (17 * V + 16) \gg 5$$

where

$$H = \sum_{\alpha=0}^3 (\alpha + 1) * (p(8, 3 - \alpha) - p(8, 5 + \alpha))$$

$$V = \sum_{\beta=0}^3 (\beta + 1) * (p(3 - \beta, 8) - p(5 + \beta, 8))$$

DC:

a. for $x=0..3$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(8,7-\alpha) + \sum_{\beta=0}^3 p(7-\beta,8) + 4) \gg 3, p(8,4..7) \wedge p(4..7,8) \rightarrow available \\ (\sum_{\alpha=0}^3 p(8,7-\alpha) + 2) \gg 2, p(8,4..7) \rightarrow available, p(4..7,8) \rightarrow unavailable \\ (\sum_{\beta=0}^3 p(7-\beta,8) + 2) \gg 2, p(8,4..7) \rightarrow unavailable, p(4..7,8) \rightarrow available \\ 128, p(8,4..7) \rightarrow unavailable, p(4..7,8) \rightarrow unavailable \end{cases}$$

b. for $x=4..7$ and $y=0..3$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(8,7-\alpha) + 2) \gg 2, p(8,0..3) \rightarrow available \\ (\sum_{\beta=0}^3 p(7-\beta,8) + 2) \gg 2, p(4..7,8) \rightarrow available \\ 128, p(8,0..3) \rightarrow unavailable, p(4..7,8) \rightarrow unavailable \end{cases}$$

c. for $x=0..3$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=0}^3 p(8,7-\alpha) + 2) \gg 2, p(8,4..7) \rightarrow available \\ (\sum_{\beta=4}^7 p(7-\beta,8) + 2) \gg 2, p(0..3,8) \rightarrow available \\ 128, p(8,4..7) \rightarrow unavailable, p(0..3,8) \rightarrow unavailable \end{cases}$$

d. for $x=4..7$ and $y=4..7$

$$pred_c(x, y) = \begin{cases} (\sum_{\alpha=4}^7 p(8,7-\alpha) + \sum_{\beta=4}^7 p(7-\beta,8) + 4) \gg 3, p(8,0..3) \wedge p(0..3,8) \rightarrow available \\ (\sum_{\alpha=4}^7 p(8,7-\alpha) + 2) \gg 2, p(8,0..3) \rightarrow available, p(0..3,8) \rightarrow unavailable \\ (\sum_{\beta=4}^7 p(7-\beta,8) + 2) \gg 2, p(8,0..3) \rightarrow unavailable, p(0..3,8) \rightarrow available \\ 128, p(8,0..3) \rightarrow unavailable, p(0..3,8) \rightarrow unavailable \end{cases}$$

A.2. Intra prediction for luminance 16x16 pixel blocks

Here, in this Section the algorithm of spatial prediction of luminance samples for spiral scan of macroblocks is proposed. The values of prediction samples for standard luminance 16x16 pixel block prediction modes are derived as follows:

- horizontal: $pred_L(x, y) = p(-1, y)$, with $x, y = 0 \dots 15$
- vertical: $pred_L(x, y) = p(x, -1)$, with $x, y = 0 \dots 15$
- plane: $pred_L(x, y) = clip((a + b * (x - 7) + c * (y - 7) + 16) \gg 5)$, with $x, y = 0 \dots 15$,

$$\text{where } clip(x) = \begin{cases} 0; & x < 0 \\ 255; & x > 255, \\ x; & \text{otherwise} \end{cases}$$

$$a = 16 * (p(-1, 15) + p(15, -1))$$

$$\text{and } b = (5 * H + 32) \gg 6,$$

$$c = (5 * V + 32) \gg 6$$

$$H = \sum_{\alpha=0}^7 (\alpha + 1) * (p(8 + \alpha, -1) - p(6 - \alpha, -1))$$

where

$$V = \sum_{\beta=0}^7 (\beta + 1) * (p(-1, 8 + \beta) - p(-1, 6 - \beta))$$

- DC:

$$pred_L(x, y) = \begin{cases} \left(\sum_{\alpha=0}^{15} p(\alpha, -1) + \sum_{\beta=0}^{15} p(-1, \beta) + 16 \right) \gg 5, p(0..15, -1) \wedge p(-1, 0..15) \rightarrow \text{available} \\ \left(\sum_{\alpha=0}^{15} p(\alpha, -1) + 8 \right) \gg 4, p(0..15, -1) \rightarrow \text{available}, p(-1, 0..15) \rightarrow \text{unavailable} \\ \left(\sum_{\beta=0}^{15} p(-1, \beta) + 8 \right) \gg 4, p(0..15, -1) \rightarrow \text{unavailable}, p(-1, 0..15) \rightarrow \text{available} \\ 128, p(0..15, -1) \rightarrow \text{unavailable}, p(-1, 0..15) \rightarrow \text{unavailable} \end{cases}$$

