FINAL STUDY DISSERTATION

In the aim obtaining of
Master Degree

Field : Electrical Engineering
Option: Automatic

Presented by:

BEN SID Khaled
BENKABOUYA Abdessamed

Topic

Multiview Video Coding

Was Publicly Debated in: June 2012 in front of The Examining Committee

Composed From:

Mr. A. MERAOUUMIA MAA President Ouargla
Mrs. F. CHERIF MAA Examiner Ouargla
Mrs. S. ZEHANI MAB Examiner Ouargla
Dr. D. SAMAI MCB Supervisor Ouargla

Academic year: 2011/2012
Dedication

I dedicate this work

Some phrases expressing my gratitude and deep affection, in
this little space to them, the day came thanks to indicate this
I dedicate this humble work to those who have given life, hope and
love to those who took care for my well-being, those who were
present during the hardest times in my life and
keep encouraged morally and materially the course of my studies,
those who were patient and what I like

I respect,

My dear mother and my father!
At my buddy Abdessamed
To my brothers and sisters
At my uncles and aunts
for the whole family BEN SID

With all the promotion of Master Automatic (2012)

Khaled BEN SID
Dedication

I dedicate this work

Some phrases expressing my gratitude and deep affection, in this little space to them, the day came thanks to indicate this I dedicate this humble work to those who have given life, hope and love to those who took care for my well-being, those who were present during the hardest times in my life and keep encouraged morally and materially the course of my studies, those who were patient and what I like

I respect,

My dear mother and my father!
At my buddy Khaled
To my brothers and sisters
At my uncles and aunts
for the whole family BNEKABOUYA and ASSAL

With all the promotion of Master Automatic (2012)

Abdessamed BENKABOUYA
Acknowledgement

We would like to fully acknowledge and thank Allah for all the blessings and guidance we are granted. Very special thanks go to our supervisor Dr. SAMAI Djamel, who was more than generous with his expertise and precious time. Also, we give lots of credit to all the personnel and teachers’ staff of Kasdi Merbah University for all the facilities they have provided through all the process of making this project.
ملخص

الدراسة المنجزة في هذه المذكرة تدخل في الإطار العام لضغط الصور المتحركة أو الفيديو. وبصورة أدق ضغط الفيديو متعدد وجهات النظر، وهذا من أجل استغلاله فيما بعد للإرسال أو التخزين. وفي هذا الإطار قمنا بعملية الضغط باستخدام MVC.H264/AVC حيث أبدت النتائج مؤشرات إيجابية من حيث نسبة الضغط وكذا نوعية الصور (الفيديو) المتحصل عليها.

الكلمات المفتاحية: ضغط الفيديو، الفيديو متعدد وجهات النظر، MVC.H264/AVC

Abstract

The study carried out in this dissertation is inscribed in the general context of sequence compression or video, more specifically compression of multi-views video, and this in order to be exploited later for transmission or storage. In this context, we carried out using MVC.H264/AVC where indicators have shown positive results in terms of compression ratio as well as the quality of the images (video) obtained.

Keywords: Video compression, multi view video, MVC.H264/AVC.

Résumé

L’étude réalisée dans ce mémoire s’inscrit dans le contexte général de la compression de séquences d’images ou vidéos, plus précisément la compression vidéo multi-vue, dans le but de l’exploiter pour la transmission et le stockage. Dans ce cadre, nous avons utilisé le MVC.H264/AVC où les résultats obtenus sont plausibles de point de vue taux de bits ou de point de vue qualité visuelle des images reconstruites.

Mots clés : compression vidéo, vidéos multi-vues, MVC.H264/AVC
Table of Contents

Dedication .................................................................................. i
Acknowledgements ................................................................. iii
Abstract .................................................................................... iv
Table of Contents ................................................................. v
List of Figures .............................................................................. viii
List of Tables ................................................................................ x
List of Acronyms .......................................................................... xi
General Introduction .............................................................. 1

Chapter 1: Video Coding Concepts

1.1 Introduction ........................................................................... 3
1.2 Video Formats and Quality .................................................. 3
  1.2.1 YCbCr colour model ....................................................... 3
  1.2.2 Frames and fields .......................................................... 4
  1.2.3 YCbCr sampling formats ................................................. 4
  1.2.4 Video quality measure .................................................. 6
1.3 Video Coding Principle ....................................................... 7
1.4 Prediction Model ..................................................................... 9
  1.4.1 Intra prediction ............................................................. 9
  1.4.2 Temporal prediction ...................................................... 10
1.5 Predictive Image Coding ..................................................... 14
  1.5.1 Transform coding ......................................................... 14
  1.5.2 Quantization ................................................................. 15
1.6 Entropy Encoder ................................................................... 15
1.7 Conclusion .............................................................................. 15
Chapter 2: Overview of H.264 Advanced Video Compression

2.1 Introduction ............................................................... 16
2.2 How does an H.264 Codec Work? ................................. 16
2.3 Slices Processing ....................................................... 17
   2.3.1 I-slices processing ............................................. 17
   2.3.2 P-slices processing ............................................. 17
   2.3.3 B-slices processing ............................................. 17
2.4 The Encoder Structure of H.264/AVC ............................. 18
   2.4.1 Input video ..................................................... 18
   2.4.2 Prediction ........................................................ 19
   2.4.3 Transform, scaling and quantization .......................... 21
   2.4.4 Entropy encoder ............................................... 21
2.5 The Decoder Structure of H.264/AVC ............................. 22
   2.5.1 Bitstream decoding ............................................. 22
   2.5.2 Rescaling and inverse transform .............................. 22
   2.5.3 Reconstruction ................................................ 22
2.6 Conclusion .............................................................. 23

Chapter 3: Overview of Multiview Video Coding

3.1 Introduction ............................................................... 24
3.2 Applications of Multiview Video .................................... 24
   3.2.1 Free viewpoint video (FVV) .................................... 24
   3.2.2 Three dimensional television (3DTV) ......................... 24
   3.2.3 Immersive teleconferencing .................................. 24
3.3 Multiview Video System Architecture ............................. 25
3.4 Requirements for Multiview Video Coding ........................ 25
   3.4.1 Temporal random access ..................................... 26
   3.4.2 Scalability ....................................................... 26
   3.4.3 Backward compatibility ....................................... 26
   3.4.4 Quality consistency ............................................ 26
List of Figures

Chapter 1: Video Coding Concepts

Figure 1.1: Interlaced video sequence ................................................................. 4
Figure 1.2: YCbCr sampling formats ................................................................. 6
Figure 1.3: Encoder / Decoder ........................................................................... 7
Figure 1.4: Video encoder block diagram .......................................................... 8
Figure 1.5: Intra prediction: spatial extrapolation .............................................. 10
Figure 1.6: Block-based motion compensation ................................................... 11
Figure 1.7: Macroblock ....................................................................................... 11
Figure 1.8: Motion estimation ............................................................................. 12
Figure 1.9: Integer, half-pixel and quarter-pixel motion estimation ..................... 14

Chapter 2: Overview of H.264 Advanced Video Compression

Figure 2.1: Video codec: high level view ............................................................ 16
Figure 2.2: I, P and B-slices relationship ............................................................ 17
Figure 2.3: Typical H.264/AVC encoder ............................................................ 18
Figure 2.4: Intra prediction modes .................................................................... 20
Figure 2.5: Macroblock partitions, sub-macroblock partitions and partition scans
compensation ..................................................................................................... 20
Figure 2.6: Typical H.264/AVC decoder ............................................................ 22
Chapter 3: Overview of Multiview Video coding

