Poznań University of Technology

Chair of Multimedia Telecommunications and Microelectronics

Doctoral Dissertation

“Motion information coding for 3D video compression”

Jacek Konieczny

Advisor: Prof. Marek Domański

Poznań, 2012

Politechnika Poznańska

Katedra Telekomunikacji Multimedialnej i Mikroelektroniki

Rozprawa Doktorska

“Kodowanie informacji o ruchu dla kompresji sekwencji trójwymiarowych”

Jacek Konieczny

Promotor: prof. dr hab. inż. Marek Domański

Poznań, 2012

This dissertation is dedicated to my beloved parents Urszula and Jacek, who gave me a wonderful childhood and all the opportunities.

I would like to thank all important people in my life, who have always been in the right place and supported me in difficult moments.

Rozprawa ta dedykowana jest moim ukochanym rodzicom Urszuli i Jackowi, którzy dali mi cudowne dzieciństwo oraz wszelkie możliwości.

Chciałbym podziękować wszystkim ważnym osobom w moim życiu, które zawsze były we właściwym miejscu i wspierały mnie w trudnych chwilach.

Table of Contents

Abstract

The dissertation deals with the problem of representation of motion information in 3D video codecs. Existing techniques of motion information representation in state-of-the-art multiview video codecs are thoroughly discussed. The problem of motion information prediction and representation in coding of 3D video with additional depth information is stated. The possible solutions are presented. Similarities and correlations in inter-view predicted motion fields are researched.

The author proposes several techniques of depth-based inter-view motion information prediction and coding. Different variants of the proposed methods are discussed, including efficiency and complexity aspects. In particular, the efficient modes of motion information coding in state-of-the-art and future multiview video codecs are also presented in the thesis.

Proposed algorithms have been experimentally tested and compared against other methods. The obtained results are presented in the dissertation.

Streszczenie

Rozprawa dotyczy problemu reprezentacji informacji o ruchu w trójwymiarowych kodekach wizyjnych. W pracy omówiono istniejące techniki reprezentacji informacji o ruchu, stanowiące obecny stan techniki w dziedzinie kompresji trójwymiarowych sekwencji wizyjnych. Sformułowany został problem predykcji oraz reprezentacji informacji o ruchu przy kodowaniu trójwymiarowych sekwencji wizyjnych wzbogaconych o informację o głębi. Przedstawiono również możliwe rozwiązania tego problemu. Przebadano podobieństwa i korelacje występujące w przewidywanych między-widokowo polach ruchu.

Autor zaprezentował kilka technik między-widokowej predykcji i kodowania informacji o ruchu w oparciu o dostępną informację o głębi. Omówiono różne warianty zaproponowanych metod, z uwzględnieniem zagadnień efektywności i złożoności. W szczególności, w pracy zaprezentowano wydajne tryby kodowania informacji o ruchu dla potrzeb obecnych i przyszłych kodeków wielowidokowych.

Zaproponowane algorytmy zostały sprawdzone eksperymentalnie i porównane z innymi stosowanymi metodami, a uzyskane rezultaty eksperymentów przedstawiono w rozprawie.

List of symbols and abbreviations

3D-TV	-	Three-dimensional Television
ASIM	-	Angular Similarity Measure
AVC	-	Advanced Video Coding
BJM	-	Bjontegaard metric
CU	-	Coding Unit
DBMP	-	Depth-Based Motion Prediction
DCT	-	Discrete Cosine Transform
DIBR	-	Depth Image Based Rendering
DV	-	Disparity Vector
f	-	Focal length
FPDBP	-	Forward Projection Depth-Based Prediction
FPIVD	-	Forward Projection Inter-View Direct
FPIVS	-	Forward Projection Inter-View Skip
FTV	-	Free-view Television
HEVC	-	High Efficiency Video Coding
Inc_{DBP}	-	Increase in encoding/decoding time due to depth-based projection of pixel coordinates, occluded area detection and unassigned area filling
Inc_{MCP}	-	Increase in encoding/decoding time due to point-to-point motion-compensated prediction
IPDBP	-	Inverse Projection Depth-Based Prediction
IPIVD	-	Inverse Projection Inter-View Direct
IPIVS	-	Inverse Projection Inter-View Skip
ISO	-	International Organization for Standardization
IVD	-	Inter-View Direct
IVS	-	Inter-View Skip
JMVM	-	Joint Multiview Model
$\boldsymbol{K}$	-	Projection matrix
MB	-	Macroblock
MPEG	-	Moving Pictures Experts Group
MS	-	Motion Skip
MSIM	-	Magnitude Similarity Measure

MV	-	Motion Vector
MVC	-	Multiview Video Coding
MVD	-	Multiview Video Plus Depth
MV-HEVC	-	Multiview High Efficiency Video Coding
$\boldsymbol{o}$	-	Principal point and its pixel coordinates
$\boldsymbol{O_c}$	-	Optical center of the camera / center of the camera coordinate system
$\boldsymbol{O_w}$	-	Center of the world coordinate system
o_x, o_y	-	Pixel coordinates of the principal point
PCC	-	Pearson Correlation Coefficient
PCCx	-	PCC calculated for horizontal components of motion vectors
PCCy	-	PCC calculated for vertical components of motion vectors
PCCavg	-	Averaged value of PCCx and PCCy
PSNR	-	Peak Signal-to-Noise Ratio
PSNRY	-	Peak Signal-to-Noise Ratio for luminance
QD	-	Quantization parameter for depth
QP	-	Quantization parameter for texture
$\boldsymbol{R}$	-	Rotation matrix
SEI	-	Supplemental Enhancement Information
s_x, s_y	-	Dimensions of the pixel in horizontal and vertical direction
$\boldsymbol{T}$	-	Translation matrix
UHDTV	-	Ultra High Definition Television
VCEG	-	Video Coding Experts Group
VSIM	-	Vector Similarity Measure
X, Y, Z	-	Coordinates of point $\boldsymbol{P}$ in 3D space
x, y, z	-	Coordinates of perspective projection of point $\boldsymbol{P}$ onto image plane Π
x_u, y_u	-	Pixel positions in the image plane Π
X_c, Y_c, Z_c	-	Camera coordinate system
X_w, Y_w, Z_w	-	World coordinate system
λ	-	Homogeneous scaling factor
Π	-	Image plane

Chapter 1

Introduction

1.1. Scope of the dissertation

The dissertation deals with improving compression of 3D video sequences. This issue is of fundamental importance for emerging 3D video services, including the new generation of three-dimensional television (3D-TV), and television with free navigation in the scene, called the free-viewpoint television (FTV). These new types of video systems are attracting a lot of interest as they offer possibilities that are far beyond the ones provided by the commercially available stereoscopic systems. FTV allows viewers to change their viewpoint without concern for the physical position of the camera. Such virtual viewpoints are created by means of a view synthesis which utilizes available texture and information about visual scene geometry in order to generate visual content for the virtual camera. On the other hand, the new generation 3D technologies provide more realistic depth effect to a viewer. As result, reproduction of movement parallax or perception of stereoscopic depth without the need to use special glasses become possible. It is expected that this new generation of 3D video will offer advantages in many fields, including entertainment and education. Hence, in the future, these types of video may be offered in various applications, like broadcast television, internet streaming or mobile video. However, these future video systems require transmitting of very large data streams to the recipient. Consequently, the capabilities of existing data transmission technologies will be pushed to the limits, especially in case of the mobile applications. As a result, there is a strong motivation to efficiently utilize existing correlation between pictures of encoded video content and develop new prediction techniques to increase the compression ratio of 3D video.

In the abovementioned new generation multimedia systems, multiview video content is used on some stage of video generation and presentation [Smo06, Kauf07]. A multiview video is a set of video sequences recorded at the same time instance from different viewpoints and representing the same scene. In the new generation 3D video applications, the multiview video is very often accompanied with additional depth information, e.g. stereoscopic depth, which describes geometry of the visual scene and is usually represented either directly by depth samples or indirectly by disparity samples. This particular kind of a multiview video content is usually referred to as the 3D video in a multiview video plus depth (MVD) format [Smo07]. Depth information in multiview video may be acquired or estimated using dedicated algorithms for one or more viewpoints. An exemplary texture and corresponding depth from a 3D video sequence are presented in Fig. 1.1. This dissertation regards to 3D video sequences especially.

a) b)

Fig. 1.1. An example of: a) texture and b) depth from 3D video test sequence *Poznan Street*.

A typical multiview video system contains modules for video acquisition, transmission and presentation (see Fig. 1.2). In such a system, depth information is usually utilized for view synthesis purposes. Based on a texture available for some views, content displayed in other viewpoints is synthesized using dedicated depth-based rendering techniques [Kauf07]. Consequently, view synthesis requires information about texture and depth, but also parameters describing location of the viewpoints in the visual scene. These parameters are often referred to as camera parameters. In order to supply the receiver with all information required for a proper presentation of a multiview video content, video or, in case of 3D video content, video plus depth together with all necessary system parameters need to be encoded and delivered to the decoder. Unfortunately, estimation of accurate depth maps is still a complex and time consuming problem, which prevents its usage in practical real-time applications in the decoder. In this dissertation, we will focus on the part of the multiview video system related to encoding and decoding process.

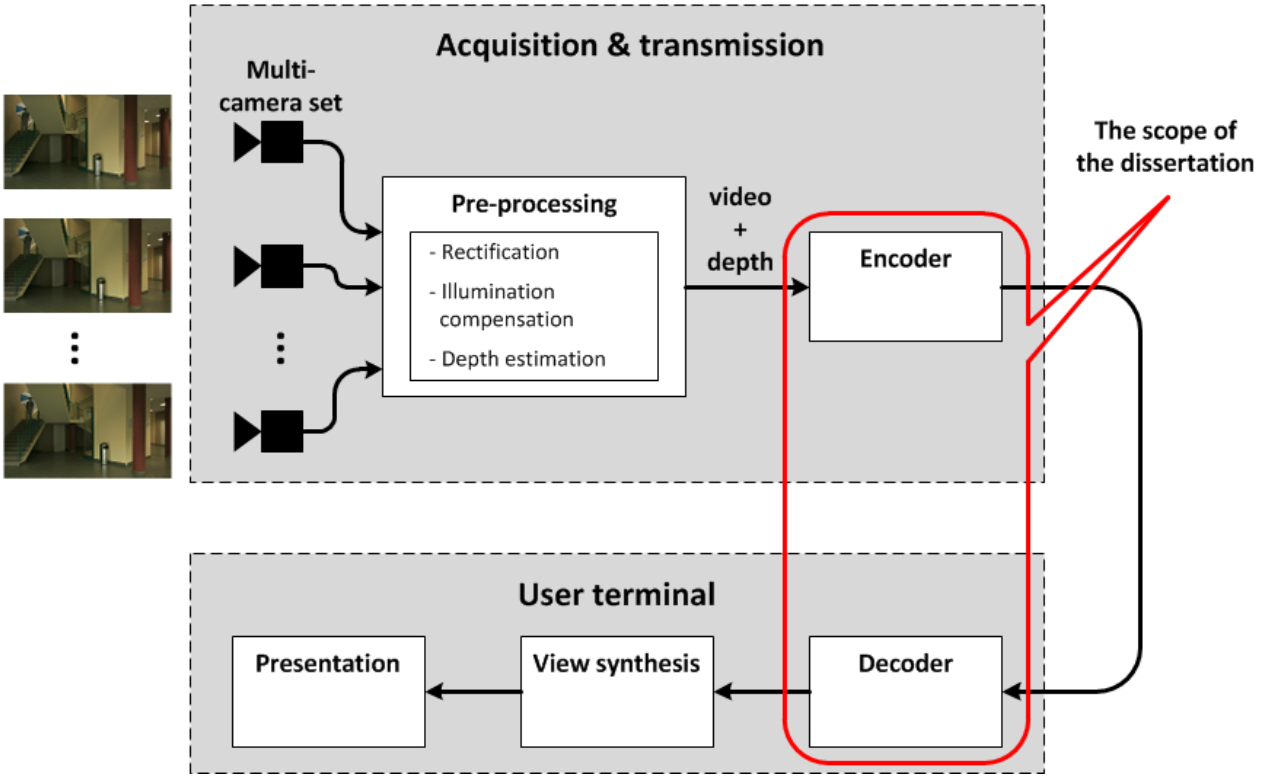


Fig. 1.2. Multiview video system.

To date, many efforts have been made to develop more efficient techniques for the multiview and 3D video sequence compression. In natural video, high level of spatial and temporal redundancy exists, which can be efficiently removed during the video coding process in order to increase the compression ratio. This can be obtained with almost negligible impact on the subjective quality of the resultant video. The most efficient and widely used class of video codecs, commonly called the hybrid video codecs [Dom98, Ska93, Shi00, Ohm04, Dom10], use motion-compensated prediction and prediction residuals coding to achieve a video compression. Motion-compensated prediction is usually performed in small, rectangular blocks. The encoder estimates motion vectors for each block and transmits this information to the decoder in the bitstream. As a consequence, the resultant bitstream produced by a typical hybrid video codec contains three main types of data: motion vectors, transform coefficients of prediction residuum and control data (side information) [Dom98, Ska93, Ric02]. Consequently, reduction of the part of the bitstream representing motion vectors will result in considerable gains in compression performance of the codec.

In the multiview video, additional spatial redundancy exists. If the distance between neighboring viewpoints of multiview video is small, high correlation between content of video sequences obtained from these viewpoints exists [Feck05, Merk07, Su06]. This inter-view correlation may be exploited to reduce the amount of data that must be transmitted from acquisition

to presentation module of the multiview video system. A simple approach is to utilize pictures from neighboring views for the inter-frame prediction [ISO11, Vet11]. On the other hand, in the new generation 3D video systems, a more advanced approach can be used. The presence of additional depth information in encoded video content makes the problem of video compression more complex. However, it also provides the possibility to apply sophisticated methods known from computer graphics to this process. Since depth information describes geometry and location of objects in the visual scene, new prediction methods based on 3D projection become possible. Consequently, improvement in compression efficiency of the multiview codec can be made [Mart06, Shim07]. Following this reasoning, the dissertation presents results of research aimed at increasing compression ratio of the 3D video sequences in which depth information is utilized for efficient inter-view prediction of motion information.

The perspective of growing demand for the multiview and 3D video coding technology motivated the Moving Picture Experts Group (MPEG) of International Organization for Standardization (ISO) to start a new activity in 2004, which aim was to develop a Multiview Video Coding (MVC) standard [Pfis04]. The result of this works was a new multiview video codec based on the AVC video coding technology, established by the MPEG committee as an ITU-T and ISO/IEC standard in 2009 [ISO11, Ric10]. The basic approach introduced in MVC is an inter-view prediction with disparity compensation, which uses a mechanism similar to motion compensation of the AVC video codec, however, with reference frames from neighboring views. This simple idea resulted in considerable coding gains when compared to simulcast coding (independent coding of each view) of a multiview video. Nevertheless, achieved compression ratio is still below requirements of the future 3D television applications. Moreover, MVC does not describe any dedicated method for the multiview video plus depth representation.

As a result, in 2010, MPEG began works on new techniques for a 3D video plus depth coding, that should allow efficient representation of a 3D video for the future 3D television applications [MP11b]. The author of this thesis actively joined the MPEG team to develop new algorithms for more efficient encoding of motion information in 3D video codec. As a result of research, a number of documents were contributed to MPEG and some of ideas presented in this dissertation were incorporated into MPEG works. In particular, the algorithms described in this dissertation were utilized in Poznań University of Technology proposal for MPEG's call on the 3D video coding technology that achieved outstanding results (refer to Chapter 7).

Algorithms for the inter-view prediction of motion information are important in improving the compression of a multiview video and, hence, are intensively researched in the area of video

sequences coding. However, there is still no ultimate solution for efficient representation of motion information in a multiview or 3D video codec, that would utilize the availability of depth information describing location of objects in the visual scene. As a consequence, in this dissertation, the problem of increasing the coding efficiency of a 3D video codec by development of new depth-based inter-view motion information prediction algorithms is investigated.

1.2. Goals and thesis of the dissertation

The **goal** of this dissertation is to develop new techniques for compression of the 3D video in the multiview video plus depth format. New methods of inter-view prediction in the 3D video are to be proposed, allowing to reduce the bitrate compared to the currently known systems by providing a more efficient representation of motion information and utilization of available depth information. New algorithms and tools are to be researched in order to improve the overall compression efficiency with minor impact on complexity and requirements of multiview video codecs. The proposed techniques are to be experimentally evaluated to accurately assess their actual impact on the coding efficiency of existing and future multiview video codecs.

The following **assumptions** are made in this dissertation:

- state-of-the–art hybrid video codecs are used as a basic video compression technology,
- depth information describing visual scene is available at the decoder,
- the proposed techniques should assure possibly high compatibility with existing state-of-the-art coding techniques.

The **main theses** of the dissertation are:

- It is possible to improve efficiency of motion information representation in coding of 3D video in the multiview video plus depth format by exploiting the correlation between motion fields of neighboring views and utilizing the available depth information.
- It is possible to develop techniques of representation of motion information that are competitive to the methods described in literature, developed simultaneously with the author's investigations.

1.3. Research methodology

The goal of the dissertation is to study whether it is possible to improve the coding efficiency of contemporary multiview video coding techniques by creating new motion information inter-view prediction methods that utilize sophisticated modeling of the visual scene. Consequently, the starting point for the research was related to existing techniques of motion information prediction in multiview video coders. A special attention was given to the most recent and the most advanced solutions proposed during work of MPEG committee on the multiview video coding standard, the MVC video codec [ISO11, Yang09, Koo06]. These techniques have been analyzed and their efficiency for a motion data encoding has been examined and experimentally tested.