And the transformed equations for modified neighborhood:

- horizontal: $pred_L(x, 15 - y) = p(-1, 15 - y)$

- vertical: $pred_L(x, 15 - y) = p(x, 16)$

then back to x, y coordinates by substitution $x = x$ and $y = 15 - y$

- horizontal: $pred_L(x, y) = p(-1, y)$

- vertical: $pred_L(x, y) = p(x, 16)$

- plane: $pred_L(x, y) = clip((a + b * (x - 7) + c * (8 - y) + 16) \gg 5)$

where $a = 16 * (p(-1,0) + p(15,16))$
 $b = (5 * H + 32) \gg 6$,
 $c = (5 * V + 32) \gg 6$

where $H = \sum_{\alpha=0}^7 (\alpha + 1) * (p(8 + \alpha, 16) - p(6 - \alpha, 16))$

$V = \sum_{\beta=0}^7 (\beta + 1) * (p(-1, 7 - \beta) - p(-1, 9 + \beta))$

DC:

$$pred_L(x, y) = \begin{cases} (\sum_{\alpha=0}^{15} p(\alpha, 16) + \sum_{\beta=0}^{15} p(-1, 15 - \beta) + 16) \gg 5, p(0..15, 16) \wedge p(-1, 0..15) \rightarrow available \\ (\sum_{\alpha=0}^{15} p(\alpha, 16) + 8) \gg 4, p(0..15, 16) \rightarrow available, p(-1, 0..15) \rightarrow unavailable \\ (\sum_{\beta=0}^{15} p(-1, 15 - \beta) + 8) \gg 4, p(0..15, 16) \rightarrow unavailable, p(-1, 0..15) \rightarrow available \\ 128, p(0..15, 16) \rightarrow unavailable, p(-1, 0..15) \rightarrow unavailable \end{cases}$$

2. horizontal: $pred_L(y, x) = p(y, -1)$

vertical: $pred_L(y, x) = p(-1, x)$

then back to x, y coordinates by substitution $x = y$ and $y = x$

horizontal: $pred_L(x, y) = p(x, -1)$

vertical: $pred_L(x, y) = p(-1, y)$

plane: $pred_L(x, y) = clip((a + b * (y - 7) + c * (x - 7) + 16) \gg 5)$

$$a = 16 * (p(15, -1) + p(-1, 15))$$

where $b = (5 * H + 32) \gg 6$,

$$c = (5 * V + 32) \gg 6$$

$$H = \sum_{\alpha=0}^7 (\alpha + 1) * (p(-1, 8 + \alpha) - p(-1, 6 - \alpha))$$

where

$$V = \sum_{\beta=0}^7 (\beta + 1) * (p(8 + \beta, -1) - p(6 - \beta, -1))$$

DC:

$$pred_L(x, y) = \begin{cases} (\sum_{\alpha=0}^{15} p(-1, \alpha) + \sum_{\beta=0}^{15} p(\beta, -1) + 16) \gg 5, p(-1, 0..15) \wedge p(0..15, -1) \rightarrow available \\ (\sum_{\alpha=0}^{15} p(-1, \alpha) + 8) \gg 4, p(-1, 0..15) \rightarrow available, p(0..15, -1) \rightarrow unavailable \\ (\sum_{\beta=0}^{15} p(\beta, -1) + 8) \gg 4, p(-1, 0..15) \rightarrow unavailable, p(0..15, -1) \rightarrow available \\ 128, p(-1, 0..15) \rightarrow unavailable, p(0..15, -1) \rightarrow unavailable \end{cases}$$