Figure 3.1: Overall MVC system architecture .............................................................. 25
Figure 3.2: Matrix of pictures (MOP) for N = 4 image sequences, each comprising of
K = 4 temporally successive pictures ........................................................................ 27
Figure 3.3: Prediction modes for first-order neighbor images .................................... 29
Figure 3.4: Probability of choice of prediction mode when minimizing a Lagrangian cost
function in motion/disparity estimation for sequences Ballroom (left) and Race1 (right)… 29
Figure 3.5: Inter-view/temporal prediction structure based on H.264/AVC ................. 30

Chapter 4: Experimental Results

Figure 4.1: Inter-view/temporal prediction structure with GOP length of 12 .............. 34
Figure 4.2: Simulcast using hierarchical B pictures in temporal dimension only with GOP
length of 12 ....................................................................................................................... 35
Figure 4.3: Reordering of multiview input for compression with GOP length Of 12 .......... 36
Figure 4.4: Multiview video test data sequences, frames 0 (left), frames 10 (right), Exit
(left) and Vassar (right) ............................................................................................... 37
Figure 4.5: Ballroom sequence coding results (PSNR Y) ............................................ 39
Figure 4.6: Exit sequence coding results (PSNR Y) .................................................... 39
Figure 4.7: Vassar sequence coding results (PSNR Y) ................................................ 40
Figure 4.8: visual qualities of Exit frames in three cases, frames (right) and a close-up view
of them (left) .................................................................................................................. 41
Figure 4.9: Ballroom sequence coded using inter-view/temporal prediction of MVC (QP=29,
average bit-rate =638.37 Kbps, average PSNR Y=36.21 dB) ...................................... 42
Figure 4.10: Exit sequence coded using inter-view/temporal prediction of MVC (QP=29,
average bit-rate=352.79 Kbps, average PSNR Y=37.90 dB) ............................... 42
Figure 4.11: Vassar sequence coded using inter-view/temporal prediction of MVC (QP=29,
average bit-rate=393.14 Kbps, average PSNR Y=36.36 dB) .................................. 43
Chapter 4: Experimental Results

Table 4.1 Parameters of testing sequences for MVC ........................................ 32
Table 4.2 MVC coding parameters .......................................................... 33
Table 4.3 MVC, Simulcast and EIF comparison results for ballroom ............... 34
Table 4.4 MVC, Simulcast and EIF comparison results for exit .......................... 38
Table 4.5 MVC, Simulcast and EIF comparison results for vassar ...................... 39
List of Acronyms

1-D : One Dimensional
2-D: Two Dimensional
3-D: Three Dimensional
RGB: Red Green Blue
AVC : Advances video coding
Y: Luminance Component
U,V: Chrominance Components
JPEG: Joint Picture Expert Group
MSE: Mean Square Error
MAE: Mean Absolute Error
PSNR: Peak Signal-to-Noise Ratio
MV: motion vector
HD: high definition
I-DCT: Inverse Discrete Cosine Transform
F-DCT : Forward Discrete Cosine Transform
I-frame: Intra-frame
P-frame: Prediction frame
B-frame: Bi-predicted frame
SP: Switching P
SI : Switching I
MB: macroblock
GOP: group of Pecture
MPEG: Motion Picture Expert Group
HVC: Human visual system
CODEC : Encoder/Decoder
ME : Motion estimation
MC : Motion compensation
CPB : coded picture buffer
DPB : Decoded picture buffer
FMO : flexible macroblock ordering
**QP**: Quantization Parameter

**CAVLC**: Context-Adaptive Variable Length Coding

**CABAC**: Context-Adaptive Binary Arithmetic Coding

**MVC**: multiview video coding

**FVV**: free viewpoint video

**3DTV**: Three dimensional television

**MOP**: Matrix of pictures

**EIF**: Exploiting the Inter-view among all Frames

**JMVC**: Joint multiview coding

**JVT**: Joint Video Team
General Introduction

Multiview video refers to a set of temporally synchronized video streams coming from different cameras that capture the same real world scene. The growing need for the adoption and compression of Multiview video is driven by its application such as 3DTV and FVV. Using multiple video cameras positioned at different angles and locations, various viewpoints of a scene can be captured simultaneously to generate a free-view point video or to create realistic 3-D impressions.

Both applications are intended to add a dynamic and immersive dimension to future multimedia experience for conventional users in the relevant areas, such as broadcasting, teleconference, surveillance, interactive video. It allows users to change viewpoint freely on a 2-D or 3-D display when multiple views are available. To enhance the perceived realism of the 3-D output, camera density around a scene needs to be increased, and thus a vast amount of multiview video data needs to be stored or transmitted for a 3-D video. Efficient multiview video coding (MVC) is then required to compress this vast amount of information for practical storage and transmission. MVC was designed as an amendment of the H.264/AVC and uses improved coding tools which have made H.264/AVC so superior over older standards. It is based on conventional block-based motion-compensated video coding, with several new features which significantly improve its rate-distortion performance.

The H.264/AVC is the state-of-the-art video coding standard, developed jointly by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Pictures Experts Group (MPEG). H.264/AVC is developed to provide efficient compression and high reliability in video transmission [1]. The benefits of high compression efficiency and error robustness of H.264/AVC standard have already become widely accepted and used for video content storage, streaming and transmission of real-time multimedia. Therefore, most recent development of VCEG and MPEG focuses on the multiview video [2].

As the video data originate from the same scene, the inherent similarities of the multiview imagery are exploited for efficient compression. These similarities can be classified into two types. First, inter-view similarity is observed between adjacent camera views. Second, temporal similarity is noticed between temporally successive images of each video.
So, in this work, we will investigate the performance of MVC Scheme that exploiting these similarities in the same time by comparing it with Simulcast Scheme (exploit only temporal similarity) and EIF (exploit only inter-view similarity among all frames).

The rest of this work is organized as follows:

Chapter 1 introduces the preliminary video coding concepts that one needs to know in order to understand the rest of the work that explains the concepts of digital video and covers source formats and visual quality measures and introduces video compression and the functions found in a typical video codec.

Chapter 2 gives a high-level overview of H.264/AVC at a relatively non-technical level.

Chapter 3 provides an overview of multiview video coding (MVC) and describes its applications, requirements.

Chapter 4 reviews the experimental results among the different Schemes.

At last, we will finish by General Conclusion that concluding remarks and the problems that remain to be solved in future research.
Chapter 1

Video Coding Concepts

1.1 Introduction

Why can video be compressed? The reason is that video contains much spatial and temporal redundancy. In a single frame, nearby pixels are often correlated with each other. This is called spatial redundancy, or the intra-frame correlation. Another one is temporal redundancy, which means adjacent frames are highly correlated, or called the inter-frame correlation. Therefore, the goal of the video compressed is to efficiently reduce spatial and temporal redundancy to achieve video compression. In this chapter we examine the basic concepts of video coding.

1.2 Video Formats and Quality

1.2.1 YCbCr colour model

The human visual system (HVS) is less sensitive to colour than to luma (luminance). The first step in converting the RGB signal into a composite signal is its conversion to a YCbCr signal, where the Y signal corresponds to luma, and chroma (chrominance) Cb and Cr correspond to two colour-difference signals. The reason for this conversion is that although the luma signal requires the full bandwidth of the original RGB signals, the bandwidth of the Cb and Cr signals can be reduced without any visual picture degradation, once the signal is converted back to RGB. YCbCr can be computed for the following widely-used conversion equations [1]:

\[
\begin{align*}
Y &= 0.299R + 0.587G + 0.114B \\
Cb &= 0.564(B - Y) \\
Cr &= 0.713(R - Y)
\end{align*}
\] (1.1)

\[
\begin{align*}
R &= Y + 1.402Cr \\
G &= Y - 0.344Cb - 0.714Cr \\
B &= Y + 1.772Cb
\end{align*}
\] (1.2)
In applications, the YCbCr transform is used in JPEG image compression and MPEG video compression.