A problem of motion information prediction in coding of 3D video sequences with additional depth information describing location of objects in the visual scene has been formulated. New techniques for such depth-based inter-view prediction of motion information in multiview video coding have been proposed. Due to high, multi-dimensional complexity of the problem, the process of developing new prediction techniques have been decomposed into stages. Next, these developed methods have been implemented within the reference anchor software and experimentally tested in order to check their usefulness in further algorithms for multiview video coding. Conclusions drawn from the experimental results have been utilized to further improve the proposed methods. This process have been repeated in subsequent iterations.

As the reference anchor, the following multiview video coding techniques have been used during the experimental verification of the research:

- MVC (Multiview Video Coding), developed by MPEG as *annex H* of AVC video coding standard (ISO/IEC MPEG-4 part 10, ITU-T H.264),
- JMVM (Joint Multiview Model), developed by MPEG during work on MVC video coding standard, and
- MV-HEVC, a HEVC-based multiview video codec, developed originally at Poznań University of Technology.

The first of the abovementioned video codecs is a multiview video codec based on the AVC video coding technology that was established as a new ITU-T and ISO/IEC standard in 2009 as a result of the work of MPEG committee [ISO11, Chen09]. The codec is briefly presented in Section 2.4.1 and has been used as the basis for implementation of algorithms proposed by the author of this work.

The JMVM multiview video codec was developed by the MPEG committee during the work on MVC coding standard and refers to the reference model described in [MP08, Pan08]. The JMVM is a reference codec model of previous version of MVC that contains a number of additional coding tools, designed especially for a multiview video coding. One of the multiview coding tools of JMVM is an inter-view motion information prediction tool named Motion Skip, which obviously aims in the area of interest of this dissertation. The JMVM multiview video codec and the Motion Skip coding tool are briefly described in Section 2.4.1 and 2.5.

The third codec, MV-HEVC [Dom11], had been build using the new generation video codec named HEVC (High Efficiency Video Coding) currently developed by the MPEG committee [MP11, Dom12]. The reference software version of the HEVC test model HM 3.0 [MP11a] was the basis for creating an implementation of multiview video compression scheme similar to MVC technology in which the AVC core was substituted by the HEVC coder. As a result, MV-HEVC codec constitutes a basis for future multiview video codec that provides mechanisms for inter-view prediction known from MVC to exploit the inter-view correlation that exists in a multiview video and reduce the bitstream representing the side views. The MV-HEVC codec is briefly presented in Section 2.4.2.

The abovementioned codecs have been chosen as they follow the most recent worldwide trends in the multiview coding technology and because the author had free access to their source code, so that modifications could be introduced in their algorithms.

During the research, efficiency of motion information coding and efficiency of overall compression have been examined: the existing techniques of motion information encoding have been compared against the original solutions proposed in the dissertation. For this purposes, correlation between estimated and inter-view predicted motion fields of multiview sequences, as well as rate and distortion have been measured. For measuring the distortions, objective quality measure PSNR and the Bjontegaard metric have been chosen, as discussed in Section 2.2.1 and 2.2.2. In order to determine the complexity of the proposed methods, execution times of the researched video codecs have been measured.

In all experiments, the standard multiview and 3D video test sequences available through the standardization committee of MPEG have been used. These test sequences were chosen among others by the MPEG committee in order to perform comparisons and experiments during developing of new tools and techniques for a multiview and 3D video compression [MP11b] and contain various types of motion and textures. The bitrate ranges of compressed video sequences

have also been chosen to meet the requirements announced by Video Coding Experts Group (VCEG) and MPEG organization during comparison of video coders [Tan08, Bos11].

1.4. Thesis overview

The dissertation is organized as follows. In Chapter 2 the basic information about multiview television systems is presented. Hybrid video coder together with the paradigm of motion-compensated prediction and algorithms of motion estimation and representation are also described. Measures of correlations of motion vector fields are introduced. Finally, multiview video coding techniques, with an emphasis on the most advanced motion data prediction algorithms, known from the literature, are discussed in detail.

Chapter 3 contains a description of two original author's techniques of inter-view prediction for multiview video codec. Different variants of these methods are discussed, including efficiency and complexity aspects.

In Chapter 4 motivation for inter-view prediction of motion data in multiview video codecs is presented. Correlations in inter-view predicted motion fields are discussed. Also, experimental results are presented regarding the contribution of individual components of the bitstream in multiview hybrid video coding.

In Chapter 5 possibilities to utilize the original author's methods of inter-view motion prediction in state-of-the-art and future multiview video codecs are discussed. Different variants of AVC and HEVC-based multiview video codecs are proposed.

Chapter 6 presents experimental results obtained for various codec variants introduced in Chapter 5. The efficiency and complexity of these implementations are discussed based on the results achieved by proposed multiview codecs.

In Chapter 7 presents considerations on utilization of the original author's methods of inter-view motion prediction in the future 3D video codecs. The most important results of the cooperation with the MPEG committee are discussed.

Chapter 8 contains a summary of achieved results and conclusions.

Chapter 2

Selected issues in digital video compression

2.1. Hybrid video coding

2.1.1. General concept

Hybrid video coding is a method of video representation which utilizes a mechanism of prediction with motion compensation in order to eliminate the existing temporal redundancy in video sequences and block-based transform coding of prediction residuals [Ska98, Dom98, Sad02]. Moreover, it is the only method of motion picture coding widely used in practical applications, especially, if the high coding efficiency is concerned [Dom10].

The general concept of hybrid video coding is based on the inter-frame prediction with motion compensation, coding of the cosine block transform coefficients computed for the prediction residuals and, finally, the entropy coding. A block diagram of a typical modern hybrid video encoder with motion-compensated prediction is presented in Fig. 2.1.

As presented in the diagram, a currently encoded input frame F_i is compared with its prediction $\widetilde{F}_i$. The difference between these two signals forms a residual signal which is lossy coded using transform coding and quantization of transform coefficients. The reconstruction step size for the quantizer determines the quality of encoded picture and is usually controlled by a quantization parameter (QP). Eventually, quantized values of transform coefficients are subjected to entropy coding and encoded into a bitstream. The better prediction $\widetilde{F}_i$ is, the smaller residual signal is

produced. Consequently, less bits need to be encoded into bitstream, which obviously leads to better compression efficiency of the coder.

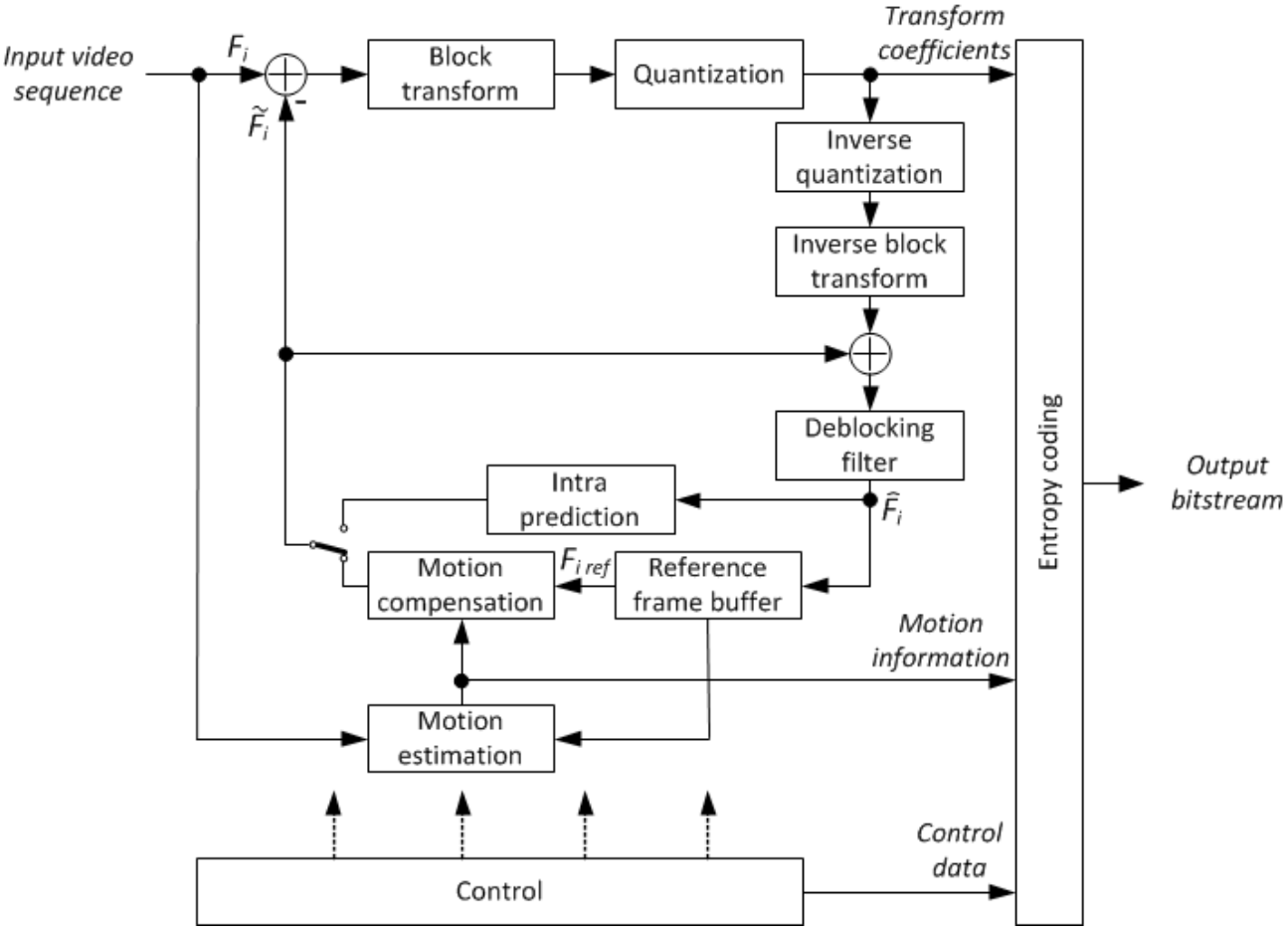


Fig. 2.1. Hybrid video encoder with motion-compensated prediction.

Inverse quantization and inverse transform are performed in the reconstruction loop of the encoder to form a residual signal which is added to $\widetilde{F}_i$ prediction and filtered using a deblocking filter. As a result, a reconstruction $\widehat{F}_i$ of the current frame F_i is obtained and stored in the reference frame buffer used for motion estimation process.

Prediction signal $\widetilde{F}_i$ is obtained using motion-compensated prediction from reference frame $F_{i\,ref}$ or using spatial intra-frame prediction from reconstructed current frame $\widehat{F}_i$. For some reasons, the inter-frame prediction may not be efficient for some areas of encoded picture. In such cases, encoder uses intra-frame prediction in which samples are predicted based on already encoded neighboring samples from the same picture.

As a result, the output bitstream of the hybrid video encoder contains three main types of information which are required for appropriate reconstruction of the input video sequence at the decoder:

- control data (prediction modes, block partitioning, frame resolution, etc.),
- transform coefficients (quantized transform coefficients of prediction residuals),
- motion information (motion vectors, reference frame indices).

Now, let us clarify the basic idea of motion-compensated prediction which is one of the fundamental concepts used in hybrid video coding. The main reason of utilizing the motion-compensated prediction is to remove temporal redundancies from a video sequence. Subsequent frames are predicted using previously encoded frames. This constitutes a temporal inter-frame prediction mechanism of hybrid video codec.

The paradigm of motion-compensated prediction is described by the following formula:

$$\widetilde{F}_i(x,y) = F_{i\,ref}(x + mv_x, y + mv_y) \tag{2.1}$$

where (x, y) are horizontal and vertical coordinates determining location of an image pixel, $\widetilde{F}_i$ is a prediction of the current frame, $F_{i\,ref}$ is a reconstructed reference frame (i.e. one of preciously encoded frames selected as the reference), (mv_x, mv_y) are horizontal and vertical components of the motion vector. Consequently, the final reconstruction of the current frame is calculated according to the equation:

$$\widehat{F}_i(x,y) = \widetilde{F}_i(x,y) + \widehat{\Delta F}_i(x,y) \tag{2.2}$$

where $\widehat{F}_i$ is a reconstruction of the current frame and $\widehat{\Delta F}_i$ is a reconstructed prediction residual.

In Eq. 2.1, motion vector (mv_x, mv_y) determines displacement value which minimizes prediction error, i.e. difference between value of a sample in current frame $F_i(x, y)$ and its prediction $\widetilde{F}_i(x, y)$. In order to perform motion-compensated prediction, motion vectors have to be estimated in the encoder and transmitted to the decoder.

The encoder searches for motion vector components in the process called motion estimation, which is one of the most complex stages performed in the hybrid video codec. The motion estimation itself may consume 40-70 [%] of computing power used for video sequence encoding [Dom10]. The most widely used method of motion estimation in video compression is the block matching algorithm [Jai81, Ska98, Sad02, Dom10]. In this approach, a common motion vector is determined for a rectangular block containing points from currently encoded frame. Such motion vector minimizes the criterion of distortion between blocks from current and reference frame [Kri97, Dom98]. Consequently, the algorithm finds the matching block in reference frame that matches the current block best (see Fig. 2.2).

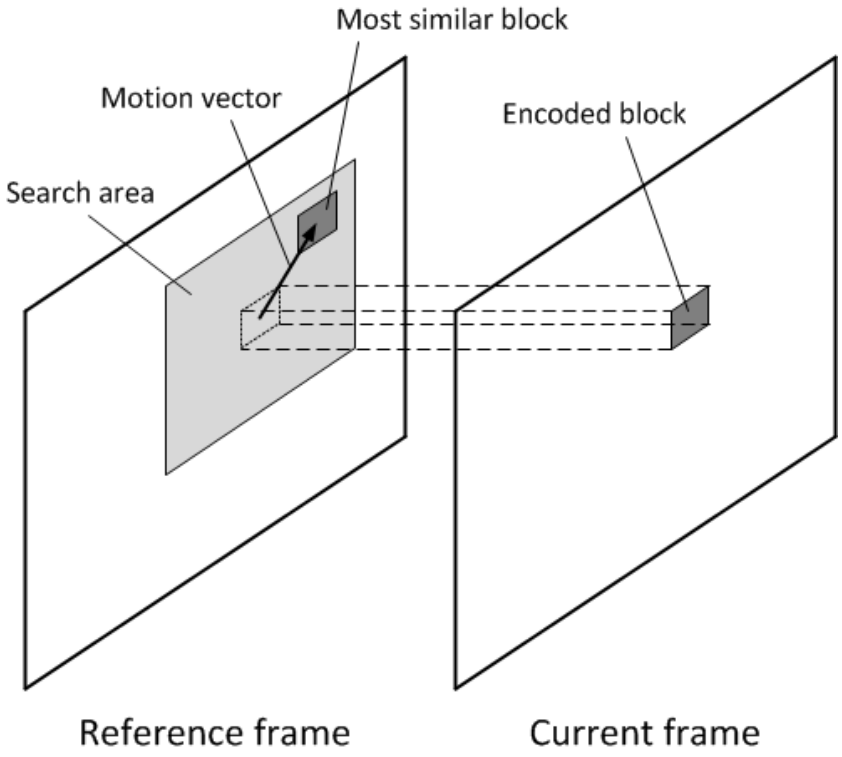


Fig. 2.2. Motion estimation using the block matching algorithm.

While early video codecs utilized motion-compensated prediction for blocks of 16×16, 8×16 or 8×8 luminance samples [Kog81, Nin82, Eri85], new generation video codecs apply variable-size blocks to further increase the video compression efficiency [Flier04]. The most advanced state-of-the-art video codecs use variable block size from 16×16 to even 4×4 luminance samples [ISO11].

Every block which uses motion-compensated prediction requires at least one motion vector to be sent in the bitstream for proper reconstruction in the decoder [ISO93, ISO94, ITUT05, ISO11]. However, more than one motion vector can be used for prediction. As a result, the following types of frames can be specified: P-frames (*predictive*) , B-frames (*bidirectional* or *bi-predictive*) and I-frames (*intra-predicted*). In P-frames only forward prediction, i.e. prediction from the past, is allowed, while in B-frames forward, backward and bidirectional (both forward and backward) predictions can be used. In this case backward prediction means prediction which utilizes reference frames from encoded sequence that are located in future relative to the current frame. The applicability of bidirectional prediction can further reduce the energy of prediction residuals [Str96, Dom10] and consequently, it is widely used in all advanced hybrid video coders. In I-frames, only intra-frame prediction is allowed, i.e. no other frame except the current frame can be used as the reference. Thus, no temporal prediction of sample values is performed. Because I-frames are

encoded independently of other frames, they are extremely useful for encoding the first frame of a sequence or to insert random access points into the bitstream.

Details of the motion compensated prediction methods used in the most advanced state-of-the-art and emerging video coding standards will be discussed in the following sections.

2.1.2. Advanced Video Coding (AVC)

The Advanced Video Coding (AVC) is an international video coding standard (ISO/IEC MPEG-4 part 10, ITU-T H.264) [ISO11] which first release was published in 2003. Despite the lapse of almost 10 years, it is still considered as the state-of-the-art solution for advanced video coding that follows the block-based hybrid video coding approach. Although the basic design of AVC is very similar to earlier video coding standards (e.g. H.261, MPEG-1, H.262/MPEG-2, H.263, or MPEG-4 Visual), AVC introduces new features to achieve a significant improvement in compression efficiency when compared to any prior video coding standard [Wie03, Sul05, Marp06]. In particular, AVC was reported to reduce the bitrate by almost two times against H.262/MPEG-2, while preserving the same video quality [Dom10]. Additionally, the most significant difference relative to previous video coding standards is the increased flexibility and adaptability of the AVC design [Vet11].