3. horizontal: $pred_L(15-x, y) = p(16, y)$
 vertical: $pred_L(15-x, y) = p(15-x, -1)$

then back to x, y coordinates by substitution $x = 15-x$ and $y = y$

horizontal: $pred_L(x, y) = p(16, y)$

vertical: $pred_L(x, y) = p(x, -1)$

plane: $pred_L(x, y) = clip((a + b * (8 - x) + c * (y - 7) + 16) >> 5)$
 $a = 16 * (p(16, 15) + p(0, -1))$

where $b = (5 * H + 32) >> 6$,
 $c = (5 * V + 32) >> 6$

where $H = \sum_{\alpha=0}^7 (\alpha + 1) * (p(7 - \alpha, -1) - p(9 + \alpha, -1))$

where $V = \sum_{\beta=0}^7 (\beta + 1) * (p(16, 8 + \beta) - p(16, 6 - \beta))$

DC:

$$pred_L(x, y) = \begin{cases} (\sum_{\alpha=0}^{15} p(15-\alpha, -1) + \sum_{\beta=0}^{15} p(16, \beta) + 16) >> 5, p(0..15, -1) \wedge p(16, 0..15) \rightarrow available \\ (\sum_{\alpha=0}^{15} p(15-\alpha, -1) + 8) >> 4, p(0..15, -1) \rightarrow available, p(16, 0..15) \rightarrow unavailable \\ (\sum_{\beta=0}^{15} p(16, \beta) + 8) >> 4, p(0..15, -1) \rightarrow unavailable, p(16, 0..15) \rightarrow available \\ 128, p(0..15, -1) \rightarrow unavailable, p(16, 0..15) \rightarrow unavailable \end{cases}$$

4. horizontal: $pred_L(15-y, 15-x) = p(15-y, 16)$
 vertical: $pred_L(15-y, 15-x) = p(16, 15-x)$

then back to x, y coordinates by substitution $x = 15-y$ and $y = 15-x$

horizontal: $pred_L(x, y) = p(x, 16)$

vertical: $pred_L(x, y) = p(16, y)$

plane: $pred_L(x, y) = clip((a + b * (8 - y) + c * (8 - x) + 16) >> 5)$
 $a = 16 * (p(0, 16) + p(16, 0))$

where $b = (5 * H + 32) >> 6$,
 $c = (5 * V + 32) >> 6$

where $H = \sum_{\alpha=0}^7 (\alpha + 1) * (p(16, 7 - \alpha) - p(16, 9 + \alpha))$

where $V = \sum_{\beta=0}^7 (\beta + 1) * (p(7 - \beta, 16) - p(9 + \beta, 16))$

DC:

$$\text{pred}_L(x, y) = \begin{cases}
 \left(\sum_{\alpha=0}^{15} p(16, 15-\alpha) + \sum_{\beta=0}^{15} p(15-\beta, 16) + 16 \right) \gg 5, p(16, 0..15) \wedge p(0..15, 16) \rightarrow \text{available} \\
 \left(\sum_{\alpha=0}^{15} p(16, 15-\alpha) + 8 \right) \gg 4, p(16, 0..15) \rightarrow \text{available}, p(0..15, 16) \rightarrow \text{unavailable} \\
 \left(\sum_{\beta=0}^{15} p(15-\beta, 16) + 8 \right) \gg 4, p(16, 0..15) \rightarrow \text{unavailable}, p(0..15, 16) \rightarrow \text{available} \\
 128, p(16, 0..15) \rightarrow \text{unavailable}, p(0..15, 16) \rightarrow \text{unavailable}
 \end{cases}$$

Annex B

Proposal for slice header syntax

Here, in this Annex the modification of slice header syntax is proposed. Proposed slice header syntax is extended by including additional information about spiral scan such as slice length, spiral type etc.

slice_header() {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
spiral	2	u(1)
if(spiral) {		
slice_length	2	ue(v)
spiral_type	2	u(1)
if(spiral_type) {		
shape_width	2	ue(v)
shape_high	2	ue(v)
width_max	2	ue(v)
width_high	2	ue(v)
}		
shape_override_flag	2	u(1)
if(shape_override_flag) {		
shape_direction	2	u(1)
shape_length	2	ue(v)
}		
}		
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type == 5)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		

if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type == B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type == P slice_type == SP slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
ref_pic_list_reordering()	2	
if((weighted_pred_flag && (slice_type == P slice_type == SP)) (weighted_bipred_idc == 1 && slice_type == B))		
pred_weight_table()	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking()	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type == SP slice_type == SI) {		
if(slice_type == SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

The default value for **shape_direction** is 1.

The default value for **shape_length** is 1.

The default value for **spiral_type** is 0.

The default value for **shape_high** is 0.

The default value for **shape_width** is 0.

The default value for **shape_length** is 1.

The **shape_direction** 0 means that spiral will be wider in horizontal direction (if the value **shape_length** is different than 1, otherwise the spiral will have the shape of the square).

The values **shape_high**, **shape_width**, **width_max**, **high_max** may be used to continue the spiral scan, but using new slice. If the value **spiral_type** is set to 0 the new slice begins the new spiral scan.