### 1.2.2 Frames and fields

A video signal may be sampled as a series of complete frames (progressive sampling) or as a sequence of interlaced fields (interlaced sampling). In an interlace video sequence, half of the data in a frame (one field) is sampled at each temporal sampling interval. A field consists of either the odd-numbered or even-numbered lines within a complete video frame and an interlaced video sequence (Figure 1.1) contains a series of fields each representing half of the information in a complete video frame. The advantage of this sampling method is that it is possible to send twice as many fields per second as the number of frames in an equivalent progressive sequence with the same data rate, giving the appearance of smoother motion [1].

![Interlaced video sequence](image)

**Figure 1.1** Interlaced video sequence

### 1.2.3 YCbCr sampling formats

Figure 1.2 shows three sampling patterns for luma Y, and chroma, Cb and Cr.

In the 4:4:4 sampling scheme, all three components have the same spatial resolution and represent the maximum spatial and colour accuracy. The term 4:4:4 is used to indicate that for every four Y pixels, there are four Cb pixels and four Cr pixels.
In the 4:2:2 sampling scheme, both Cb and Cr components are low-pass filtered in the horizontal direction and sampled by a factor of 2. Thus, for every four Y pixels, there are two Cb pixels and two Cr pixels.

In the 4:2:0 sampling scheme, Cb and Cr components are low-pass filtered in horizontal and vertical directions and sampled by a factor of 2 in both dimensions. Thus, for every four Y pixels, there is one Cb pixel and one Cr pixel [3].
1.2.4 Video quality measure

In order to evaluate the performance of video compression coding, it is necessary to define a measure to compare the original video and the video compressed. Most video
compression systems are designed to minimize the mean square error (MSE) between two video sequences \( \Psi_1 \) and \( \Psi_2 \), which is defined as:

\[
MSE = \frac{1}{N} \sum_{t} \sum_{x,y} [\Psi_1(x,y,t) - \Psi_2(x,y,t)]^2
\]  

(1.3)

Where N is the total number of pixels in either sequence.

Instead of the MSE, the peak-signal-to-noise ratio (PSNR) in decibel (dB) is more often used as a quality measure in video coding, which is defined as:

\[
PSNR = 10 \log_{10} \left( \frac{(2^n - 1)^2}{MSE} \right)
\]  

(1.4)

where \((2^n - 1)^2\) is the square of the highest-possible signal value in the image, and n is the number of bits per image sample [4].

### 1.3 Video Coding Principle

Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder). The encoder converts the source data into a compressed form occupying a reduced number of bits, prior to transmission or storage, and the decoder converts the compressed form back into a representation of the original video data. The encoder/decoder pair is often described as a CODEC (enCOder/DECoder) (Figure 1.3).

![Figure 1.3 Encoder / Decoder](image)

A video encoder (Figure 1.4) consists of three main functional units: a prediction model, a spatial model and an entropy encoder.
The input to the prediction model is an uncompressed ‘raw’ video sequence. The prediction model attempts to reduce redundancy by exploiting the similarities between neighbouring video frames and/or neighbouring image samples, typically by constructing a prediction of the current video frame or block of video data. It is created by spatial extrapolation from neighbouring image samples, intra prediction, or by compensating for differences between the frames, inter or motion compensated prediction. The output of the prediction model is a residual frame, created by subtracting the prediction from the actual current frame, and a set of model parameters indicating the intra/inter prediction type.

The residual frame forms the input to the spatial model which makes use of similarities between local samples in the residual frame to reduce spatial redundancy.

The parameters of the prediction model, i.e. intra prediction mode(s) or inter prediction mode(s) and motion vectors, and the spatial model, i.e. coefficients, are compressed by the entropy encoder. This removes statistical redundancy in the data.

The entropy encoder produces a compressed bit stream or file that may be transmitted and/or stored. A compressed sequence consists of coded prediction parameters, coded residual coefficients and header information.

The video decoder reconstructs a video frame from the compressed bit stream. The coefficients and prediction parameters are decoded by an entropy decoder after which the spatial model is decoded to reconstruct a version of the residual frame. The decoder uses the prediction parameters, together with previously decoded image pixels, to create a prediction of the current frame and the frame itself is reconstructed by adding the residual frame to this prediction [1].

Multiview Video Coding 8
1.4 Prediction Model

The data to be processed are a set of image samples in the current frame or field. The goal of the prediction model is to reduce redundancy by forming a prediction of the data and subtracting this prediction from the current data. The prediction may be formed from previously coded frames (a temporal prediction or inter prediction) or from previously coded image samples in the same frame (a spatial prediction or intra prediction). The output of this process is a set of residual or difference samples and the more accurate the prediction process, the less energy is contained in the residual. The residual is encoded and sent to the decoder which re-creates the same prediction so that it can add the decoded residual and reconstruct the current frame. In order that the decoder can create an identical prediction, it is essential that the encoder forms the prediction using only data available to the decoder.

1.4.1 Intra prediction

The prediction for the current block of image samples is created from previously coded samples in the same frame. Many different approaches to intra prediction have been proposed. H.264/AVC uses spatial extrapolation to create an intra prediction for a block or macroblock. Figure 1.5 shows the general concept.

One or more prediction(s) are formed by extrapolating samples from the top and/or left sides of the current block. In general, the nearest samples are most likely to be highly correlated with the samples in the current block and so only the pixels along the top and/or left edges are used to create the prediction block. Once the prediction has been generated, it is subtracted from the current block to form a residual in a similar way to inter prediction. The residual is transformed and encoded, together with an indication of how the prediction was generated [1].
1.4.2 Temporal prediction

The predicted frame is created from one or more past or future frames known as reference frames. The accuracy of the prediction can usually be improved by compensating for motion between the reference frame(s) and the current frame.

1.4.2.1 Block-based motion estimation and compensation

To reduce temporal redundancy in video sequence, we use the motion-compensated prediction method. A practical and widely-used method of motion compensation is to compensate for movement of blocks of the current frame, as show in Figure 1.6. The following procedure is carried out for each M×N block in the current frame:

- Search an area in the reference frame (past or future frame, previously coded and transmitted) to find the best matching M×N block. This is carried out by comparing the M×N block in the current frame with all of the possible M×N regions in the search area. This process of finding the best match is known as motion estimation (ME).
- The best matching block found by the above step becomes the predictor for the current M×N block and is subtracted from the current block to form a residual M×N block, motion compensation (MC).
1.4.2.2 Motion compensated prediction of a macroblock

The macroblock, corresponding to a 16x16 pixel region of a frame, is the basic unit for motion compensated prediction in a number of important visual coding standards including MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263 and H.264/AVC. For source video material in the popular 4:2:0 format, a macroblock is organised as shown in Figure 1.7. A 16x16 pixel region of the source frame is represented by 256 luminance samples arranged in four 8x8 sample blocks, 64 red chrominance samples in one 8x8 block and 64 blue chrominance samples in one 8x8 block, giving a total of six 8x8 blocks. An H.264/AVC codec processes each video frame in units of a macroblock.

![Figure 1.6](image)

**Figure 1.6** Block-based motion compensation

**Figure 1.7** Macroblock (4:2:0)

a) Motion estimation

At first, we divide current frame into non-overlapping macroblocks. For each macroblock, find the best matching block in a reference frame. That is to say, motion estimation of a macroblock involves finding 16x16 block in the search area in a reference frame that closely matches the current macroblock (Figure 1.8).
The reference frame is a previously-encoded frame from the sequence and may be before or after the current frame in display order. The search area in the reference frame is centered on the current macroblock position.