In AVC pictures are partitioned into smaller coding units called *slices*, which in turn are subdivided into *macroblocks*. Each macroblock (MB) covers a rectangular picture area of 16×16 luma samples. AVC supports three basic slice coding types: I slices, P slices and B slices, which specify the degree of freedom for generating the prediction signal and a set of available coding tools for each macroblock within the slice.

AVC specifies many new coding tools and solutions for advanced video coding. Some of the most significant improvements introduced in H.264/MPEG-4 AVC are briefly described below. More detailed information can be found in [ISO11, Wie03, Sul05, Marp06, Dom10].

- Adaptive entropy coding – two methods of conditional probability distributions modeling can be selected: UVLC/CAVLC (universal variable length coding/context-based adaptive variable length coding) and CABAC (context-based adaptive binary arithmetic coding). The later one utilizes arithmetic coding and provides more sophisticated mechanism for employing statistical dependencies, which in turn leads to typical bit rate savings of 10–15 [%] relative to CAVLC [Vet11].
- Integer 4×4 and 8×8 transforms – this enables fast and efficient implementation of discrete cosine transform (DCT) and inverse DCT on 16-bit fixed-point processors.

- Adaptive deblocking filter – this especially designed filter operates within the motion-compensated prediction loop to reduce blocking artifacts – the most disturbing artifacts in block-based coding.
- Directional intra-picture prediction modes – spatial intra-picture prediction is performed using the decoded samples of preceding neighboring blocks. Intra-picture prediction can be applied individually to each 4×4 or 8×8 luma blocks, or to the full 16×16 macroblock.
- Variable block size motion-compensated prediction with multiple reference pictures – partitioning of a macroblock into blocks of 16×16, 16×8, 8×16, or 8×8 luma samples can be applied. In case of partitioning into four 8×8 blocks, each of these sub-macroblocks can be further split into 8×4, 4×8, or 4×4 blocks. This enables a better adaptation to the shape of objects moving in the scene of encoded sequence. Moreover, the reference picture to be used for inter-picture prediction can be independently chosen for each 16×16, 16×8, 8×16 or 8×8 macroblock motion partition. For P slices, one motion vector and reference picture index is transmitted for each inter-picture prediction block. In B slices, up to two motion vectors and reference picture indices can be chosen for each block. Also, the resolution of motion vectors was increased to ¼ luminance point, which also leads to better accuracy of the motion-compensated prediction.

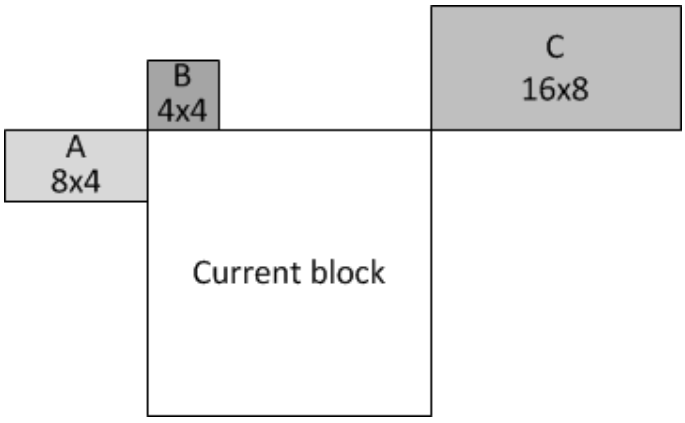


Fig. 2.3. Median prediction of motion vectors in AVC – example for various block sizes (based on [Ric10]).

In addition, AVC specifies a method of effective motion information prediction, which, in many cases, allows for resignation of transmitting motion vectors and reference picture indices for a macroblock. In particular, special modes referred to as *Direct* (B slices) and *Skip* (P and B slices) modes are introduced [Wie03, Tou05], in which the motion information is simply derived based on previously encoded neighboring regions without the necessity of indicating it by the macroblock

syntax. The main difference between Direct and Skip modes is the fact that no prediction residual signal is transmitted for the skip-coded macroblocks.

Both Direct and Skip modes use so-called *median prediction* algorithm to calculate the median of each component of motion vectors assigned to three selected neighboring blocks of the current block (see Fig. 2.3). The result of the median is the predicted motion vector for the current block. The median prediction usually performs very well - motion vector prediction error produced by such predictor is often less or close to one sampling period [Lang06, Dom10].

As a result, Direct and Skip modes provide a very efficient way of encoding inter-predicted blocks of samples in AVC codec. In order to fully utilize the potential of these two modes, a dedicated macroblock mode signaling strategy is used [ISO11]. In case of the Skip mode, the binary valued *skip_flag* is signaled for each macroblock prior to any other parameter. If *skip_flag* is equal to 1, the current macroblock is skipped and no further parameters are sent for this macroblock. Otherwise, syntax element describing the macroblock mode (*mb_type*) and other parameters required for the selected mode are transmitted. While for the Direct mode, *skip_flag* is equal to 0 and *mb_type* indicate the usage of Direct mode by means of the shortest available code. Next, elements for coding the prediction residual signal are transmitted, however, no parameters describing motion information are encoded. Consequently, among all available inter-predicted modes, Direct and Skip modes are signaled using the most limited syntax as possible, which is also the reason for naming them the *low-cost modes* of AVC codec.

2.1.3. High Efficiency Video Coding (HEVC)

In 2010, ITU-T VCEG and ISO/IEC MPEG launched a joint video coding standardization activity, called the Joint Collaborative Team on Video Coding (JCT-VC), to develop a video coding standard for High Efficient Video Coding (HEVC) [Wie10]. HEVC is designed in order to efficiently compress high and very high resolution video data (UHDTV), but also with the aim of wireless telecommunication applications. The final draft of HEVC standard is expected to be specified in the beginning of 2013.

The Call for Proposals for new video compression technology [ISO10] received twenty-seven proposals with several of them able to provide the same subjective quality of the AVC High profile at approximately half the bitrate [Sul10]. Based on these proposals, JCT-VC developed a HEVC Test Model (HM) that is still emerging. To improve the compression efficiency beyond the AVC standard, a number of novel coding tools have been introduced into previous structure of a hybrid video coder [Wie11]:

- Larger block sizes - the picture is subdivided into Large Coding Units (LCU), which correspond to area of 64×64 luminance samples. Each LCU can be recursively subdivided into smaller Coding Units (CU), according to the quadtree pattern [Han10, Karc10], until the smallest CU size of 8×8 is reached. Analogous to macroblocks in AVC, a CU can be inter- or intra-predicted. Prediction type and flag indicating whether the block is skipped or not are also defined on the CU level. Every leaf CU of the quadtree contains one or more Prediction Units (PU) and Transform Units (TU) [Bos10, Marp10]. PU defines CU split into rectangular blocks of 2N×2N, 2N×N, N×2N and N×N size. TU signals transform related information and residual data.
- DCT calculated for TU blocks of size 4×4 to 32×32 samples.
- New intra-prediction modes.
- Improved adaptive loop filters – three adaptive loop filters for reduction of noise in decoded video frames are integrated: deblocking filter [List03], Sample Adaptive Offset (SAO) [Fu11], adaptive loop filtering (ALF) based on a Wiener filtering approach [McC11]. As reported, all these approaches bring a significant amount of gain compared to the state-of-the-art.
- Improved interpolation filters.
- Adaptive selection of motion vectors resolution.
- Motion-compensated prediction with multiple reference pictures and variable block size and shape - each PU is predicted using one or two reference pictures enlisted in two reference picture lists (L0 and L1) and a combined list (LC) that contains pictures from both L0 and L1 lists. The motion information is signaled at the PU level and contains a reference picture index, a motion vector prediction index and a motion vector difference. Block sizes available for motion-compensated prediction are within the range of 64×64 to 4×4 luminance samples.

Similarly as in AVC standard, motion-compensated inter prediction is the main technique for temporal redundancy reduction in HEVC. Because a motion vector field resulting from blockwise motion estimation is highly redundant as the motion of adjacent blocks is very similar [Tok12], motion vectors of a current PU can be efficiently predicted from already encoded, surrounding blocks. Consequently, HEVC utilizes a combined set of different motion vector predictors that are enlisted in form of an ordered list [Bros11]. During the encoding process, the most efficient predictor in the created list is selected for each PU and signaled to the decoder by means of the *motion vector prediction index*.

In the current draft of HEVC [Wie11] five different types of motion vector predictors are considered (Fig. 2.4): *left*, *top*, *co-located* (block with the same spatial position, but located in a

different frame), *right-top corner* and *left-bottom corner*. These predictors are arranged in form of an ordered list (Fig. 2.5). Position on the candidate list is determined based on the likelihood of the candidate to be chosen for prediction. This provides a simple but efficient mechanism for manipulating cost of selecting each predictor from the list. The great advantage of such motion vector prediction strategy is the fact that prediction error of a motion vector can be reduced to small amplitude and thus compressed very efficiently. However, an additional information, i.e. the motion vector prediction index, must be transmitted to the decoder for each predicted motion vector.

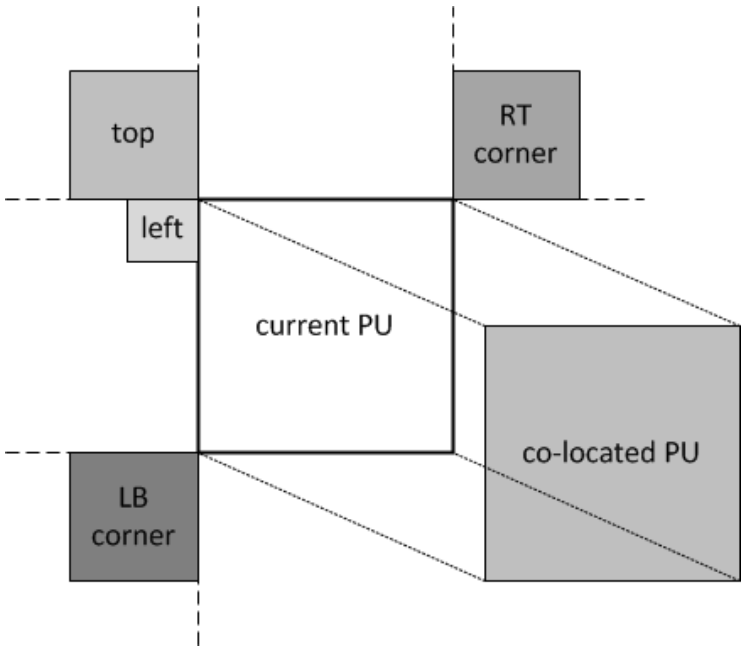


Fig. 2.4. Motion vector predictors in HEVC.

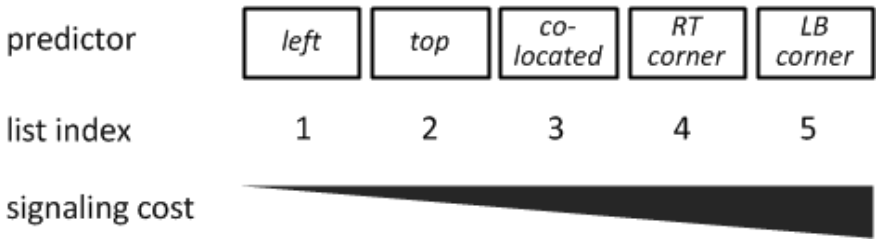


Fig. 2.5. Candidate list of motion vector predictors in HEVC (based on [Kon12]).

In addition, a new coding tool called *block merging* [Oud11, Win10, Marp10, Mat10] was introduced in HEVC, which is conceptually similar to the Direct mode of AVC. Block merging is designed to exploit the spatial redundancy of motion information in neighboring blocks belonging to potentially different branches and levels in the quadtree hierarchy. For this purpose, the merging algorithm uses the abovementioned list of motion vector predictors, which are called the *merge candidates*. Merge candidates are utilized to efficiently represent areas of homogeneous motion and

arbitrary shape with a single motion parameter. In case the block merging is used for a current PU, no other motion information except an index describing motion vector predictor need to transmitted for the block. As a result, block merging is a very efficient way of motion information coding.

The usage of block merging algorithm is signaled using dedicated syntax elements [Wie11]: *merge_flag* and *merge_index*. The binary valued *merge_flag* is signaled for each motion-compensated partition block prior to any prediction parameter. The *merge_flag* flag is transmitted only if the list of merge candidates is not empty, i.e. at least one of the neighboring blocks is predicted using motion compensation. In this case, if *merge_flag* is equal to 0, the current block is not merged with any of its neighboring candidate blocks and motion parameters for this block are signaled explicitly. Otherwise, one of the available merge candidates is selected as the motion information predictor for the current block and signaled using the *merge_index*. If the motion information of different merge candidates is identical, the list of candidates can be reduced. Consequently, the reduced list of candidates contains only motion information sets that differ from each other and its size is as small as possible. The position of selected candidate in the reduced list is identified by the *merge_index*, however, if the list is composed of only one candidate *merge_index* is not need to be transmitted.

Despite the similarity between block merging of HEVC and Direct mode of AVC, these two algorithms differ in the way they handle motion information from neighboring, previously encoded blocks. While the Direct mode infers motion parameters from adjacent blocks based on the median calculation, merging creates regions where all the blocks share the same motion information. The creation of these regions can be performed using only simple operations, such as comparing and copying the complete motion information from a neighboring block. In contrast, the calculations performed by median predictor in Direct mode require more computational complexity.

Considering the high effectiveness of the motion vector prediction scheme and block merging tool introduced in HEVC, there is still one basic assumption which must be fulfilled - the motion of neighboring blocks have to be very similar. This assumption works well for smooth translational motion, however, it fails when higher order motion as zoom or rotation appears in encoded video content.

2.2. Codec evaluation methods

2.2.1. Video quality assessment

Image and video quality assessment is an important issue in efficiency comparison of different coding algorithms. However, the complexity of the human visual system causes many difficulties in measuring the influence of distortion in visual content on the perceptual feelings of the viewer [Sul98, Dom10]. As a result, there are many methods for video quality assessment, among which two main classes can be distinguished:

- subjective quality evaluation [Wink05, ITUR03],
- objective quality evaluation [Oja03, Wink05].

The subjective quality evaluation of visual content is still considered as the most reliable assessment method, however, it must take into account many different aspects affecting the individual opinion of the viewer. Issues like individual interests, expectations and habits, congenital or acquired features of human visual system related to age and past illnesses are important for proper selection of the group of viewers. Obtained results depend also from presentation order of the evaluated content, lighting conditions, distance from the display and type of the equipment used during the assessment. The procedure requires also the involvement of a large group of viewers. Consequently, subjective quality evaluation is the most expensive, time consuming and laborious method of visual content assessment.

There are several recommendations describing the procedure of proper subjective quality evaluation [ITUR97, ITUR98, ITUR98a, ITUR03]. In particular, the recommendation of International Telecommunication Union [ITUR03] specifies two classes of subjective quality evaluation methods: double stimulus and single stimulus techniques. In Double Stimulus Continuous Quality Scale (DSCQS) method evaluated visual content is compared against the original one. On the other hand, in Single Stimulus Continuous Quality Evaluation (SSCQE) no original content is presented to the viewer. The quality scale used for the assessment is continuous, however, the variants of the abovementioned methods with discrete 5-grades scale are also used [ITUR03].

The most often utilized objective quality measure for the visual content quality assessment is peak signal-to-noise ratio (PSNR) [Wink05, Dom10]. The PSNR measure is defined as follows:

$$PSNR\ [\mathrm{dB}] = -10 \cdot \log_{10}\left(\frac{\sum_i {e_i}^2}{N \cdot (2^{N_b}-1)^2}\right) \tag{2.3}$$

where: N is the total number of samples in a picture, N_b is the number of bits representing the value of a sample (dynamic range) and e_i is the difference between corresponding pixels in original and distorted pictures.

The value of PSNR can be computed for each visual component representing the analyzed picture, however, it is often measured only for the luminance component, as the luminance distortions are the most visible [Sul98]. In such a case, the PSNR measure is usually represented by the abbreviation PSNRY - peak signal-to-noise ratio for luminance.

The PSNR measure usually gives quite good evaluation of the visual content quality, especially when analyzed distortions are of limited range and have the same character. Unfortunately, if the above conditions are not fulfilled, PSNR values may deviate significantly from the results of subjective quality assessment [Dom10].

Also, methods aimed at automatic assessment of video quality are intensively developed. These techniques are able to produce results highly correlated with subjective tests measurements, nevertheless, they still suffer from limited scope of applications [Wink05, Pas06, Dom10, Wink10, ANSI03, ITUR04, ITUT04]. Consequently, automatic methods are usually not suitable for the assessment of new coding algorithms.

Despite all the above mentioned methods differ substantially, selection of the method used for efficiency comparison of different codecs results from a compromise between measurement accuracy, time required to perform the assessment and its cost. The subjective quality evaluation is expensive and difficult to carry out, as it requires a number of observers, specialist equipment and a lot of tests. On the other hand, differences in quality of compared video are often slight. This requires high accuracy and a fine grain of the scale to properly assess the quality of the video sequence. Subjective tests are important for evaluating new technology and comparison of different transmission systems. On the other hand, PSNR objective measure usually provides fine visual content quality evaluation if distortions are of the same type and change to a limited extent [Dom10]. Because of the above reasons, in this dissertation, the objective measure PSNR has been chosen for the video quality evaluation.

2.2.2. Coding efficiency evaluation

Efficiency of a video coder is described by a rate-distortion curve (R-D curve), called also a R-D characteristics [Ort98, Ska98, Shi00, Dom10]. By selecting the values of control parameters, a specific operating point can be set for the coder. Such operating point is characterized by values of

bitrate R and distortion D. Consequently, the R-D curve is generated by plotting the distortion measure obtained by the analyzed coder against each tested bitrate (Fig. 2.6).