![Figure 1.8 Motion estimation](image)

b) Motion compensation

When the best matching block is found in the reference frame by motion estimation, we subtract the best matching block from the current macroblock to produce a residual macroblock.

Within the encoder, the residual is encoded and decoded and added to the matching region to form a reconstructed macroblock which is stored as a reference for further motion-compensated prediction.

c) Energy Measure: metrics for determining the best matching block

A popular matching criterion is the energy in the residual formed by subtracting the candidate block in the search area from the current macroblock, so the candidate block that minimizes the residual energy is chosen as the best match. Here are some metrics for determining the best match:
• Mean squared error:

\[ MSE = \frac{1}{N} \sum_{x,y \in \text{Block}} \left[ f(x, y, t_{\text{current}}) - f(x, y, t_{\text{reference}}) \right]^2 \]  \hspace{1cm} (1.5)

• Mean absolute Error:

\[ MAE = \frac{1}{N} \sum_{x,y \in \text{Block}} \left| f(x, y, t_{\text{current}}) - f(x, y, t_{\text{reference}}) \right| \]  \hspace{1cm} (1.6)

1.4.2.3 Motion compensation block size

Different motion compensation block sizes produce different motion compensation results. Experimentally, smaller motion compensation block sizes can produce better motion compensation results. However, a smaller block size leads to increased complexity and an increase in the number of motion vectors that need to be transmitted.

An effective compromise is to adapt the block size to the image characteristics, for example choosing a larger block in flat, homogeneous regions of a frame and choosing a small block size around areas of high detail and complex motion.

1.4.2.4 Sub-pixel motion compensation

Sub-pixel motion estimation and compensation involves searching sub-sample interpolated positions as well as integer-sample positions, choosing the position that gives the best match (i.e. minimizes the residual energy) and using the integer- or sub sample values at this position for motion compensated prediction which is shown in Figure 1.9.
1.5 Predictive Image Coding

1.5.1 Transform coding

Transform coding has been widely used to remove redundancy between data samples. In transform coding, a set of data samples is first linearly transformed into a set of transform coefficients. These coefficients are then quantized and coded.

The most commonly used transform for video coding is the DCT [5]. It operates on $X$, a block of $N \times N$ samples, typically image samples or residual values after prediction, to create $Y$, an $N \times N$ block of coefficients. The action of the DCT and its inverse, the IDCT can be described in terms of a transform matrix $A$. The forward DCT (FDCT) of an $N \times N$ sample block is given by [1]:

$$Y = AXA^T$$ (1.7)

and the inverse DCT (IDCT) by:

$$X = A^T YA$$ (1.8)

Where $X$ is a matrix of samples, $Y$ is a matrix of coefficients and $A$ is an $N \times N$ transform matrix. The elements of $A$ are:

$$A_{ij} = C_i \cos \left(\frac{(2j + 1)i\pi}{2N}\right)$$

where

$$C_i = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } i = 0, \\ \frac{2}{\sqrt{N}} & \text{for } i > 0 \end{cases}$$

(1.9)
1.5.2 Quantization

A quantizer maps a signal with a range of values $X'$ to a quantized signal with a reduced range of values $Y$. It should be possible to represent the quantized signal with fewer bits than the original since the range of possible values is smaller.

1.6 Entropy Encoder

The entropy encoder converts a series of symbols representing elements of the video sequence into a bitstream suitable for transmission or storage. The bitstream is composed by:

- Quantized transform coefficients;
- Information to enable the decoder to re-create the prediction;
- Information about the structure of the compressed data and the compression tools used during encoding;
- Information about the complete video sequence.

1.7 Conclusion

In this chapter, we have described the video coding tools, motion compensated prediction, transform coding, quantization and entropy coding. This coding model is at the heart of the H.264/AVC standard. The next chapter introduces the main features of H.264/AVC and the standard in slightly more detail.
Chapter 2

Overview of H.264 Advanced Video Compression

2.1 Introduction

H.264 Advanced Video Coding is an industry standard for video coding. The main goals of this standard are a simple and straightforward video coding design, with enhanced compression performance. H.264/AVC has achieved a significant improvement in the rate-distortion efficiency providing, typically, a factor of two in bit-rate savings when compared with existing standards such as MPEG-2 Video. In this chapter we provide an overview of the technical features of H.264/AVC.

2.2 How does an H.264 Codec Work?

A frame or field to be coded, e.g. Frame A in Figure 2.1, is processed by an H.264 compatible video encoder. As well as coding and sending the frame as part of the coded bitstream or coded file, the encoder reconstructs the frame, i.e. creates a copy of the decoded frame A’ that will eventually be produced by the decoder. This reconstructed copy may be stored in a coded picture buffer, CPB, and used during the encoding of further frames. The decoder receives the coded bitstream and decodes frame A’ for display or further processing. At the same time, the decoder may store a copy of frame A’ in a decoded picture buffer, DPB, to be used during the decoding of further frames [1].

![Video codec: high level view](image-url)

**Figure 2.1** Video codec: high level view
2.3 Slices Processing

Depending upon the subset of coding tools used, the coding type of a slice can be I (Intra), P (Predicted), B (Bi-predicted), SP (Switching P) or SI (Switching I). A picture may contain different slice types, and pictures come in two basic types: reference and non-reference. Reference pictures can be used as references for inter frame prediction during the decoding of later pictures (in bitstream order) while non-reference pictures cannot. The main slice types are I, P and B. SP and SI slices were designed in order to introduce additional functionalities such as random access, fast forward, and stream splicing. The relationship among I, P and B-slices is shown in Figure 2.2, in which each arrow indicates the prediction direction.

![Figure 2.2 I, P and B-slices relationship](image)

2.3.1 I-slices processing

In I-slices pixel values are first spatially predicted from their neighboring pixel values. After the spatial prediction, the residual information is transformed using a 4×4 transform or an 8×8 transform and then quantized, scanned and entropy coded.

2.3.2 P-slices processing

In P-slices temporal (rather than spatial) prediction is used by estimating motion between previously encoded pictures.

2.3.3 B-slices processing

In B-slices two motion vectors, representing two estimates of the motion per macroblock partition or sub-macroblock partition, are allowed for temporal prediction.
2.4 The Encoder Structure of H.264/AVC

The structure of a typical H.264 encoder is shown in Figure 2.3 in slightly more detail.

![Diagram of H.264/AVC encoder]

Figure 2.3 Typical H.264/AVC encoder

2.4.1 Input video

H.264/AVC uses 4:2:0 sampling format [6], and supports all type of the common Intermediate Format (CIF), like QCIF (176 X 144) and 4CIF (704 X 576). Each picture of a video, which can either be a frame or a field, is partitioned into fixed-size macroblocks that cover a rectangular picture area of 16×16 samples of the luma component and 8×8 samples of each of the two chroma components.
2.4.2 Prediction

The encoder forms a prediction of the current macroblock based on previously-coded data, either from the current frame using intra prediction or from other frames that have already been coded and transmitted using inter prediction. The encoder subtracts the prediction from the current macroblock to form a residual (Figure 2.3).

The prediction methods supported by H.264/AVC are more flexible than those in previous standards, enabling accurate predictions and hence efficient video compression [1].

2.4.2.1 H264 /AVC intra frame prediction

Each macroblock can be transmitted in one of several coding types depending on the slice-coding type. In all slice-coding types, two classes of intra coding types are supported, which are denoted as INTRA-4×4 and INTRA-16×16 in the following. In contrast to previous video coding standards where prediction is conducted in the transform domain, prediction in H.264/AVC is always conducted in the spatial domain by referring to neighboring samples of already coded blocks.

When using the INTRA-4×4 mode, each of the 4×4 luma blocks can be predicted using either the dc mode or one of the eight coding directions listed in Figure 2.4 (c) and illustrated in Figure 2.4 (a). For the purpose of illustration, Figure 2.4 (b) shows a 4×4 block of pixels a, b, c, .., p, belonging to a macroblock to be coded [7]. Pixels A, B, C, .., H and I, J, K, L, M are already decoded neighboring pixels used in computation of prediction of pixels of current 4×4 block. Directional predictions use a linear weighted average of pixels of A through H and I through M, depending on the specific direction of the prediction.

When utilizing the Intra-16×16 mode, four prediction modes are supported. Prediction mode 0 (vertical prediction), mode 1 (horizontal prediction), mode 2 (DC prediction), and mode 3 (plane prediction) are specified similar to the modes in Intra-4×4 prediction except the number of neighboring pixels. The 8×8 chroma mode also uses a prediction technique which is similar to the one for Intra-16×16.
H.264/AVC standard is more flexible in the selection of motion compensation (MC) block sizes and shapes than any previous standard, with a minimum luma MC block size as small as 4x4. Figure 2.5 illustrates the macroblock partitioning for MC prediction [8]. Prediction values at quarter-sample positions are generated by averaging samples at integer and half-sample positions. Since it is 4:2:0 video format, the displacements used for chroma have one-eight sample position accuracy. The motion vector components are differentially coded using either median or directional prediction from neighboring blocks.

**Figure 2.4** Intra prediction modes

### 2.4.2.2 Inter frame prediction

H.264/AVC standard is more flexible in the selection of motion compensation (MC) block sizes and shapes than any previous standard, with a minimum luma MC block size as small as 4x4. Figure 2.5 illustrates the macroblock partitioning for MC prediction [8]. Prediction values at quarter-sample positions are generated by averaging samples at integer and half-sample positions. Since it is 4:2:0 video format, the displacements used for chroma have one-eight sample position accuracy. The motion vector components are differentially coded using either median or directional prediction from neighboring blocks.

**Figure 2.5** Macroblock partitions, sub-macroblock partitions and partition scans
2.4.3 Transform, scaling and quantization

In H.264/AVC, the transformation is applied to 4×4 blocks, and instead of a 4×4 discrete cosine transform (DCT), a separable integer transform with basically the same properties as a 4×4 DCT is used [9].

For the quantization of transform coefficients, H.264/AVC uses scalar quantization. One of 52 quantizers is selected for each macroblock by the Quantization Parameter (QP). The quantizers are arranged so that there is an increase of approximately 12.5% in the quantization step size when incrementing the QP by one. The quantized transform coefficients of a block are generally scanned in a zigzag fashion and transmitted using entropy coding methods [9].

2.4.4 Entropy encoder

In H.264/AVC, two methods of entropy coding are supported. The first one is Context-Adaptive Variable Length Coding (CAVLC) and the other one is Context-Adaptive Binary Arithmetic Coding (CABAC).

The first (CAVLC) is based on a variable-length approach. The conversion tables that are used to compress the data are computed based on the content of the video. The second approach (CABAC) works by modifying the coding parameters to improve efficiency.

These new techniques vastly improve the compression available through entropy coding [10].

Each of these encoding methods produces an efficient, compact binary representation of the information. The encoded bitstream can then be stored and/or transmitted.
2.5 The Decoder Structure of H.264/AVC

The structure of a typical H.264 decoder is shown in Figure 2.6 in slightly more detail.

![Figure 2.6 Typical H.264/AVC decoder](image)

2.5.1 Bitstream decoding

A video decoder receives the compressed H.264/AVC bitstream, decodes each of the syntax elements and extracts the information described above, i.e. quantized transform coefficients, prediction information, etc. This information is then used to reverse the coding process and recreate a sequence of video images.

2.5.2 Rescaling and inverse transform

The quantized transform coefficients are re-scaled. Each coefficient is multiplied by an integer value to restore its original scale.

2.5.3 Reconstruction

For each macroblock, the decoder forms an identical prediction to the one created by the encoder using inter prediction from previously-decoded frames or intra prediction from previously-decoded samples in the current frame. The decoder adds the prediction to the decoded residual to reconstruct a decoded macroblock which can then be displayed as part of a video frame.
2.6 Conclusion

In this chapter, we have described the mechanisms for coding video that are optimised for compression efficiency. The next chapter we will provide an overview of multiview video that encoded by H.264/AVC.
Chapter 3

Overview of Multiview Video Coding

3.1 Introduction

The multiview video includes multiple video sequences captured by several cameras at the same time, but different locations. Because of the increased number of cameras, the multiview video contains a large amount of data. Therefore, we need to compress the multiview sequence efficiently without sacrificing its visual quality significantly. This chapter provides an overview of multiview video coding (MVC) and describes its applications, requirements, and its current techniques.

3.2 Applications of Multiview Video

The growing need for the adoption and compression of multiview video is driven by three recent technological advancements: free viewpoint video (FVV), three dimensional television (3DTV), and Immersive teleconferencing [11].

3.2.1 Free viewpoint video (FVV)

In this kind of application, the viewpoint and view direction can be interactively changed to ones, which may be different from any of the input ones.

3.2.2 Three dimensional television (3DTV)

In this kind of applications, multiple cameras are used to capture the light field of the scene, so that multiple viewers can see different stereoscopic views consistent with their relative locations.

3.2.3 Immersive teleconferencing

In this kind of application, participants at different geographical sites meet virtually and see one another in either free viewpoint or 3DTV style, with the interaction between participants.
3.3 Multiview Video System Architecture

Figure 3.1 shows the overall MVC system structure. In addition to temporal redundancies between adjacent frames of individual camera view, multiple camera signals also contain a large amount of statistical dependencies. The encoder receives N temporally synchronized video streams and generates one output bitstream. The decoder receives the bitstream, decodes and outputs N video signals.

![Overall MVC system architecture](image)

**Figure 3.1** Overall MVC system architecture

3.4 Requirements for Multiview Video Coding

The central requirement for any video coding standard is high compression efficiency. Compression efficiency measures the trade-off between costs (in terms of bit rate) and benefits (in term of video quality), i.e. the quality at a certain bit rate or the bit rate at a certain quality. However, compression efficiency is not the only thing to be required from a video coding standard. Some requirements of a video coding standard may even be contradictory such as compression efficiency and low delay in some cases. Then a good trade-off has to be found. General requirements for video coding such as minimum resource consumption (memory, processing power), low delay, error robustness, or support of different pixel and
colour resolutions, are often applicable to all video coding standards. Some requirements are specific to MVC as highlighted in the following [12]:

3.4.1 Temporal random access

Temporal random access is a requirement for any video codec. For MVC also view random access becomes important. Both together ensure that any image can be accessed and for instance displayed. Random access can be provided by insertion of I pictures that don’t use any prediction from other pictures.

3.4.2 Scalability

Scalability is a desirable feature for some video coding standards. This means that a decoder can access a portion of a bitstream in order to generate a low-quality video output. This may be a reduced temporal or spatial resolution, or a reduced video quality. For MVC, additionally view scalability is required. In this case a portion of the bitstream can be accessed in order to output a limited number out of the N views.

3.4.3 Backward compatibility

Backward compatibility is required for MVC. This means that one bitstream corresponding to one view that is extracted from the MVC bitstream shall be conforming to H.264/AVC.

3.4.4 Quality consistency

Quality consistency among views is also addressed. It should be possible to adjust the encoding for instance to provide approximately constant quality over all views.

3.4.5 Parallel processing

Parallel processing of different views or segments of the multi-view video is required to facilitate efficient encoder and decoder implementations.