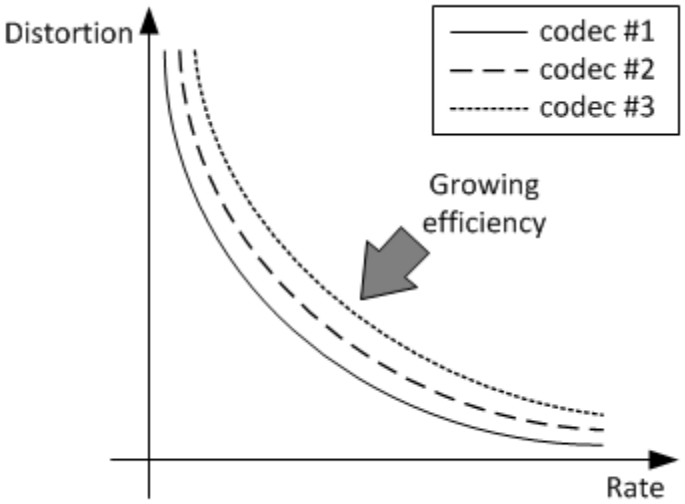


Fig. 2.6. Exemplary R-D characteristics.

In practice, R-D curves are usually presented as function of image quality measures against bitrate (see Fig. 2.7). As stated in Section 2.2.1, due to many difficulties in conducting the subjective quality assessment, the objective measure PSNR is the most frequently used for quality evaluation. In such a case, the value of PSNR is often measured only for the luminance component, as the chrominance distortions are less visible and annoying for the viewer [Sul98].

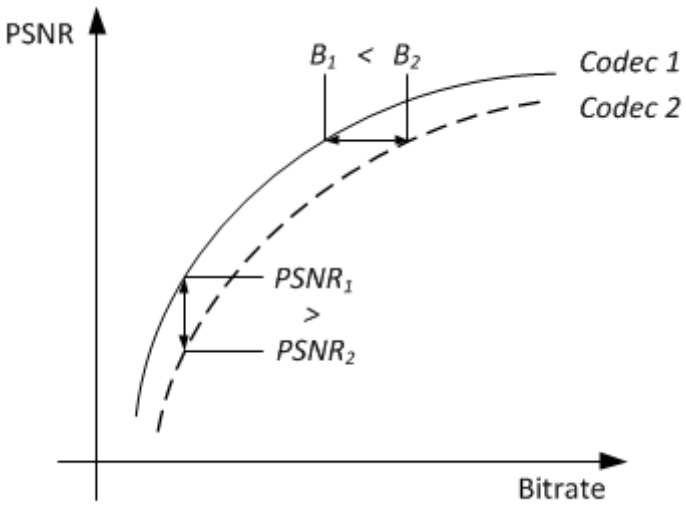


Fig. 2.7. Coding efficiency evaluation.

Coding efficiency comparison of two different codecs is conducted based on the R-D curves generated for each codec [Dom10]. Specifically, bitrates of the streams generated by each of the analyzed codecs for the same quality of decoded pictures are compared. Alternatively, the quality of decoded pictures achieved for the same bitrate can be considered. As shown in Fig. 2.7, *Codec 1* presents better coding efficiency than *Codec 2* as it achieves smaller bitrate at the same quality of decoded picture. Similarly, *Codec 1* achieves better quality of decoded picture for the same bitrate than *Codec 2*, which also testifies its better coding efficiency.

However, the abovementioned comparison can be made only for specified value of bitrate or quality. In practice, due to the fact that it is often difficult to obtain exactly the same bitrate or quality of decoded picture for all tested codecs, we usually compare the position of each of the analyzed R-D curves. In this case, better coding efficiency is related to the R-D curve located higher on the plot (see Fig. 2.7). Unfortunately, the problem of this approach reveals when intersecting R-D curves are analyzed. As a result, a dedicated metric for more systematic comparison of coding efficiency was introduced. This metric, called the *Bjontegaard metric* (BJM) [Bjo01], will be further discussed in this section.

The idea of complex comparison of coding efficiency introduced in Bjontegaard metrics is based on the interpolation of R-D curves calculated from the measured operating points of analyzed codecs. For practical reasons connected with the time and complexity of calculating Bjontegaard metric, the number of required data points used for interpolating each of R-D curve was limited to 4 (see Fig. 2.8). Consequently, the interpolated R-D curve for each codec is determined by a third order polynomial.

On the basis of interpolated R-D curves, an average bitrate difference between two analyzed codecs is calculated for a considered quality (usually PSNR) range. Alternatively, differences in quality are averaged among a considered bitrate range. As a result, there are two Bjontegaard metrics presenting average bitrate and quality differences, represented by ΔB [kbps] and $\Delta PSNR$ [dB] respectively. Both quality and bitrate ranges used for averaging procedure are determined by outermost operating points measured for each codec (see Fig. 2.8).

Today, Bjontegaard metrics are widely used for comparing the coding efficiency of the codecs, especially by VCEG and MPEG. Detailed description of the procedure for calculating Bjontegaard metrics can be found in [Bjo01, Pat07].

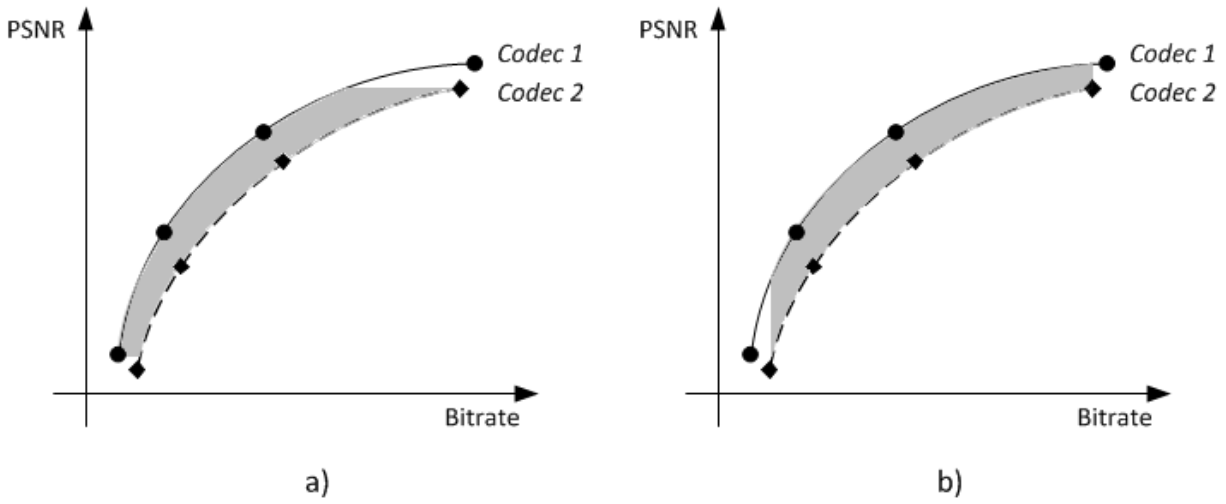


Fig. 2.8. R-D curves interpolation. Averaged values range for: a) ΔB, b) $\Delta PSNR$ Bjontegaard metric evaluation.

2.2.3. Complexity analysis (coding and decoding time measure)

Today, there are many methods to assess the computational complexity of the software. There are several informatics tools, such as VTune, which can be easily adopted for measuring encoding and decoding time of a video codec. However, all these methods suffer from inaccuracies caused by the CPU load due to other tasks, which fluctuates in time, and changing hard drive access time. This results in no repeatability of the experimental results and may lead to significant measurement errors. On the other hand, single encoding of a multiview sequence performed with a modern video codec may take tens of hours. This definitely limits the possibility of repeating experimental tests when employing statistical analysis in order to achieve narrower confidence intervals.

As a consequence, JCT-VC adopted a simple and rough video codec computational complexity analysis method which consists in measuring a single runtime of encoder or decoder using the same machine for each tested codec [Sul10, Sue10]. Results obtained this way are not free from the abovementioned disadvantages, however, can be utilized for a rough computational complexity assessment, which is usually sufficient at the stage of development of new video coding techniques. Moreover, measurements can be repeated on different platforms, which is also a workaround for the problem of code optimization. In case of usage of specialized instruction sets in the code, results obtained for various platforms may differ significantly. By comparing the results from different platforms, the influence of code optimization on the final assessment of the analyzed codec can be reduced.

2.3. Accuracy measures of motion field prediction

In this section we present methods for evaluating the accuracy of predicted motion vectors. The methods are Pearson correlation coefficient (PCC) and Vector Similarity Measure (VSIM) which are commonly used correlation measures [Ryu11]. Both methods indicate how close the calculated values are to the maximum accuracy which is an obvious requirement for evaluating motion vector prediction accuracy.

2.3.1. Pearson correlation coefficient (PCC)

Pearson correlation coefficient (PCC) is a widely used measure of correlation between sets of scalar variable. For two scalar variable sets *X* and *Y*, each containing of *n* samples, the sample PCC, indicated as *r*, can be defined as [Pre07]:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2.4}$$

Based on a sample of paired data (X_i, Y_i) from the analyzed data sets *X* and *Y* an equivalent expression defining the sample PCC as the mean of the products of the standard scores is:

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}}{s_X}\right)\left(\frac{Y_i - \bar{Y}}{s_Y}\right) \tag{2.5}$$

where $\frac{X_i - \bar{X}}{s_X}$ is the standard score, $\bar{X}$ is sample mean and s_X is sample standard deviation respectively.

The absolute values of PCC correlation are less than or equal to 1. Correlation value of 1 indicates that a perfect linear equation describes the relationship between *X* and *Y*.

Application of PCC correlation measure as an accuracy or similarity of motion fields measure is implemented separately for horizontal and vertical components of motion vectors [Ryu11]. Consequently, PCCx(*X*,*Y*), PCCy(*X*,*Y*) and PCCavg(*X*,*Y*) measures are calculated for data sets *X* and *Y*, containing motion vectors from compared motion fields. PCCx(*X*,*Y*) indicates PCC calculated for horizontal component of motion vectors from data sets *X* and *Y*, PCCy(*X*,*Y*) is PCC for vertical component of motion vectors, and PCCavg(*X*,*Y*) is an averaged value of the two abovementioned coefficients.

2.3.2. Vector Similarity Measure (VSIM)

Vector Similarity Measure (VSIM) [Ryu11] is a measure for evaluating the similarity of two vector quantities, with values in a range of {-∞, 1}. The value of VSIM closer to 1 indicates that two compared vectors are more similar. The VSIM combines two similarity types: the angular similarity $ASIM(\vec{X},\vec{Y})$ and the magnitude similarity $MSIM(\vec{X},\vec{Y})$, where, in general, $\vec{X}$ and $\vec{Y}$ are two compared vectors.

The VSIM value between two vectors $\vec{X},\vec{Y}$ is defined as follows:

$$VSIM(\vec{X},\vec{Y}) = \begin{cases} ASIM(\vec{X},\vec{Y}) \times MSIM(\vec{X},\vec{Y}) & if\ |\vec{X}| \neq |\vec{Y}|, ASIM(\vec{X},\vec{Y}) > 0 \\ ASIM(\vec{X},\vec{Y}) \times MSIM(\vec{X},\vec{Y})^{-1} & if\ |\vec{X}| \neq |\vec{Y}|, ASIM(\vec{X},\vec{Y}) < 0 \\ 1 & if\ |\vec{X}| = |\vec{Y}| = 0 \\ 0 & if\ |\vec{X}| = 0\ or\ |\vec{Y}| = 0 \end{cases} \quad (2.6)$$

$$ASIM(\vec{X},\vec{Y}) = \frac{\vec{X}\cdot\vec{Y}}{|\vec{X}||\vec{Y}|} = \cos\theta \quad (2.7)$$

$$MSIM(\vec{X},\vec{Y}) = \frac{\min(|\vec{X}|,|\vec{Y}|)}{\max(|\vec{X}|,|\vec{Y}|)} \quad (2.8)$$

where θ is the angle between $\vec{X},\vec{Y}$, and $|\cdot|$ represents the magnitude of a vector.

The angular similarity of VSIM describes the relative magnitude of the two vectors. If the angle between $\vec{X},\vec{Y}$ decreases the value of VSIM increases. Similarly, the value of VSIM increases when the magnitude ratio of $\vec{X},\vec{Y}$ approaches 1.

The VSIM measure may be used to compare two motion fields. In such application, two data sets: *X* and *Y* are created, each containing motion vectors from different motion field. Next, for each element of data set *X* a VSIM similarity value is calculated with the corresponding element from data set *Y*. Finally, VSIM values calculated for each corresponding motion vector pairs from *X* and *Y* data sets are averaged creating the mean VSIM(*X*,*Y*) value used for evaluation of two motion fields similarity [Ryu11].

2.4. Multiview video coding techniques

2.4.1. Multiview Video Coding (MVC)

The Multiview Video Coding (MVC) is a video coding standard based on the AVC technology, established by the MPEG committee as a new ITU-T and ISO/IEC standard in 2009 [ISO11, Vet11, Chen09]. The MVC codec was designed to utilize the existing correlation between neighboring views in multiview video sequences in order to increase the compression ratio using the inter-frame prediction [Feck05, Kaup06]. The principal reason for efficiency of such approach results from the fact that camera positions in recent multiview acquisition systems are very close. Consequently, view-fields of the cameras overlap in large part producing highly correlated video content.

The basic approach to multiview video coding introduced in MVC is an inter-view prediction with multiview disparity compensation. This inter-frame prediction uses a mechanism similar to motion compensation of the AVC codec, however, with reference frames from neighboring views. As a result, original reference lists of the AVC used for motion compensation are extended with additional frames from the same time instance, but different views. For example (see Fig. 2.9), a frame from view c_j and time instance t_i can be predicted using not only reconstructed frames from time instances t_{i-1} and t_{i+1} of the same view, but also reference frames from neighboring views c_{j-1} and c_{j+1} with the same time index t_i.

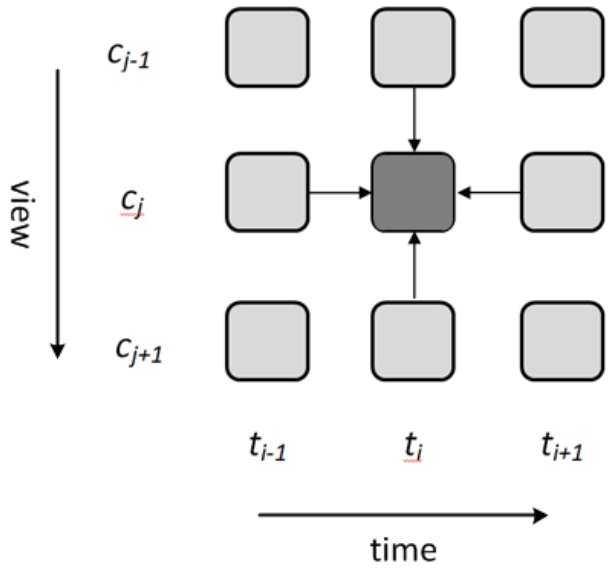


Fig. 2.9. Exemplary inter-view prediction in MVC.

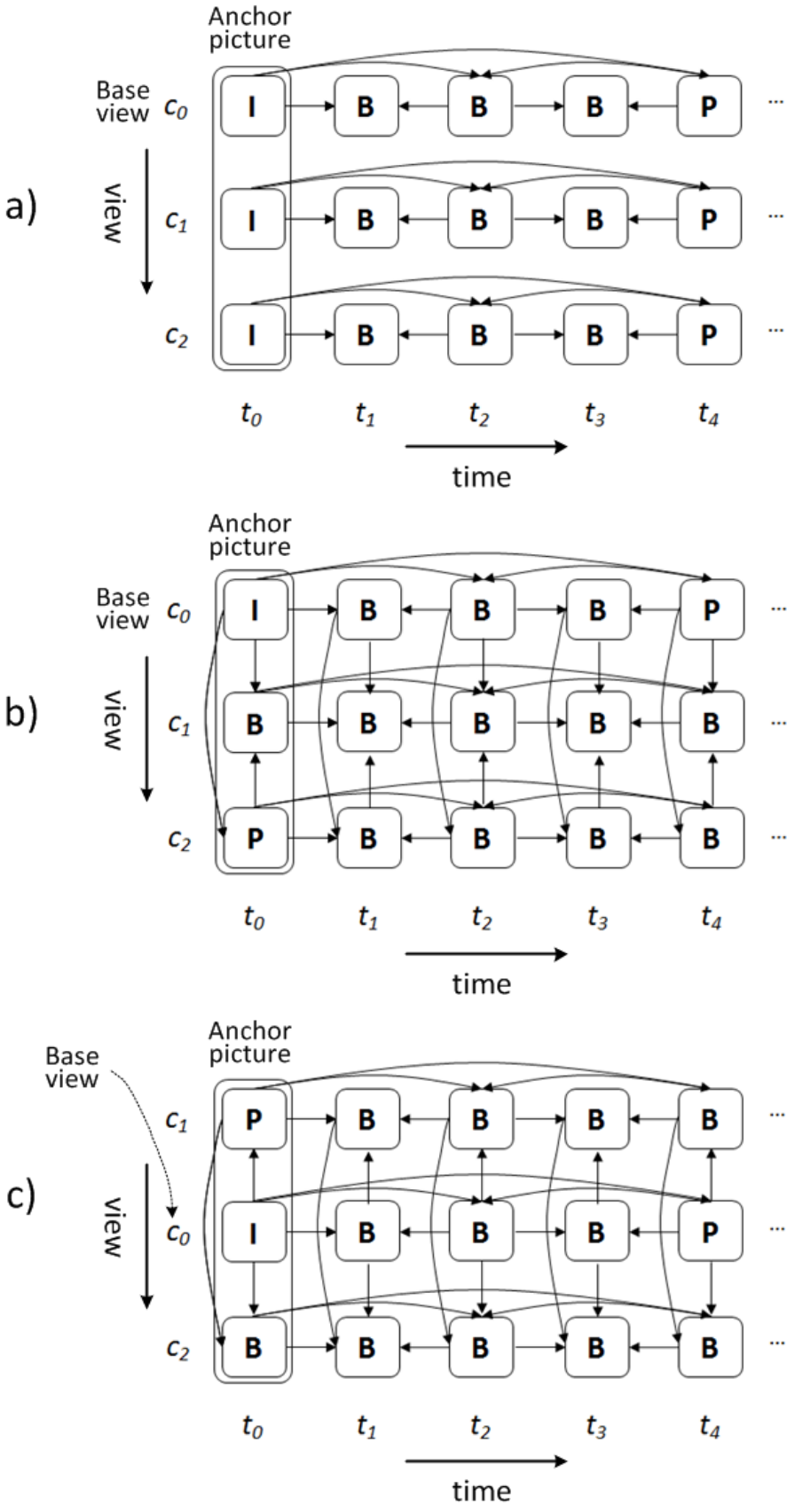


Fig. 2.10. Multiview Video Coding prediction schemes: a) with independent views (simulcast), b) with outermost base view, c) with central base view.