3.4.6 Camera parameters

Camera parameters (extrinsic and intrinsic) should be transmitted with the bitstream in order to support intermediate view interpolation at the decoder.
3.5 Predictive Coding for Multiview Sequences

Two inherent similarities exist between video frames captured. The first one is inter-view similarities between adjacent camera views, and the other one is temporal similarity between successive temporal images of each video. Using these two properties, we can group the captured multiview images into a matrix of pictures (MOP) of size $N \times K$ [13]. Each row of MOP consists of $K$ temporally successive pictures of an individual view, and each column includes $N$ spatially neighboring views captured at the same time instant. In the temporal direction or one row of the MOP, the same object can appear in successive images, and possibly appear at different pixel locations when it is in motion. In this case, motion compensation techniques are utilized to exploit the temporal redundancies. Similarly, in the column of MOP where spatial neighboring views are captured, the same object appears in neighboring views but at different locations. Instead of motions, the object between each frame is subject to parallax. Therefore, disparity compensation can be used to exploit these inter-view similarities. Figure 3.2 illustrates a MOP of $N=4$ and $K=4$, with index $X_{n,k}$, and $n$ denotes the $n^{th}$ camera view, and $k$ denotes the $k^{th}$ temporal view.

\begin{figure}[h]
    \centering
    \begin{tabular}{cccc}
    View 0 & X1,1 & X1,2 & X1,3 & X1,4 \\
    View 1 & X2,1 & X2,2 & X2,3 & X2,4 \\
    View 2 & X3,1 & X3,2 & X3,3 & X3,4 \\
    View 3 & X4,1 & X4,2 & X4,3 & X4,4 \\
    \end{tabular}
    \caption{Matrix of pictures (MOP) for $N = 4$ image sequences, each comprising of $K = 4$ temporally successive pictures [14].}
\end{figure}
3.5.1 Motion compensation

Block matching is the most well-known technique of motion compensation, in which a motion vector establishes the displacement relationship between two similar chosen blocks from two images. To improve prediction and reduce any possible redundancies, multi frame and superposition techniques are adopted to create multiple motion correspondences resulting from multiple references frames in both forward and backward temporal directions.

3.5.2 Disparity compensation

Consider spatially neighboring views captured at the same time instant, i.e., images in one column of the MOP. Objects in each image are subject to parallax and appear at different pixel locations. To exploit these inter-view similarities, disparity compensation techniques are used.

The simplest approach to disparity compensation is block matching techniques similar to those used for motion compensation. These techniques offer the advantage of not requiring knowledge of the geometry of the underlying 3D objects. However, if the cameras are sparsely distributed, the block-based translatory disparity model fails to compensate accurately.

More advanced approaches to disparity compensation are depth-image-based rendering algorithms [16]. They synthesize an image as seen from a given view-point by using the reference texture and depth image as input data. These techniques offer the advantage that the given view-point image is compensated more accurately even when the cameras are sparsely distributed. However, these techniques rely on accurate depth images, which are difficult to estimate.

3.6 Overview of the H.264 MVC standard

A straightforward solution for the multiview video coding issue would be to encode all the video signals independently using a state-of-the-art video codec such as H.264/AVC.
3.6.1 Temporal and Inter-view correlation

The key for efficient MVC is inter-view prediction in addition to temporal prediction. For the case of linear camera settings, the inter-view/temporal first order neighbors are shown in Figure 3.3.

![Figure 3.3 Prediction modes for first-order neighbor images](image)

With the exception of the leftmost and rightmost cameras, each picture of the multiview sequence has 8 inter-view/temporal neighbors. In order to determine by which percentage a rate-distortion optimized encoder such as H.264/AVC would choose either one of the available modes, specific analyses has been computed. Some results in [12] are shown in the bar graphs of Figure 3.4 for the two data sets “Ballroom” and “Race1” indicating the likelihood of prediction mode selection. Here the prediction mode was chosen with the lowest Lagrangian cost value for Lagrangian motion estimation as described in [16].

![Figure 3.4 Probability of choice of prediction mode when minimizing a Lagrangian cost function in motion/disparity estimation for sequences Ballroom (left) and Race1 (right)](image)
The first conclusion drawn from the analysis over a larger set of multiview sequences was that temporal prediction is always the most efficient prediction mode, as it is highlighted in Figure 3.4. However, there are significant differences between the test data sets, regarding the relationship between temporal and inter-view prediction. Results show that there is a connection to the spatiotemporal density of the multi-view data. Specifically, inter-view prediction is used more often for low frame rates and very close cameras, which is intuitively understandable. Further, there is a connection to the scene complexity: inter-view prediction is used more often for scenes with rapidly moving objects and less for scenes with large areas being covered by static background. As a conclusion, the inter-view prediction could significantly improve the coding performance whenever the specific sequence being encoded presents favourable features [12].

### 3.6.2 MVC Prediction Structures

In order to efficiently exploit all the statistical dependencies within the multiview video datasets, several dedicated inter-view/temporal prediction structures have been developed. Figure 3.5 shows a structure developed by Fraunhofer HHI for the case of a 1D camera arrangement (linear or arc), which was proposed to MPEG as response to the “Call for Proposals” in [17].

![Inter-view/temporal prediction structure based on H.264/AVC](image)
This scheme uses the prediction structure of hierarchical B pictures for each view in temporal direction. Hierarchical B pictures provide significantly improved compression performance when the quantization parameters for the various pictures are assigned appropriately. Additionally, inter-view prediction is applied to every 2nd view, i.e. View 1, View 3 and View 5. For an even number of views, the last view (View 7) is coded starting with a P picture and followed by hierarchical B pictures, which are also inter-view predicted from the previous views. Thus, the coding scheme can be applied to any multi-view setting with more than 2 views. In order to allow random access, I pictures are inserted (View 0/T0, View 0/T8, etc.). The inter-view/temporal prediction structure in Figure 3.5 applies hierarchical B pictures in temporal and inter-view direction. After that, the multi-view video sequences are combined into one single uncompressed video stream using a specific scan. This uncompressed video stream can be fed into standard encoder software, and the inter-view/temporal prediction structure discussed can be realized by appropriate setting of the hierarchical B picture prediction scheme [12].

This is a pure encoder optimization, thus the resulting bitstream is standard conforming and can be decoded by any standard H.264/AVC decoder. The example above is for a Group of Pictures (GOP) length of 8, meaning that every 8th picture of the base view (View 0) is an I picture to allow random access. However, the syntax of hierarchical B pictures is very flexible and multi-view GOPs of any length can be specified. Moreover, other types of camera arrangements can be handled efficiently as well [12].

3.7 Conclusion

In this chapter we have introduced the technical features of the multiview video coding and described its application, requirements.
Chapter 4

Experimental Results

4.1 Introduction

In this chapter we compare the performance of three schemes that will encode the sequences. The first one is MVC scheme using the reference software JMVC 3.01 provided by MPEG/JVT [19] while the second one so-called Simulcast, i.e. encoding multiple views independently with H.264/AVC that we can use JMVC 3.01 as well. The third one is to encoding all the frames by exploiting the inter-view similarities among them. Experimental quality comparison of coding efficiency is given. The peak signal-to-noise ration (PSNR) is used as objective quality metrics.

4.2 Test Setting

We adopted three MVC test sequences as specified in Table 4.1 to evaluate the rate-distortion performance of three schemes.

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Resolution</th>
<th>Number of views</th>
<th>Horizontal camera spacing</th>
<th>Number of frames encoded</th>
<th>YUV format</th>
<th>Frame rate (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballroom</td>
<td>640×480</td>
<td>8</td>
<td>20 cm</td>
<td>13</td>
<td>4:2:0</td>
<td>25</td>
</tr>
<tr>
<td>Exit</td>
<td>640×480</td>
<td>8</td>
<td>20 cm</td>
<td>13</td>
<td>4:2:0</td>
<td>25</td>
</tr>
<tr>
<td>Vassar</td>
<td>640×480</td>
<td>8</td>
<td>20 cm</td>
<td>13</td>
<td>4:2:0</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 4.2 summarizes the coding parameters of the reference software JMVC 3.0 used to generate the experimental results for three schemes and each coding parameters described in the following:
Table 4.2 MVC coding parameters

<table>
<thead>
<tr>
<th>Coding Parameters</th>
<th>MVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SymbolMode</td>
<td>CABAC</td>
</tr>
<tr>
<td>GOPSize</td>
<td>12</td>
</tr>
<tr>
<td>BasisQP</td>
<td>29, 31, 34, 37, 40</td>
</tr>
<tr>
<td>SearchRange</td>
<td>±64</td>
</tr>
<tr>
<td>NumberReferenceFrames</td>
<td>2</td>
</tr>
<tr>
<td>ViewOrder</td>
<td>0-2-1-4-3-6-5-7</td>
</tr>
</tbody>
</table>

- **SymbolMode**: Specifies the entropy coding mode. The video sequence is encoded using context-adaptive binary arithmetic coding (CABAC). CABAC usually provides an increased coding efficiency.

- **GOPSize**: Specifies the GOP size that shall be used for encoding a video sequence. A GOP (group of pictures) consists of an anchor picture and several hierarchically coded B pictures that are located between the anchor pictures. The parameter GOP Size must be equal to a power of 2 or 12 and 15 for MVC. The maximum allowed value is 64.

- **BasisQP**: Specifies the basic quantization parameter. This parameter shall be used to control the bit-rate of a bitstream and therefore we change this 29 to 40 to evaluate the PSNR.

- **SearchRange**: Specifies the maximum search range for the motion search.

- **NumberReferenceFrames**: Specifies the maximum number of active entries for the reference pictures lists 0 and 1. The actual number of active entries that are used for encoding a specific frame, is additionally dependent on the location of a frame inside the group of pictures. For MVC this value should be set to 2 for now to reflect the prediction structure of JMVC description.

- **ViewOrder**: Specifies the order in which the views are to be coded.
4.3 MVC Scheme

For encoding the sequences and obtaining experimental results provided in this work, GOP length of 12 pictures is used as shown in Figure 4.1. Meaning that every 12\textsuperscript{th} picture of the base View 0 is an I picture. The syntax of the hierarchical B pictures implemented in H.264/AVC is very flexible and allows multiview GOPs of any length to be used.

![Inter-view/temporal prediction structure with GOP length of 12](image)

**Figure 4.1** Inter-view/temporal prediction structure with GOP length of 12

This scheme first uses inter-view prediction to provide P pictures for even views. Odd camera views are obtained by combining inter-view prediction from 2 adjacent even views and hierarchical B coding structure in temporal direction. For an even number of views, the last view represents a specific case for prediction. View 7 is coded as shown, starting with an inter-view predicted P picture, followed by hierarchical B pictures, which are also inter-view predicted from the previous view.
4.4 Simulcast Scheme

A straight method to compress multiview video is to encode each view independently using the state-of-the-art H.264/AVC encoder. This approach is denoted, in the literature, as simulcast coding. However, since all the cameras capture the same scene through different viewpoints, there is an inter-view statistical expected dependencies between adjacent cameras, which is not exploited in the Simulcast case (Figure 4.2).

![Diagram of Simulcast Scheme](image)

**Figure 4.2** Simulcast using hierarchical B pictures in temporal dimension only with GOP length of 12

4.5 Scheme of Exploiting the Inter-view among all Frames (EIF)

As we seen in Figure 4.2, the inter-view statistical dependencies between adjacent cameras is not exploited in the Simulcast case, therefore we implement a scheme that can exploits all inter-views among the views but do not exploits the temporal predictions. For that the multiview video sequences are combined into one single uncompressed video stream as illustrated in Fig. 4.3. The benefit of this scheme is resulting standard-conforming bitstream and can be decoded by any standard H.264/AVC decoder, but the H.264/AVC encoder must
increase the decoder picture buffer (DPB) size to store all necessary images needed for this scheme as well as a potentially larger number of output pictures per second than it is currently allowed in H.264/AVC.

![Diagram showing reordering of multiview input for compression with GOP length of 12]

**Figure 4.3** Reordering of multiview input for compression with GOP length of 12

### 4.6 Results and Discussions

In order to check the performance of the three schemes, we used three sequences Ballroom, Exit and Vassar. Figure 4.4 represents all available views in one time instance. The top most frame represents the frame 0 and frame 10 of the view 0, while the bottom most one represents these frames of the view 7.

The Ballroom sequence represents a dynamic scene containing fast movement of the dancers within the scene and more objects overlapping than present in the sequences Exit and Vassar. The Exit sequence (in the left of Figure 4.4) represents mostly static scene with few persons slowly moving from the right part of the scene towards the door in the middle of the scene. The Vassar sequence (in the right of Figure 4.4) represent a person passes a route with a few cars and persons.
Figure 4.4 Multiview video test data sequences, frames 0 (left), frames 10 (right), Exit (left) and Vassar (right)
4.6.1 Results

The bit-rate of the MVC stream is controlled by varying the quantization parameter (QP) (29, 31, 34, 37, 40). In general, the higher the QP, the higher the compression ratio, but lower quality of the coded video is provided. Table 4.3, table 4.4 and table 4.5 summarize the obtained coding results for Ballroom, Exit and Vassar respectively. The reported tables show the average PSNR value for all the reconstructed views as a function of the coded bit-rate.

**Table 4.3** MVC, Simulcast and EIF comparison results for ballroom

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Average PSNR Y (dB)</th>
<th>Average bit-rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MVC</td>
<td>Simulcast</td>
</tr>
<tr>
<td>Ballroom</td>
<td>29</td>
<td>36,2190</td>
<td>36,2408</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>35,1931</td>
<td>35,2704</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>33,4770</td>
<td>33,6628</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>31,7615</td>
<td>32,0672</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>29,9607</td>
<td>30,3450</td>
</tr>
</tbody>
</table>

**Table 4.4** MVC, Simulcast and EIF comparison results for exit

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Average PSNR Y (dB)</th>
<th>Average bit-rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MVC</td>
<td>Simulcast</td>
</tr>
<tr>
<td>Exit</td>
<td>29</td>
<td>37,9095</td>
<td>37,9780</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>37,1584</td>
<td>37,2658</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>35,8111</td>
<td>35,9866</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>34,3565</td>
<td>34,6628</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>32,7108</td>
<td>33,0484</td>
</tr>
</tbody>
</table>
**Table 4.5** MVC, Simulcast and EIF comparison results for vassar

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Average PSNR Y (dB)</th>
<th>Average bit-rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MVC</td>
<td>Simulcast</td>
</tr>
<tr>
<td>Vassar</td>
<td>29</td>
<td>36,3679</td>
<td>36,0363</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>35,6160</td>
<td>35,7125</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>34,3152</td>
<td>34,5287</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>32,9330</td>
<td>33,3714</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>31,4261</td>
<td>31,9716</td>
</tr>
</tbody>
</table>

Figure 4.5, Figure 4.6, and Figure 4.7 represent PSNR Y results of encoding Ballroom, Exit and Vassar sequences, respectively. The solid line in these graphs represents average of PSNR Y values across all cameras views obtained by the schemes of coding (MVC, Simulcast and EIF) for a set of targeted average bitrates.