The abovementioned mechanism imposes a defined view coding order which must be uniform at the encoder and decoder side. Exemplary schemes of encoding using different view coding orders are presented in Fig. 2.10. The view encoded as first according to the view coding order is called a *base view*. All other views are referred to as *side views*. Each frame which starts the group of pictures (GOP) in every view of the multiview sequence is described as *anchor picture*. One of the most important consequences which results from application of inter-view prediction schemes in MVC is substitution of the anchor I frames in the non-base views with the P and B frames. This obviously reduces the overall compression performance of the codec as the P and B frames represent the video content much more efficiently than I frames. Other prediction schemes discussed during standardization process of the MVC can be found in [Feck05, Kaup06, Merk07, Flier07, Huo11].

In the design process of the MVC, a reference codec model called the Joint Multiview Model (JMVM) was formed and intensively developed. During this process, a number of coding tools were explored, including especially the following.

- Illumination compensation – the purpose of this tool is to compensate for illumination differences between the views as part of the inter-view prediction process [Lee06, Hur07].
- Adaptive reference filtering – a scheme of adaptive reference filtering to compensate for focus mismatches between different views of a multiview sequence [Lai07, Lai08].
- Motion skip mode – method of inferring motion vectors from inter-view reference pictures in order to utilize the existing correlation between motion fields of different views [Koo06, Koo07, Koo08]. This method will be further discussed in Section 2.5.
- View synthesis prediction – in this method, a prediction of a picture in the current view is made based on synthesized references generated from neighboring views [Mart06, Kit06, Yea09].

It was shown that additional coding gains can be achieved by utilization of these coding tools. However, due to required low-level design and syntax changes affecting the coding process, the abovementioned tools were not included in the MVC standard. There was also some concern that achieved coding gains might be reduced by growing quality of video acquisition and improved preprocessing algorithms [Vet11]. As a result, a revised model for MVC codec was formed – the JMVC, which includes all functionalities of the final MVC release.

MVC standard [ISO11] defines two main profiles for multiview video coding: *stereo high profile* and *multiview high profile*. First of these profiles is designed for stereoscopic video coding, including interleaved stereoscopic video. The second one defines coding of multiview video containing more than two views and applies only to progressive video coding. Stereo high profile

has been included in the Blue-ray standards and is commonly used for recording stereoscopic movies on Blu-ray discs. Additionally, the MVC standard specifies a number of dedicated Supplemental Enhancement Information (SEI) messages for transmitting the maximum value of disparity between encoded views and parameters of the camera system used for acquisition.

2.4.2. Multiview High Efficiency Video Coding (MV-HEVC)

Multiview High Efficiency Video Coding (MV-HEVC) is an implementation of multiview video coding in the framework of emerging HEVC technology [Dom11, Sta12]. Similarly as in the MVC approach, MV-HEVC uses inter-view prediction in addition to the classic inter-frame prediction to encode multiview video more efficiently.

The inter-view prediction of MV-HEVC is applied by modifying the reference picture lists of side views. Pictures from the same time instant but other views may be also selected as the reference pictures by the codec. Consequently, MV-HEVC implements compression scheme similar to MVC technology, however, the AVC core of the codec has been substituted by the emerging HEVC coder.

As the structure of the HEVC codec is quite similar to that of AVC, the adaptation of multiview coding tools of MVC is almost straightforward. Nevertheless, due to existence of new coding tools introduced in HEVC codec, the following modifications for improved inter-view prediction were proposed in MV-HEVC [Sta12].

- Inter-view motion vector scaling – additional inter-view scaling of motion vector used by motion vector predictors of HEVC in case the reference pictures form other views are utilized for prediction.
- Inter-view support for co-located prediction – one of four possible co-located motion vector predictors is selected as the candidate predictor depending on the location of the co-located unit and the reference frame that is being used.
- Nested prediction – a new coding tool for predicting motion information when no motion vector prediction candidates can be derived from the neighboring blocks due to temporal/inter-view reference picture mismatch.

Experimental results on compression performance of MV-HEVC indicate a significant bitrate reduction of 50 [%] when compared to the state-of-the-art MVC technology and a 20-30 [%] bitrate reduction when compared to the simulcast HEVC [Dom11, Sta12].

2.5. Inter-view prediction of motion information

The fundamental idea utilized in multiview video coding is based on the question of how to efficiently exploit the existing correlation between adjacent views. As reported, in case the distance between neighboring viewpoints is small, there is a high correlation between content of the video sequences obtained from these viewpoints [Feck05, Merk07, Su06]. Consequently, one of the concepts for improving the compression efficiency of the multiview video is based on the idea of inter-view motion prediction. In particular, since the motion of neighboring views is highly correlated, the motion information of encoded view, including motion vectors and reference picture indices, can be derived from the neighboring view at the same time instant.

This concept was investigated in [Guo06], where the authors point out that, besides traditional skip and direct modes known well from AVC standard, there are no more efficient ways to predict motion vectors in MVC. As presented in the article, usage of global geometric model representing differences in location and angle between cameras is sufficient to derive motion vectors from neighboring view in case of simple local motion. In the proposed approach, authors employ a six-parameter affine transformation to determine the global disparity representing the displacement between neighboring views.

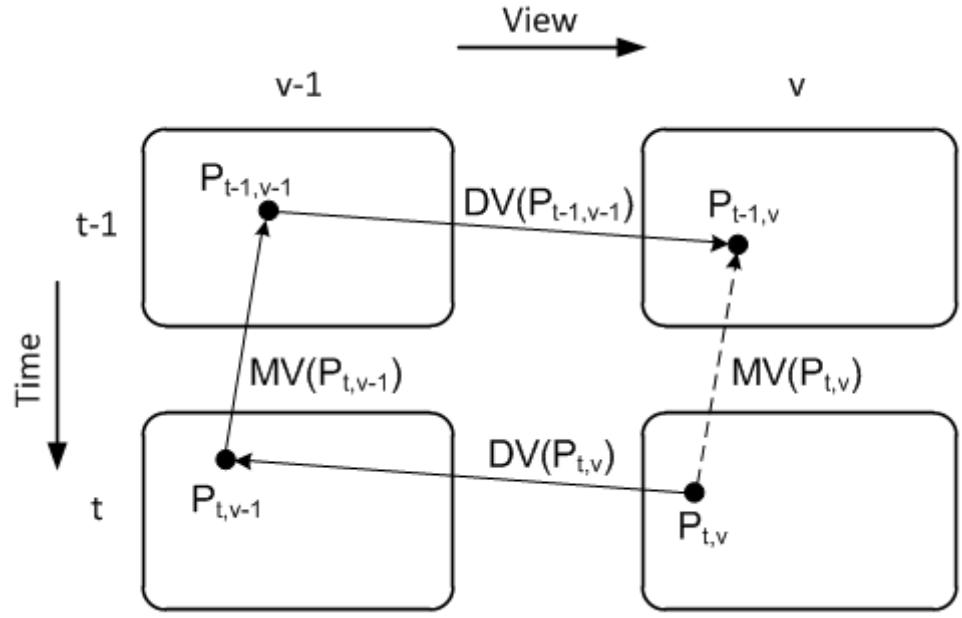


Fig. 2.11. Geometric model of motion and disparity vectors correspondence (based on [Guo06]).

As shown in Fig. 2.11, the inter-view motion correlation can be deduced based on the mapping along both temporal and view directions. Using the disparity vector (DV), a motion vector (MV) can be identified for each point $P_{t,v}$ of encoded image. This inter-view correspondence of motion fields was utilized in a macroblock prediction mode named *inter-view direct mode* [Guo06]. In this mode (see Fig. 2.11), for each macroblock of encoded view, a motion vector $MV(P_{t,v})$ is determined based on the known global disparity vectors $DV(P_{t,v})$ and $DV(P_{t-1,v-1})$ and a corresponding motion vector in the neighboring view $MV(P_{t,v-1})$. Moreover, coding of prediction directions for derived motion vectors into bitstream is not necessary for the inter-view direct mode as these information can be also derived from the neighboring view. Results presented by the authors show that utilization of the proposed motion model is accurate and causes only minor increase of the coder and decoder complexity. However, large prediction errors are produced in case of complicated motion in the encoded scene and in the regions where global disparity vector differs from local disparity.

The inter-view motion information prediction had also called the interest of Joint Video Team (JVT). As a result, in addition to the final draft of MVC, JVT had also maintained a Joint Multiview Model (JMVM) for MVC [MP08] in which an additional coding tool called the *Motion Skip* (MS) was included. The Motion Skip is a coding tool which enables reusing motion information from other views employing a given disparity for each macroblock that utilizes MS [Yang09].

The concept of MS was first introduced in [Koo06] and is motivated by the idea that a similarity in motion fields between neighboring views exists. The idea of MS is similar to inter-layer motion prediction introduced in Scalable Video Coding (SVC) [ISO11, Lang04], however, does not enable motion refinement by coding differential motion vectors. In the proposed MS algorithm [Koo06], for each encoded macroblock, a global disparity of 16-pixel accuracy, signaled in the slice header, is used to determine the corresponding macroblock in the neighboring reference view. The usage of MS is indicated by an additional flag called *motion_skip_flag* that is included in the head of macroblock layer syntax of side views [Chen07]. If the *motion_skip_flag* is set to 1, macroblock mode, motion vectors and reference picture indices are derived from the corresponding macroblock in the reference view and reused for the motion-compensated inter prediction of the current macroblock. Consequently, no further macroblock information have to be transmitted for the current macroblock, which obviously increases the compression efficiency of the multiview codec.

Although the initial version of MS algorithm proposed in [Koo06] improved the coding efficiency of MVC and had been adopted into early version of JMVM [Vet07a], some modification

proposals appeared to further increase its performance. One of the most important modifications resulted from the observation that utilization of the global disparity vector does not provide the sufficient accuracy for determining the position of corresponding macroblock in the neighboring reference view because of the depth differences between objects in encoded visual scene [Koo07]. As a result, a region-based estimation and transmission of disparity corrections were proposed [Song07, Lin07, Yang07]. For each macroblock in MS mode, a local disparity vector is searched at the encoder within a given search area (see Fig. 2.12). The local disparity vector defines an offset to the global disparity vector and is selected from all available candidates defined by the search window based on the rate-distortion performance criterion. Obviously, the offset needs to be transmitted in head of the macroblock layer as an additional side information.

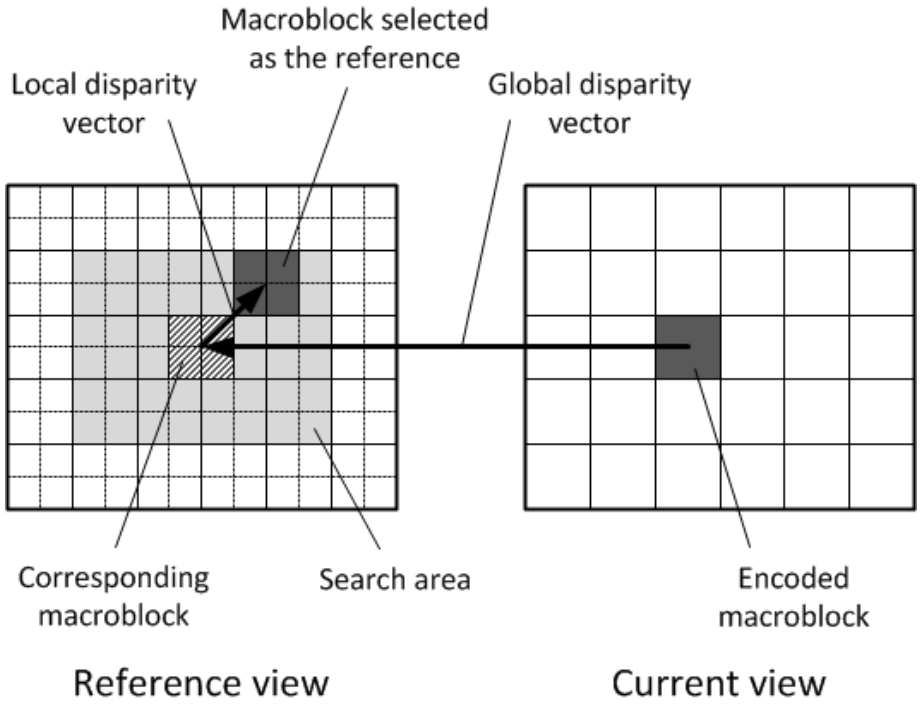


Fig. 2.12. Inter-view motion prediction in Motion Skip.

The second significant change was increasing the disparity vector accuracy to 8-pixel and, as a consequence, the accuracy of the position from which the motion information is derived. This concept was proposed in [Yang07, Yang08, Yang09] as the *fine-granular motion matching algorithm* for the Motion Skip mode. The main motivation originated from the observation that the basic unit to perform motion compensation in AVC standard is a 8×8 block. Therefore it would be reasonable to use the motion information of the 8×8 block as the basic unit to derive motion information for the current macroblock. Consequently, the current macroblock uses a disparity that points to four 8×8 blocks within the corresponding area in the neighboring reference view (Fig.

2.12). Next, motion information of these four 8×8 blocks is reused, resulting in more accurate motion-compensated prediction of the current macroblock.

As a consequence of the above mentioned modifications, the MS tool was updated resulting in a new JMVM release [MP08]. Experiments have shown that Motion Skip together with another coding tool called Illumination Compensation [Lee06, Hur07] present a good coding performance in comparison to simulcast AVC, providing approximately 26 [%] of average bitrate saving [Jeon08]. However, when compared to MVC codec, the coding efficiency increase of approximately 4 [%] provided by the MS was concluded as relatively low [Chen09a], especially when compared with the expectations set in the future video coding standard for multiview video.

Similarly to the solution called inter-view direct mode which was described earlier, the complexity increase due to MS is minor at the decoder, however, it is substantial at the encoder because of a local disparity search performed in order to increase the accuracy of the inter-view prediction provided by the global disparity vector.

Following the development of MS algorithm, it can be clearly observed that most of the attempts aimed at improving the performance of MS addressed the issue of increasing the accuracy of the method to determine the inter-view correspondence region for a macroblock. The main reason is that, in some cases, the block-wise global and local disparity vectors are not sufficient for this purpose.

Besides the pure compression domain, the concept of deriving the motion information on the basis of inter-view reference is also present in issues related to reducing computation complexity of the motion estimation process. In this application, the accuracy of derived information is crucial in terms of reducing the number of motion estimation search steps, as it is used to determine initial search point and the macroblock partition scheme. Although the final goal is different from the case of coding with inter-view motion information derivation tool, both issues are based on the assumption that the motion information in neighboring views is highly correlated and can be reused in encoded view. As a consequence, the concepts invented in this issue should also be mentioned to present a wide range of techniques and conceptions utilized to reuse motion information in multiview video.

In [Ding08a] authors present a complete system for low-computation multiview coding. The concept introduced in the article uses the motion information derived from the reference view to reduce the number of motion estimation steps for encoded macroblock and predict the macroblock partition mode. The position of the encoded macroblock in the reference view is determined with a simple block disparity estimation which chooses best matching position according to Sum of

Absolute Differences (SAD) measure. Next, the encoded macroblock is split into sixteen 4×4 blocks and the motion information from the reference view is derived to each 4×4-pel blocks independently in order to preserve the original AVC macroblock partitioning (Fig. 2.13). In case the 4×4 area which is covered by the block contains more than one different motion vector, the vector with the largest overlapped area is assigned. Then a partition labeling begins and the most representative motion vector is chosen as an initial guess for motion estimation. Finally, the refinement within a small search range is performed around initial guess and the motion vector for encoded macroblock is determined. Important assumption of this method is that there are no early termination and fast prediction schemes applied in the reference view. This means all kinds of cost, including especially the best inter prediction mode and its motion vectors, must be calculated during the reference view coding process and remain available for the encoded view. This prevents the prediction error propagation.