**Figure 4.5** Ballroom sequence coding results (PSNR Y)

**Figure 4.6** Exit sequence coding results (PSNR Y)
The results produced by the presented MVC scheme, utilizing hierarchical B pictures in inter-view and temporal direction are shown by the blue curve (MVC). EIF coding results are represented by the red curve. Independent encoding of each view but with hierarchical B pictures in temporal direction is represented by the green curve (Simulcast).

Figure 4.8 illustrates the subjective quality of reconstructed frames using the MVC, Simulcast and EIF with the approximate bit-rate applied on the sequence Exit in third view (View 3). The left images show frame. (a) the original frame. (b) the reconstructed frame when we have compressed it by using MVC scheme with QP=29, and its results are average bit-rate 254.35 Kbps, PSNR Y is 38.33 dB. (c) the reconstructed frame when we have compressed it by using Simulcast scheme with QP=34, and its results are average bit-rate 242.36 Kbps, PSNR Y is 35.96 dB. (d) the reconstructed frame when we have compressed it by using EIF scheme with QP=40, and its results are average bit-rate 216.12 Kbps, PSNR Y is 31.20 dB. While the right images shows a close-up view of part of these frames.
Figure 4.8 visual qualities of Exit frames in three cases, frames (right) and a close-up view of them (left)
Figure 4.9, Figure 4.10 and Figure 4.11 shows the benefit of exploiting the inter-view/temporal prediction structure described in section (4.3) (MVC Scheme) for three sequences Ballroom, Exit and Vassar respectively. Used view order (reflecting scheme in which views are coded) is 0 2 1 4 3 6 5 7.

**Figure 4.9** Ballroom sequence coded using inter-view/temporal prediction of MVC (QP=29, average bit-rate=638.37 Kbps, average PSNR Y=36.21 dB)

**Figure 4.10** Exit sequence coded using inter-view/temporal prediction of MVC (QP=29, average bit-rate=352.79 Kbps, average PSNR Y=37.90 dB)
The MVC scheme outperforms the Simulcast scheme by about 2 dB, 1 dB and 0,9 dB in Ballroom, Exit and Vassar as shown in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. Meaning that, a good portion of the gain already comes from the exploiting the inter-view among the views (View 0 to View 7) (Figure 4.1). Since no inter-view redundancy is utilized in simulcast structure, as the number of views increases, the compressed output bit-rate increases linearly. Nevertheless the results prove that specific MVC algorithms, namely B pictures in inter-view direction exploiting inter-view statistical dependencies, significantly improve compression performance. As for the EIF scheme, the MVC scheme also outperforms it by about 3 dB, 4,5 dB and 4 dB in Ballroom, Exit and Vassar as shown in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. But the difference at this time has increased. Therefore, the exploiting the temporal prediction performs higher compression than the exploiting the inter-view due to the neighboring frames strongly correlated, namely B pictures is encoded efficiency. As for the exploiting the inter-view, the correlation between adjacent views is less than the other case due to the distance between two cameras is 20 cm (Table 4.1) and this perform a displacement by about 1 to 6 Pixels (Figure 4.4) as well as appearing new objects in the scene, thus B pictures is encoded feebly as well as the shifts between (T0/View 7 and T1/View 0, T1/View 7 and T2/View 0…) in Figure 4.3 reduce the

Figure 4.11 Vassar sequence coded using inter-view/temporal prediction of MVC (QP=29, average bit-rate=393,14 Kbps, average PSNR Y=36,36 dB)

4.6.2 Discussions

The MVC scheme outperforms the Simulcast scheme by about 2 dB, 1 dB and 0,9 dB in Ballroom, Exit and Vassar as shown in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. Meaning that, a good portion of the gain already comes from the exploiting the inter-view among the views (View 0 to View 7) (Figure 4.1). Since no inter-view redundancy is utilized in simulcast structure, as the number of views increases, the compressed output bit-rate increases linearly. Nevertheless the results prove that specific MVC algorithms, namely B pictures in inter-view direction exploiting inter-view statistical dependencies, significantly improve compression performance. As for the EIF scheme, the MVC scheme also outperforms it by about 3 dB, 4,5 dB and 4 dB in Ballroom, Exit and Vassar as shown in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. But the difference at this time has increased. Therefore, the exploiting the temporal prediction performs higher compression than the exploiting the inter-view due to the neighboring frames strongly correlated, namely B pictures is encoded efficiency. As for the exploiting the inter-view, the correlation between adjacent views is less than the other case due to the distance between two cameras is 20 cm (Table 4.1) and this perform a displacement by about 1 to 6 Pixels (Figure 4.4) as well as appearing new objects in the scene, thus B pictures is encoded feebly as well as the shifts between (T0/View 7 and T1/View 0, T1/View 7 and T2/View 0…) in Figure 4.3 reduce the
process of compression due to the very low correlation between them. So MVC scheme achieves higher compression efficiency at lower bitrates and should also be taken into account for services and applications using limited capacity communication channels like video for mobile services.

As reported in the Tables, significant visual quality was observed from close-up view of part of the frames in Figure 4.8 when we have compared visual quality at the approximate bit rate. So MVC scheme achieve a better visual quality than Simulcast and EIF scheme and as reported in Table 4.2 for Exit sequence, the MVC scheme required only about half the bit rate to achieve equivalent or better visual quality than the EIF scheme.

Figure 4.8, Figure 4.9 and Figure 4.10 shows the benefit of exploiting the interview/temporal prediction structure by using view order is 0 2 1 4 3 6 5 7. The results presented on these graphs show that even views require more bit-rate than odd ones. This conclusion is expected, because inter-view prediction is used only to provide P pictures at the beginning of each GOP within individual even views. It can also be seen that odd views 1, 3 and 5 are using less bit-rate because all the pictures contained within these views are obtained by combining inter-view prediction from 2 adjacent even views and hierarchical B coding structure in temporal direction. Additionally, it is noticeable that view 7 consumes more bit-rate than views 1, 3 and 5 due to the fact that it’s predicted exploiting inter-view dependency from only one adjacent view (view 6).

4.7 Conclusion

In this chapter, we have concluded the benefit of utilize the inter-view prediction by the obtained results that indicate the MVC scheme provides a higher PSNR for the same bit-rate in comparison with Simulcast scheme which doesn’t exploit the inter-view dependencies and in comparison with EIF scheme which doesn’t exploit the temporal prediction.
General Conclusion

We have presented in this work the video coding tools, motion compensated prediction, transform coding, quantization and entropy coding, form the basis of the reliable and effective coding model that has dominated the field of video compression for over 10 years and this coding model is at the heart of the H.264/AVC standard. As well as we have provided an overview of H.264AVC that be used for MVC due to provides tools that can be used to deliver efficient, flexible and robust video compression for a wide range of applications, from low-complexity, low bit-rate mobile video applications to high-definition broadcast services.

And we have presented also the basic concepts of multiview video coding and we have concluded the benefit of exploiting the inter-view/temporal prediction i.e. MVC Scheme that is provides a higher PSNR for the same bit-rate in comparison with Simulcast scheme which doesn’t exploit the inter-view dependencies and in comparison with EIF scheme which doesn’t exploit the temporal prediction. And by comparing Simulcast Scheme by EIF Scheme, we have shown the exploiting the temporal similarities a high compression than EIF Scheme and this explain the correlation between temporally successive images of each video is strong than the correlation between adjacent camera views.

From these results, the exploiting the inter-view seem low, therefore there is another method has been proposed to exploit scene geometry for improving compression efficiency and to generate a virtual view as an additional reference for inter-view prediction [21, 22, 23, 24, 25]. So the next proposed method is view synthesis prediction (VSP) because this technique has been introduced to take advantage of these geometric redundancies.
Reference Library