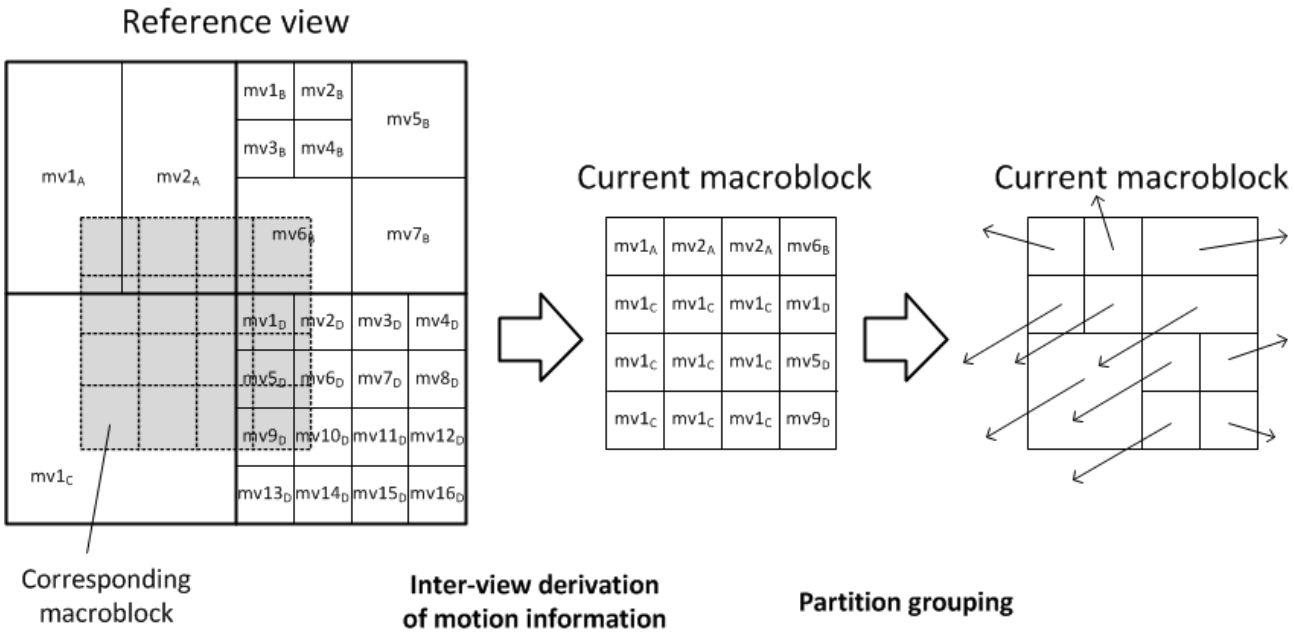


Fig. 2.13. Derivation of motion information with partition grouping for fast motion estimation (based on [Ding08]).

Similar concept is used in [Ding08] and [Ding07] where the inter-view motion information derivation is utilized to perform fast or computation-free motion estimation respectively. As the authors point out, the computation complexity reduction equals 95-100 [%] with only small quality degradation. Nevertheless, both methods present early termination limitations mentioned above and employ additional refinement of the motion information derived from the neighboring view.

On the above it can be concluded, that the application of the global disparity model or simple block disparity estimation as the determinant of correspondence between different views does not allow to fully utilize the existing inter-view redundancy. The most important reason is that such a basic approach does not include many complex geometry dependencies between objects in the scene. Utilization of the block approach, even with the block size of 4×4 pixels, still does not preserve object borders. Moreover, the simple, translational motion model with motion vectors assigned on a block level limits the correlation of motion fields between the views. As a result, a field for further improvement in multiview video compression exists.

2.6. Perspective projection and multiview depth-based rendering

2.6.1. Pinhole camera model

Every image acquisition system, including human visual system or visual systems used in machine vision applications, utilizes a transformation of a point in 3D space into native coordinate space, e.g. 2D image plane. In case of camera models, the most widely used approaches are *parallel* and *perspective* projections which define the geometric relationship between a 3D point and its 2D corresponding projection onto the image plane [Cyg02, Jav06, Cyg09, Mend09].

Parallel transform assumes parallel projection of 3D objects onto 2D image plane, which is a simplified approximation of a transform conducted by the camera. As a result, the main advantages of this approach are linearity and small complexity. However, due to very limited accuracy, the method is usually used when the distance between the camera and recorded visual scene is small or in case of applications for which accuracy aspect is not crucial [Cyg02].

The second projection method, called the *perspective* projection, performs much better, offering accuracy adequate for stereovision or motion estimation applications [Adam01, Davi97, Deve95, Faug93, Heik00, Youn94]. Consequently, this is the camera model which is regularly employed as a basis in this thesis. The pinhole camera model with perspective projection, considering world and camera coordinate systems, is presented in Fig. 2.14.

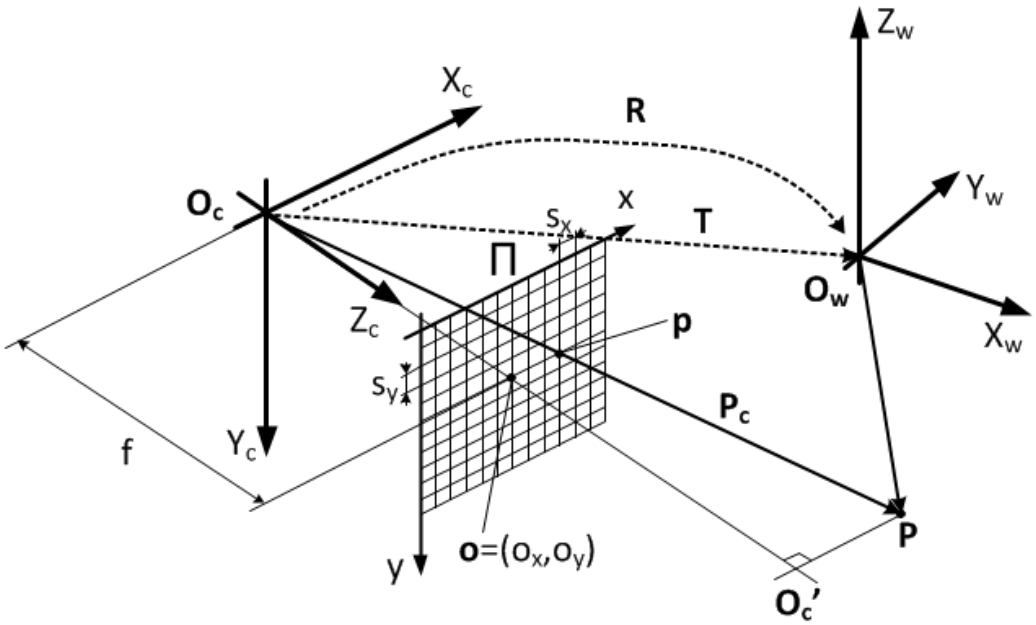


Fig. 2.14. Pinhole camera model (based on [Cyg02]).

When using a pinhole camera model [Cyg02], we denote the center of the perspective projection as the *optical center* $\boldsymbol{O_c}$, the point in which all the rays intersect (see Fig. 2.14). The line perpendicular to the image plane Π passing through the optical center is the *optical axis* (line $\overline{\boldsymbol{O_c O_c'}}$). Additionally, the intersection point of the image plane Π with the optical axis is called the *principal point* $\boldsymbol{o}$ with pixel coordinates (o_x, o_y). The distance between the image plane Π and the optical center $\boldsymbol{O_c}$ determines the *focal length* f. s_x and s_y are the physical diameters of a pixel on the image plane Π in horizontal and vertical direction respectively.

Point $\boldsymbol{O_c}$, together with X_c, Y_c, Z_c axes, create the coordinate system associated with the camera. On the other hand, X_w, Y_w, Z_w axes create the world coordinate system with the center point $\boldsymbol{O_w}$.

The location of a point $\boldsymbol{P}$ in 3D space is described by a vector $\boldsymbol{P_c}$ in the camera coordinate system and a vector $\boldsymbol{P_w}$ in the world coordinate system. The perspective projection of the point $\boldsymbol{P}$ onto image plane Π is point $\boldsymbol{p}$. Point $\boldsymbol{P}$ and its projection $\boldsymbol{p}$ have the following coordinates in the coordinate system associated with the camera:

$$\boldsymbol{P} = (X, Y, Z) \tag{2.9}$$

$$\boldsymbol{p} = (x, y, z) \tag{2.10}$$

Now, let us consider the similarity of triangles $\Delta \boldsymbol{O_c p o}$ and $\Delta \boldsymbol{O_c P O_c'}$, and also assume $z = f$. As the optical axis is perpendicular to the image plane we can write the following equations, which constitute the pinhole camera model:

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}, \qquad z = f \tag{2.11}$$

Mathematical generalization of the pinhole camera model is known as affine camera model which is well described in [Truc98, Moh96, Mun92, Faug93]. There are also more accurate camera models that consider modeling of the real camera lenses, used to solve more advanced and complex problems of computer graphics. These methods are well described in [Kol95, Shan97]. However, for the purpose of applications discussed in this thesis (e.g. stereovision) the pinhole camera model is assumed to be sufficient and commonly used [Cyg02].

The pinhole camera model is further specified by the set of *intrinsic* and *extrinsic camera parameters* which are discussed in the following section.

2.6.2. Camera parameters

2.6.2.1. Intrinsic camera parameters

The *intrinsic camera parameters* are related to: focal length, principal point offset, image sensor characteristics and geometric distortion from optical circuit of the camera [Cyg02, Jav06, Cyg09].

In pinhole camera model, the only parameter required to define perspective projection is the focal length f of the camera that determines distance between the optical center and the image plane (see Eq. 2.11). The intrinsic camera parameters set may be represented using matrix notation by a *projection matrix* $\boldsymbol{K}$:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

The origin of the pixel coordinate system in most of the current imaging systems is defined at the top-left pixel of the image. However, as it was previously assumed, the origin of the pixel coordinate system corresponds to the principal point $\boldsymbol{o} = (o_x, o_y)$, located at the center of the image (see Fig. 2.14). Consequently, a conversion of coordinate systems is necessary:

$$x = s_x(x_u - o_x) \tag{2.13}$$

$$y = s_y(y_u - o_y) \tag{2.14}$$

where x and y determine the position of a 3D world point $\boldsymbol{P}$ projected onto the image plane Π (see Eq. 2.11), x_u and y_u are pixel positions in the image plane Π, s_x and s_y are the physical diameters

of a pixel on the image plane Π in horizontal and vertical direction respectively. As a result, the principal point position can be readily integrated into the projection matrix $\boldsymbol{K}$:

$$\boldsymbol{K} = \begin{bmatrix} \frac{f}{s_x} & 0 & o_x \\ 0 & \frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

In Eq. 2.11 it was implicitly assumed that the pixels of the image sensor are square (i.e. their aspect ratio is 1:1) and not skewed. However, in physical camera systems both assumptions may not always be true. In case of image acquisition by means of frame grabber pixels can potentially be skewed. Also, pixels may be non-square. In that case the pixel aspect ratio is usually provided by the sensor manufacturer (e.g. 10:11 in NTSC TV systems). These imperfections of the camera system can be successfully addressed by introducing parameters τ and η into camera model. The parameter τ models skew of the pixels, while the parameter η models pixel aspect ratio (see Fig. 2.15). Consequently, projection matrix $\boldsymbol{K}$ can be updated as:

$$\boldsymbol{K} = \begin{bmatrix} \frac{f}{s_x} & \tau & o_x \\ 0 & \eta\frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

However, in practice, when employing recent digital cameras, it can be safely assumed that pixels are square ($\eta = 1$) and non-skewed ($\tau = 0$).

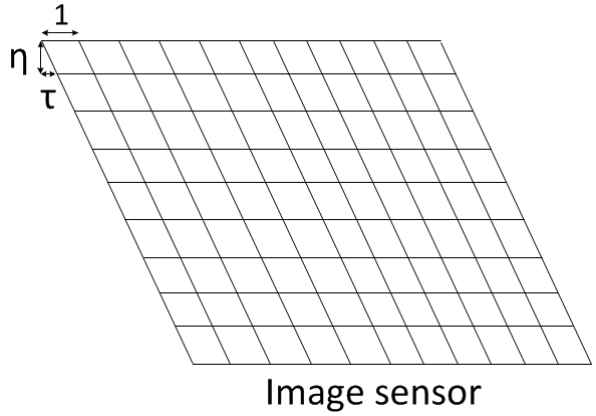


Fig. 2.15. Image sensor characteristics distortions.

Real camera lenses usually suffer from non-linear lens distortions and dependencies from light wavelength [Hech98, Shan97, Pedr98]. In case of lens distortions, spherical aberration, coma,

astigmatism, curvature of field of view, as well as barrel distortion should be mentioned. In the latter case, the most common representative is chromatic aberration. In practice, the abovementioned distortions are usually modeled by means of a simple radial distortion used to determine the dependency between distorted and undistorted pixel positions [Cyg02]. The strength of radial distortion grows with the distance from the principal point $\boldsymbol{o} = (o_x, o_y)$:

$$x_d - o_x = \frac{x_u - o_x}{1 + k_1 r^2 + k_2 r^4} \tag{2.17}$$

$$y_d - o_y = \frac{y_u - o_y}{1 + k_1 r^2 + k_2 r^4} \tag{2.18}$$

$$r = (x_d - o_x)^2 + \left(y_d - o_y\right)^2 \tag{2.19}$$

where k_1 and k_2 are intrinsic camera parameters defining the radial distortion of the lens, x_u and y_u are undistorted pixel positions in the image plane Π, x_d and y_d are distorted pixel positions in the image plane Π. It is often assumed that $k_2 = 0$, as this does not introduce a noticeable degradation of the quality of the model [Truc98].

In order to estimate the distortion parameters a minimization of a cost function that measures the curvature of lines in the distorted image is performed [Thor05, Klet98]. One of the possible approaches to measure the curvature is to detect feature points belonging to the same line on a calibration rig, e.g. a checkerboard calibration pattern. In the distorted image, each point belonging to the same line forms a bended line. Consequently, the distortion parameters can be calculated by analyzing the deviation of the bended line from the theoretical straight line model [Thor05].

2.6.2.2. Extrinsic camera parameters

The *extrinsic camera parameters* indicate the external position and orientation of the camera in the 3D world [Fol94, Truc98]. Mathematically, the transition from the camera coordinate system to the world coordinate system is defined by a 3 × 1 translation vector $\boldsymbol{T}$ and by a 3 × 3 rotation matrix $\boldsymbol{R}$ (see Fig. 2.14). Translation $\boldsymbol{T}$ vector defines displacement between centers of coordinate systems: $\boldsymbol{O_c}$ and $\boldsymbol{O_w}$. Rotation, defined by a orthogonal matrix $\boldsymbol{R}$, transforms corresponding axes of both coordinate systems.

Relation between coordinates of point $\boldsymbol{P}$ in the camera and word coordinate systems, using notation introduced in Section 2.6.1, is defined by the following equation:

$$\boldsymbol{P_c} = \boldsymbol{R}(\boldsymbol{P_w} - \boldsymbol{T}) \tag{2.20}$$

where $\boldsymbol{P_c}$ is position of point $\boldsymbol{P}$ in the camera coordinate system, $\boldsymbol{P_w}$ is position of point $\boldsymbol{P}$ in the world coordinate system, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation matrix between both coordinate systems. Additionally, we can define:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \qquad \boldsymbol{T} = \boldsymbol{O_w} - \boldsymbol{O_c} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.21}$$

Extrinsic camera parameters of the perspective camera model are defined as a set of geometric parameters that explicitly determine transformation from the camera coordinate system into extrinsic coordinates.

2.6.3. Projective geometry and perspective projection using homogeneous coordinates

Projective geometry provides a very useful framework for analysis and description for machine vision and 3D multiview imaging in computer graphics [Cyg02, Sem98, Hart00, Faug01, Mun92, Moh96]. The most important reason for adopting the projective geometry to abovementioned applications instead of Euclidean geometry is the ability to describe easier the geometric objects (e.g. point, line or surface) that are projectively transformed. The main disadvantage of Euclidean geometry is the problem of modeling the points at infinity. In perspective, two parallel lines meet at vanishing point at infinity, however, such a case is not easily modeled by the Euclidean geometry. Additionally, projection of a 3D point onto an image plane in Euclidean geometry requires a perspective scaling operation which means a division by the scaling factor and, thus, becomes a non-linear operation.

The projective geometry introduces *homogeneous coordinates*: a point in 3D space is described using a 4-element vector $(X_1, X_2, X_3, X_4)^T$ instead of *inhomogeneous coordinates*, a 3-element vector $(X, Y, Z)^T$ used in Euclidean geometry. The relation between two types of coordinates is:

$$X = \frac{X_1}{X_4}, \qquad Y = \frac{X_2}{X_4}, \qquad Z = \frac{X_3}{X_4}, \qquad X_4 \neq 0 \tag{2.22}$$

As a generalization, the isomorphic mapping between a point in the n-dimensional Euclidean space to projective space can be written as:

$$(X_1, X_2, \dots, X_n)^T \rightarrow (\lambda X_1, \lambda X_2, \dots, \lambda X_n, \lambda)^T \tag{2.23}$$

where λ corresponds to a free scaling parameter, called the *homogeneous scaling factor*.

The relation described in Eq. 2.11 can be expressed using the projective geometry in a matrix notation as:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.24}$$

where $\lambda = Z$ is the homogeneous scaling factor.

Additionally, if we incorporate the intrinsic and extrinsic camera parameters to the above relation, an equation describing the perspective projection for the pinhole camera model is obtained [Cyg02]:

$$\lambda \boldsymbol{p} = [\boldsymbol{K}|\boldsymbol{0_3}] \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{RT} \\ \boldsymbol{0_3}^T & 1 \end{bmatrix} \boldsymbol{P} = \boldsymbol{KR} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \boldsymbol{KRT} \tag{2.25}$$

where $\boldsymbol{p} = (x_u, y_u, 1)^T$ represents homogeneous coordinates of pixel positions of a 3D world point (X, Y, Z) projected onto the image plane Π (Eq. 2.10), $\boldsymbol{P} = (X, Y, Z, 1)^T$ represents homogeneous coordinates of a 3D world point with Euclidean coordinates (X, Y, Z), $\boldsymbol{K}$ is the projection matrix (Eq. 2.15), $\boldsymbol{0_3}$ is the all zero element vector, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation matrix (Eq. 2.21).

2.6.4. Depth Image Based Rendering

Depth Image Based Rendering (DIBR) is a 3D image warping technique, designed for the rendering of a synthetic image using a reference texture image and a corresponding depth image [McMi97]. Let us consider a 3D world point at homogeneous coordinates $\boldsymbol{P} = (X, Y, Z, 1)^T$, captured by two cameras and projected onto the reference and synthetic image planes at pixel positions $\boldsymbol{p}_1 = (x_1, y_1, 1)^T$ and $\boldsymbol{p}_2 = (x_2, y_2, 1)^T$ respectively (see Fig. 2.16).

Based on the relation introduced in Section 2.6.3 (Eq. 2.25) we can determine the pixel positions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ as follows:

$$\lambda_1 \boldsymbol{p_1} = \boldsymbol{K_1 R_1} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \boldsymbol{K_1 R_1 T_1} \tag{2.26}$$

$$\lambda_2 \boldsymbol{p_2} = \boldsymbol{K_2 R_2} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \boldsymbol{K_2 R_2 T_2} \tag{2.27}$$

where $i = \{1,2\}$ is a camera index, λ_i represent the homogeneous scaling factors, $\boldsymbol{K_i}$ is the projection matrix, $\boldsymbol{R_i}$ is the rotation matrix and $\boldsymbol{T_i}$ is translation matrix of the camera i.

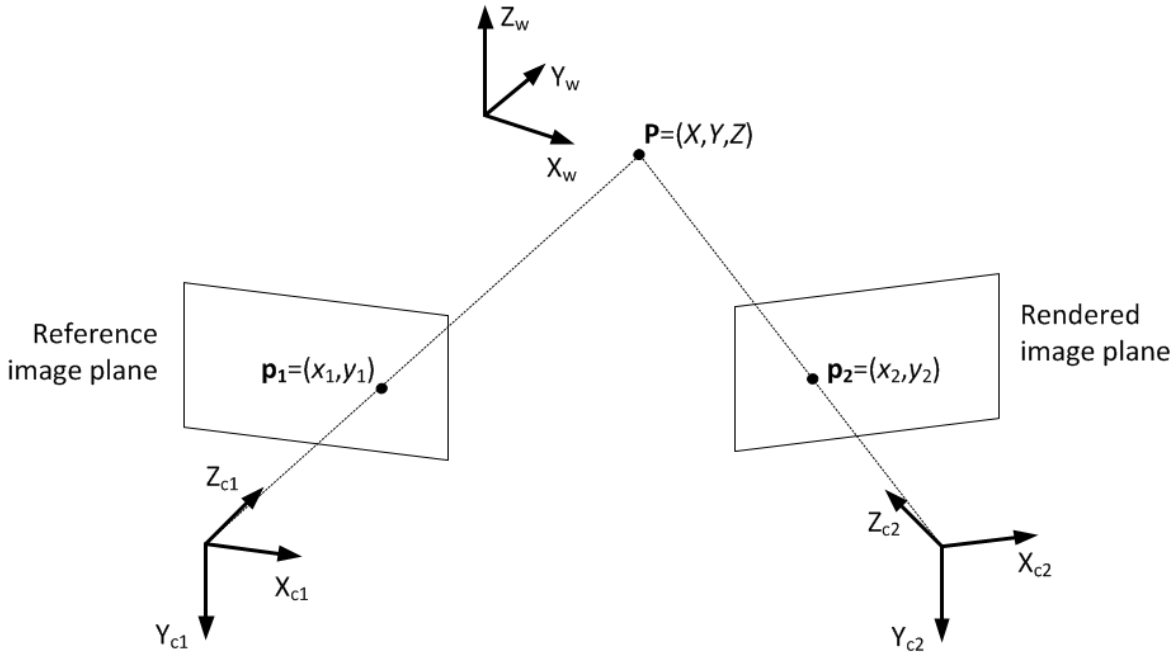


Fig. 2.16. Rendering using 3D image warping.

From Eq. 2.26, the position of the 3D world point $\boldsymbol{P}$ in Euclidean coordinates can be determine as:

$$(X, Y, Z)^T = (\boldsymbol{K_1 R_1})^{-1} \cdot (\lambda_1 \boldsymbol{p_1} + \boldsymbol{K_1 R_1 T_1}) \tag{2.28}$$

Finally, the relation between pixel position $\boldsymbol{p}_2$ in the virtual view and $\boldsymbol{p}_1$ in the reference view can be written as:

$$\lambda_2 \boldsymbol{p_2} = \boldsymbol{K_2 R_2} (\boldsymbol{K_1 R_1})^{-1} \cdot (\lambda_1 \boldsymbol{p_1} + \boldsymbol{K_1 R_1 T_1}) - \boldsymbol{K_2 R_2 T_2} \tag{2.29}$$

Eq. 2.29 constitutes the 3D image warping equation [McMi97] that enables the synthesis of the virtual view from a reference texture view and a corresponding depth image.

As a consequence of 3D image warping defined in DIBR algorithm, the synthesized image consists of *visible*, *overlapped* and *undefined* pixels. Overlapped pixels result from the fact that multiple pixels from the reference texture view can be projected onto the same pixel position in the virtual view. This happens for example, when a foreground pixel occludes a background pixel in the rendered view. Additionally, some regions in the virtual view are not visible from the original viewpoint. Consequently, holes containing pixels with undefined values are produced in the virtual image.

Chapter 3

Proposed depth-based inter-view prediction techniques

3.1. Main idea and motivation

In this chapter, new techniques for depth-based inter-view prediction of motion information are proposed to improve the overall coding efficiency of a multiview video codec with available depth information, e.g. stereoscopic depth. In previous approaches, relatively simple techniques of inter-view data prediction based on global disparity model or simple block disparity estimation have been used. These techniques were introduced in coding tools such as *Motion Skip* [Koo06, Yang09] (see Section 2.5). Here, utilization of more sophisticated and accurate techniques is presented. The author has proposed two novel methods of incorporation of 3D-space modeling based on depth information into multiview video coding. The methods allow accurate point-to-point prediction of motion information from the reference view. Number of variants of the proposed methods have been also introduced and analyzed.

The purpose of the proposed algorithms is to predict motion information for every image point in the coded view using the data assigned to image point in available reference views. In order to allow such prediction, a mapping between coordinates of each point of the coded image with corresponding point in the reference image must be found (Fig. 3.1c). However, as the coded and reference images originate from the same time instance, but different views of the same multiview sequence, the mapping algorithm should take into consideration complex scene geometry dependencies in order to assure that accuracy requirement of the assignment is fulfilled. Utilization

of the global disparity vector or simple block disparity estimation as the determinants of correspondence between different views may lead to significant errors. This happens when unified value of global disparity is used to describe disparity between areas of the visual scene that have very different distance from the camera (see Fig. 3.1a). Similarly, usage of local disparity assigned to blocks, even as small as 4×4 pixels, may cause inaccuracy, as it does not preserve depth discontinuities on object borders (Fig. 3.1b). Therefore, in the proposed approach, 3D-space modeling techniques based on the image warping with available depth information have been incorporated into the proposed inter-view prediction algorithms. In the proposed methods, each point of the coded image is mapped using Depth Image Based Rendering (DIBR) technique (see Section 2.6.4) with a corresponding point in the reference view. The mapping is done independently for each point of the coded image using available depth information assigned to this point. Consequently, every image point has its own, individual disparity value assigned and discontinuities between objects in the scene are preserved (Fig. 3.1c). Problems of both global disparity and block disparity approaches are reduced. As a result, the proposed methods can lead to more efficient utilization of the existing inter-view redundancy between the views of a multiview sequence.

If the distance between neighboring viewpoints is small, i.e. comparable with the eye interocular distance, a high correlation between content of the video sequences obtained from the viewpoints exists [Feck05, Merk07, Su06]. In particular, motion fields of the neighboring views are highly correlated [Guo06]. Consequently, motion information of a coded side view of the multiview sequence, including motion vectors and reference picture indices, can be derived from the neighboring, already encoded view at the same time instant without the need to retransmit it again in the coded view. Obviously, this results in improved compression efficiency of the multiview codec.

However, simple translational motion model with motion vectors assigned on a block level limits the correlation of motion fields if motion information is derived between the views with a simple block-to-block reassignment. The abovementioned problem is illustrated in Fig. 3.2. If we consider the video content of the pictures in the reference and coded views, block partitioning in the reference view does not correspond to block partitioning in the coded view. Moreover, a rectangular block of pixels in the reference view usually does not correspond to a rectangular block in the coded view. As a result, motion information assigned to a block in the reference view might not be a good prediction for a rectangular block in the coded view. Therefore utilization of the proposed method of point-to-point depth-based inter-view prediction as a new technique of motion information prediction should further increase the compression ratio of multiview sequence coding.

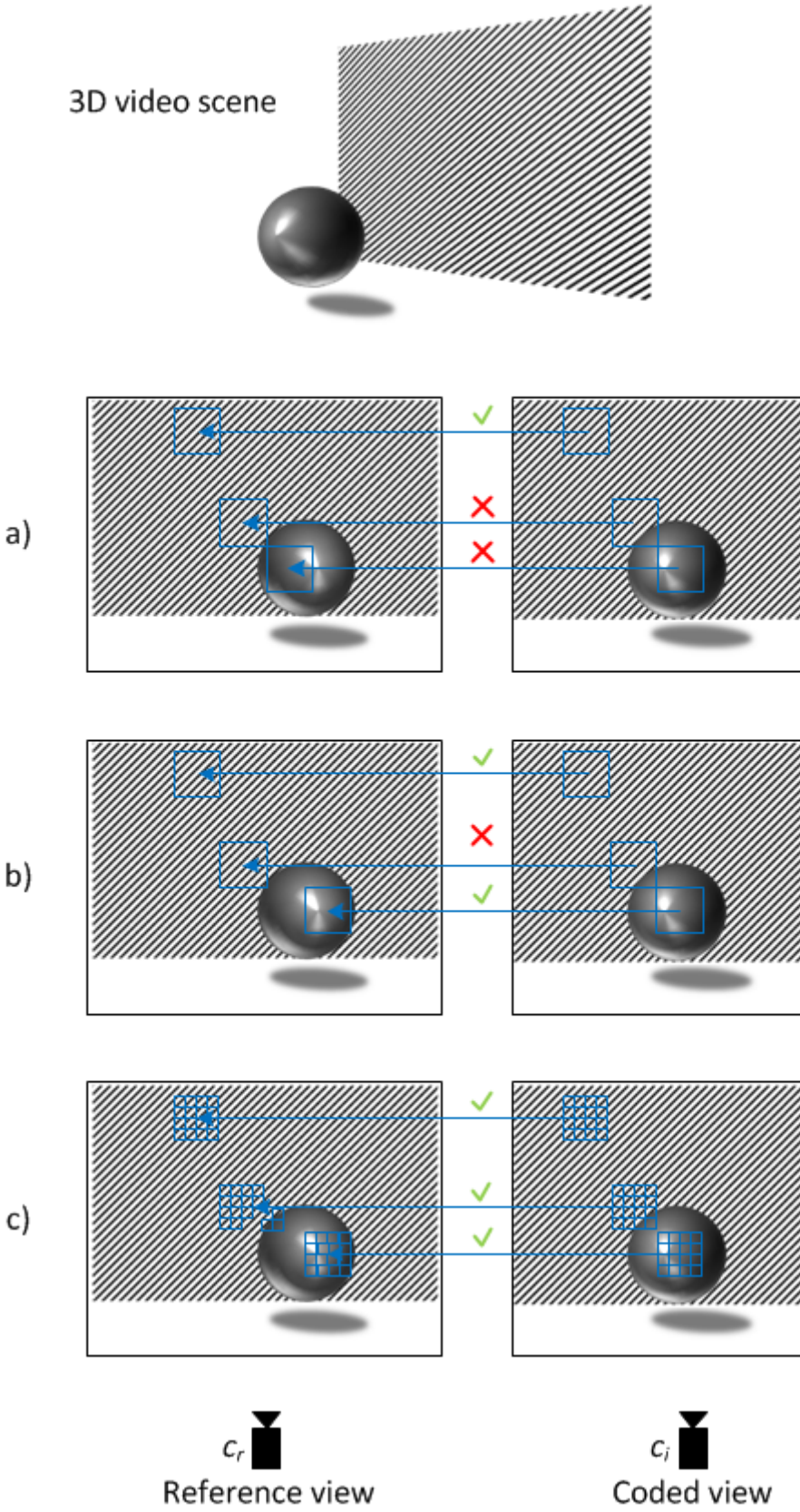


Fig. 3.1. Inter-view prediction approaches: a) global disparity vector, b) local block disparity, c) proposed – point-to-point correspondence based on depth information.

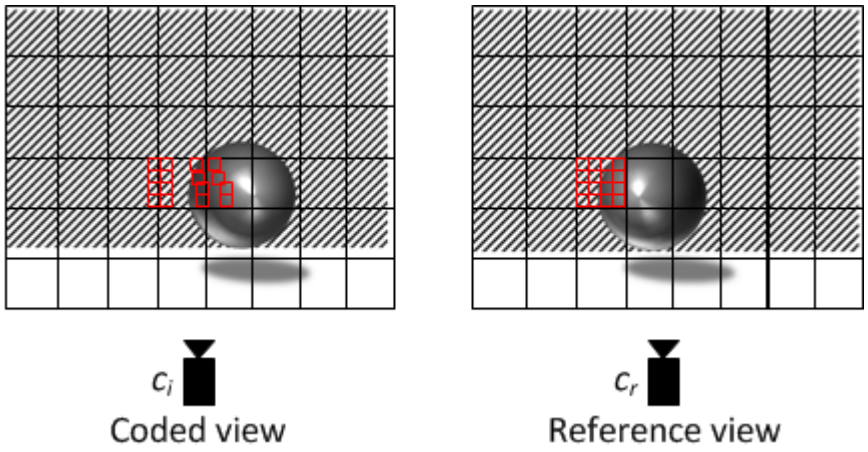


Fig. 3.2. Problems of inter-view block-to-block assignment.

In the next section, two point-to-point inter-view prediction techniques utilizing depth information are introduced. Following the *forward* and *inverse mapping* approaches of virtual images rendering known from computer graphics [Lav94, McMi97, Sha98], the proposed methods have been called Forward Projection Depth-Based Prediction (FPDBP) and Inverse Projection Depth-Based Prediction (IPDBP) respectively. Each method is designed for special cases of availability of depth information in reference and coded views. As a consequence, the proposed idea can be successfully adopted into wide range of multiview coding scenarios.

3.2. Proposed depth-based inter-view prediction techniques

3.2.1. Forward Projection Depth-Based Prediction (FPDBP)

Forward Projection Depth-Based Prediction (FPDBP) algorithm is a point-to-point inter-view prediction technique utilizing depth information of the coded view. The algorithm uses Depth Image Based Rendering (DIBR) technique and forward mapping approach of virtual images rendering known from computer graphics in order to obtain the mapping between corresponding points in coded and reference views. Positions of the corresponding points in the reference view are determined as follows (Fig. 3.3):

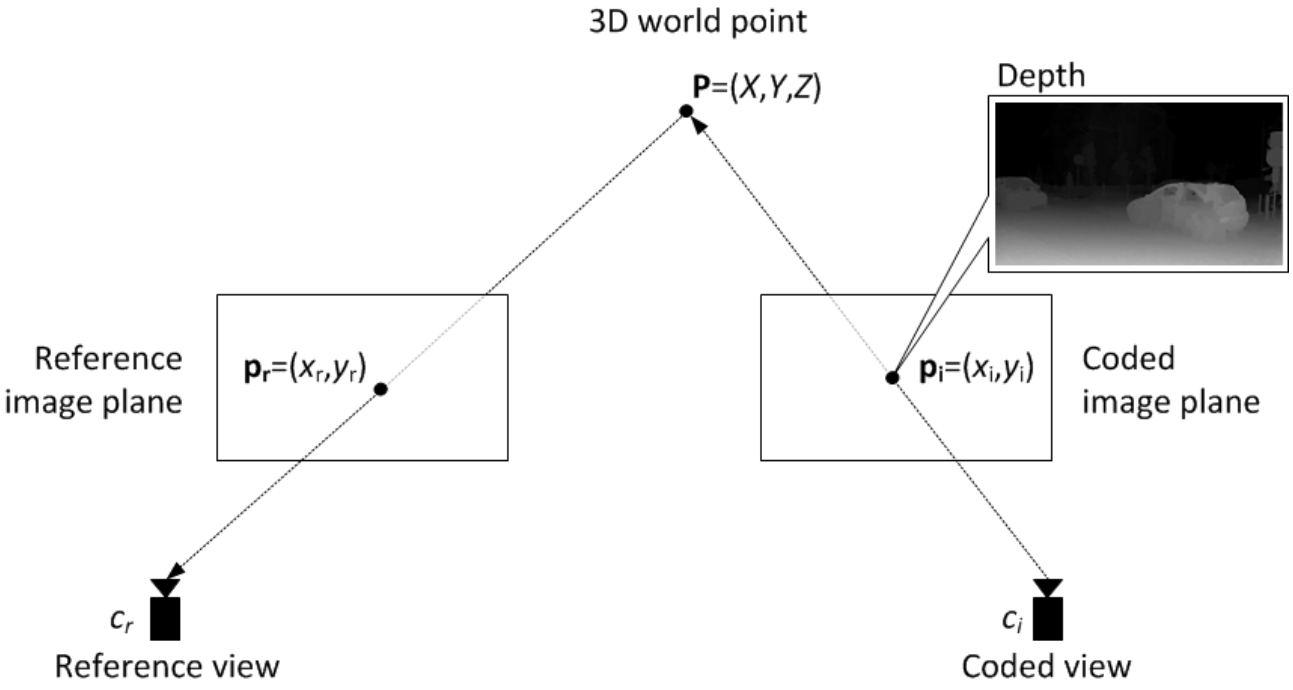


Fig. 3.3. Scheme of Forward Projection Depth-Based Prediction.

1) For each image pixel in the coded view c_i, project pixel location $\boldsymbol{p_i} = (x_i, y_i, 1)^T$ in the coded image from view c_i into image plane of the reference view c_r using DIBR technique (refer to Eq. 2.29 in Section 2.6.4):

$$\lambda_r \boldsymbol{p_r} = \boldsymbol{K_r R_r} (\boldsymbol{K_i R_i})^{-1} \cdot (\lambda_i \boldsymbol{p_i} + \boldsymbol{K_i R_i T_i}) - \boldsymbol{K_r R_r T_r} \tag{3.1}$$

where λ represents the homogeneous scaling factor, $\boldsymbol{K}$ is the projection matrix, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is translation matrix describing camera parameters for each of the views: c_i and c_r. The result of such projection is the corresponding pixel position in image plane of the reference view c_r, equal to $\boldsymbol{p_r} = (x_r, y_r, 1)^T$. In order to obtain this position, the homogeneous scaling factor λ_i must be known. As stated in Section 2.6.3, for the purpose of projective geometry, we assume λ_i to be equal to the distance of projected world point $\boldsymbol{P}$ from optical center of the view c_i, which is specified by the depth information available e.g. in form of stereoscopic depth for view c_i.

2) Next, corresponding pixel positions $\boldsymbol{p_i}$ and $\boldsymbol{p_r}$ are tested to determine visible pixels. This is done in a step called occlusion detection, which is described in details in Section 3.3. If pixel $\boldsymbol{p_i}$ is visible in the reference view c_r, the corresponding pixel $\boldsymbol{p_r}$ can be assigned as the reference for inter-view prediction. Otherwise, prediction from the corresponding image point $\boldsymbol{p_r}$ in the reference view is not possible and pixel $\boldsymbol{p_i}$ remains unassigned. In such a case another available reference view should be inspected.

3) If no other reference view is available, the corresponding pixel position for pixel $\boldsymbol{p}_i$ is derived from the neighboring pixels of the same view, for which the corresponding pixel $\boldsymbol{p}_r$ was successfully found in the reference view. This step is done by a dedicated filling algorithm which variants are described in Section 3.4. The filling algorithm is applied to every image area that contains unassigned pixels.
4) In cases the filling algorithm fails to find the corresponding pixel $\boldsymbol{p}_r$ or there is no data to be derived from the corresponding pixel $\boldsymbol{p}_r$, another possibility to gain predicted data for pixel $\boldsymbol{p}_i$ is to use a standard intra-view predictor available in the codec. Of course, this predictor is codec-dependent and may be different for each type of predicted data. E.g. in case of AVC codec and motion information prediction, the standard Direct mode prediction with median predictor can be applied. For HEVC codec, the best available predictor in the merge candidate list may be utilized.
5) Finally, when the source of predicted data is determined for each pixel of encoded block, the block can be encoded. As the derived data may be different for each pixel, the encoding process is, to some extent, independent for every pixel of the block.

The only information which is derived from the reference view is the data originally encoded in the bitstream and there are no additional calculations or restrictions to encoding process for the reference views. The computational overhead in the encoder/decoder is related to DIBR, occlusion testing and filling algorithm procedures. For detailed discussion on the computational complexity of the proposed algorithms, please refer to Section 6.5.

The issue of pixel coordinates rounding should also be clarified. In the projection of pixel position $\boldsymbol{p}_i$ from the coded to the reference view, coordinates of the corresponding pixel $\boldsymbol{p}_r$ are, in general, the floating point numbers. This obviously makes it necessary to perform the rounding operation to obtain an integer pixel position, following the equation:

$$\widehat{\boldsymbol{p}}_r = (\hat{x}_r, \hat{y}_r, 1)^T = (\lfloor x_r + 0.5 \rfloor, \lfloor y_r + 0.5 \rfloor, 1)^T \tag{3.2}$$

This simple heuristic technique relays on mapping the sub-pixel coordinate $\boldsymbol{p}_r$ to the nearest integer pixel position $\widehat{\boldsymbol{p}}_r$. In case of texture synthesis in computer graphics, the above procedure is a generally accepted approach which does not affect the quality of synthesized view significantly [Iyer10]. However, it is also possible to apply an additional re-sampling and interpolation of texture value in the target point of the synthesis. Some of the well-known solutions are triangular meshes rendering or linear interpolation based on the neighboring points values and utilization of an intermediate image [Wol90]. Nevertheless, for the purpose of inter-view prediction of e.g. motion information, attempts to apply such interpolation or re-sampling methods are difficult to justify. If

the pixels surrounding the sub-pixel position $\boldsymbol{p}_r$ have the same motion information assigned, interpolation is not necessary. On the other hand, if the surrounding pixels have different motion information assigned, which occurs e.g. in case of pixels belonging to separated objects moving in various directions, interpolation of such different motion fields is also inappropriate.

3.2.2. Inverse Projection Depth-Based Prediction (IPDBP)

Inverse Projection Depth-Based Prediction (IPDBP) algorithm is another point-to-point inter-view prediction technique. However, the IPDBP algorithm does not require the availability of depth information from the coded view, as it uses depth of the reference view only. The algorithm also utilizes Depth Image Based Rendering (DIBR) technique in order to obtain the mapping between corresponding points in coded and reference views, but forward mapping approach of virtual images rendering used in FPDBP algorithm has been substituted by the inverse mapping approach. As a consequence, positions of the corresponding points in the reference view are determined as follows (Fig. 3.4):

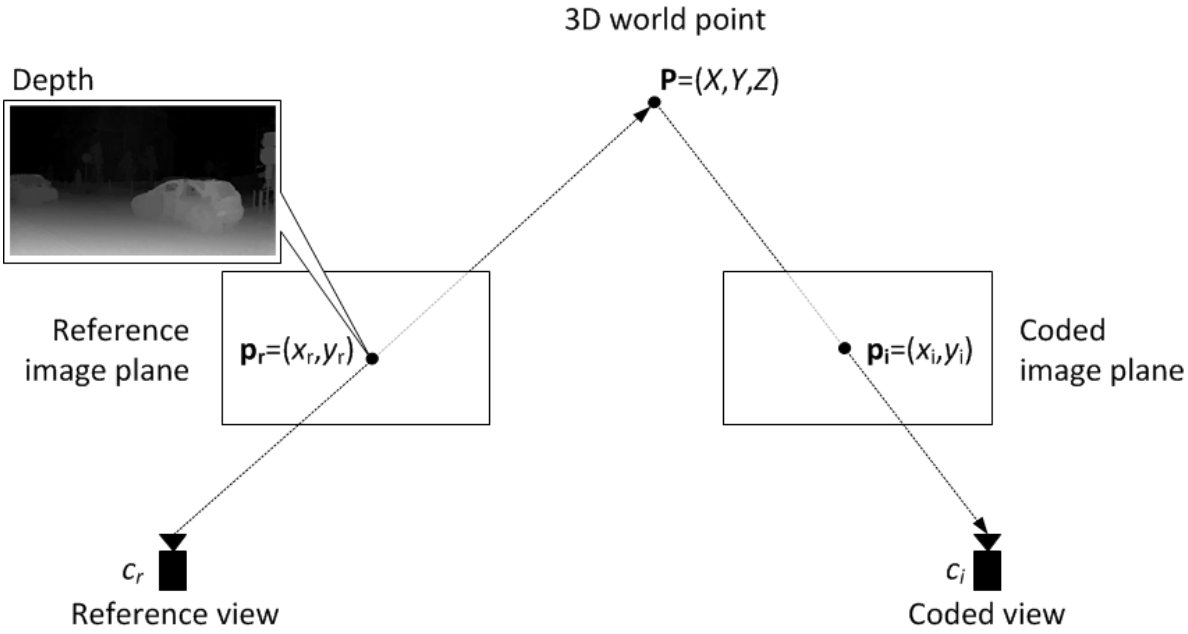


Fig. 3.4. Scheme of Inverse Projection Depth-Based Prediction.

1) Mark all pixels in the coded view c_i as not visible.
2) Take first available reference view. For each image pixel in the reference view c_r, project pixel location $\boldsymbol{p}_r = (x_r, y_r, 1)^T$ in the reference image from view c_r into image plane of the coded view c_i using DIBR technique (see Eq. 2.29 in Section 2.6.4):

$$\lambda_i \boldsymbol{p_i} = \boldsymbol{K_i R_i} (\boldsymbol{K_r R_r})^{-1} \cdot (\lambda_r \boldsymbol{p_r} + \boldsymbol{K_r R_r T_r}) - \boldsymbol{K_i R_i T_i} \quad (3.3)$$

where λ represents the homogeneous scaling factor, $\boldsymbol{K}$ is the projection matrix, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is translation matrix describing camera parameters for each of the views: c_i and c_r. The result of such projection is the corresponding pixel position in image plane of the coded view c_i, equal to $\boldsymbol{p}_i = (x_i, y_i, 1)^T$. In this case, the position is obtained based on the known homogeneous scaling factor λ_r, which is equal to the distance of projected world point $\boldsymbol{P}$ from optical center of the view c_r (see Section 2.6.3) and specified by the depth information available e.g. in form of stereoscopic depth for view c_r.

3) If pixel $\boldsymbol{p}_i$, determined with the above method, is still marked as not visible, position of pixel $\boldsymbol{p}_r$ in the reference view is assigned to $\boldsymbol{p}_i$ as the corresponding position for inter-view prediction and $\boldsymbol{p}_i$ is marked as visible. Otherwise, a corresponding pixel position in the reference view was already assigned to pixel $\boldsymbol{p}_i$. Let us refer to this position as $\boldsymbol{p}_r{}'$. If the pixels $\boldsymbol{p}_r$ and $\boldsymbol{p}_r{}'$ originate from different views, the corresponding pixel position for $\boldsymbol{p}_i$ remains unchanged. If not, pixels $\boldsymbol{p}_r$ and $\boldsymbol{p}_r{}'$ represent different 3D world points that are projected into the same position $\boldsymbol{p}_i$ in the coded view. This conflict is a typical problem of overlapping pixels, which is solved by simply comparing the depth values assigned to pixels $\boldsymbol{p}_r$ and $\boldsymbol{p}_r{}'$ and choosing the one that is closer to the camera.
4) Next, if there are still pixels marked as not visible in the coded view c_i after projecting all pixel positions from the reference view c_r, another available reference view is used for the inverse mapping.

   An important consequence of this procedure is that every pixel in the coded view c_i which remains unassigned after inspecting all available reference views is assumed to be not visible. As a result, no further occlusion detection is required in IPDBP, which is a significant difference when compared to FPDBP algorithm.
5) Finally, for all pixels of the coded view c_i which still remain unassigned, filling algorithm and intra-view predictor are utilized to determine the source of predicted data. These steps are the same as described in points 3) and 4) of FPDBP algorithm (see Section 3.2.1).
6) Perform encoding process of the block using data predicted independently for each pixel of encoded block (refer to step 5) of FPDBP algorithm in Section 3.2.1).

Similarly as in FPDBP algorithm, the data originally encoded in the bitstream is the only information derived from the reference view for IPDBP algorithm and there are no additional calculations or restrictions to encoding process for the reference views. The computational overhead

in the encoder/decoder is related to DIBR with overlapped pixel testing and filling algorithm procedures. Detailed discussion on the computational complexity of the proposed algorithms is presented in Section 6.5.

The issue of pixel coordinates rounding is also present in IPDBP algorithm. During the projection of pixel position $\boldsymbol{p}_r$ from the reference to the coded view, coordinates of the corresponding pixel $\boldsymbol{p}_i$ need to be rounded. The procedure applied for that purpose is the same as the one described for FPDBP algorithm (see Eq. 3.2 in Section 3.2.1).

3.3. Occlusion detection algorithms

A consequence of 3D image warping with DIBR algorithm is that some of visible pixels in one view are overlapped in the other view by pixels representing objects located closer to the camera. The coordinates of overlapping pixels might differ significantly in the view from which pixels are projected (Fig. 3.5). Thus, it is important to detect such cases to avoid assignment of incorrect inter-view corresponding points. This procedure is called an occlusion detection and is especially required in FPDBP algorithm (Section 3.2.1) as no other pixel visibility tests are implemented in this algorithm. In the following paragraphs, a number of proposed algorithms for occlusion detection are presented.

Without the loss of generality, in the following description of the algorithms, we assume that occlusion detection is performed for each pixel $\boldsymbol{p}_r = (x_r, y_r, 1)^T$ in the reference view. Pixel $\boldsymbol{p}_r$ is an inter-view corresponding pixel for pixel $\boldsymbol{p}_i = (x_i, y_i, 1)^T$ in the coded view and its position was determined using projection of pixel $\boldsymbol{p}_i$ position from coded view into image plane of the reference view.

The first of the proposed algorithms is a *back-projection* algorithm which was proposed by the author for occlusion detection. For each pixel $\boldsymbol{p}_r$ in the reference view, a reverse projection is performed. This means that pixel position $\boldsymbol{p}_r$ in the reference view is projected back to the coded view based on the depth information of the reference view, resulting in corresponding pixel position $\boldsymbol{p}_i{}' = (x_i{}', y_i{}', 1)^T$ (see Fig. 3.6). If the original pixel position in the coded view $\boldsymbol{p}_i$ differs from the position of back-projected pixel $\boldsymbol{p}_i{}'$, pixel $\boldsymbol{p}_i$ is assumed to be not visible in the reference view. Consequently, the assignment of the inter-view corresponding pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_r$ is assumed to be incorrect.

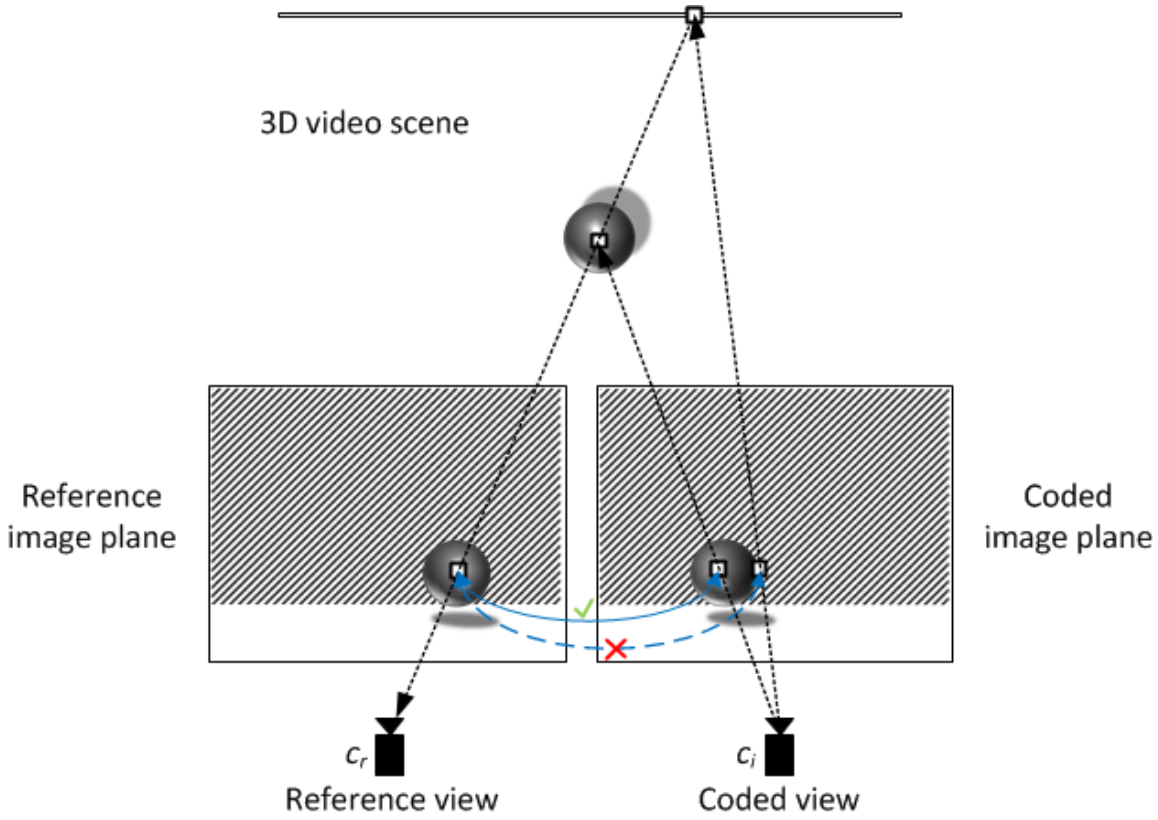


Fig. 3.5. Problem of incorrect assignment of inter-view corresponding pixels due to occlusion.

However, due to rounding operation applied while calculating integer coordinates for pixels $\boldsymbol{p}_r$ and $\boldsymbol{p}_i'$ (refer to Section 3.2.1), a small margin of error should be preferably accepted during the comparison of pixel $\boldsymbol{p}_i'$ and $\boldsymbol{p}_i$ positions. Thus, we propose to assume that pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$ represent the same 3D world point if the distance between their positions is not bigger than some pre-defined value T:

$$Not\ occluded\ if: \quad \begin{cases} |x_i - x_i'| \leq T \\ |y_i - y_i'| \leq T \end{cases} \tag{3.4}$$

As a result of studies with synthetic 3D sequences, the value of T has been set to 2. This value results also from the theoretical considerations on maximum possible truncation error of the rounding operation of floating point pixel coordinates to integer values. Truncation to the nearest integer position produces maximum error equal to 1. The truncation is performed twice: in projection from the coded view to the reference view and in back-projection. As the truncation errors sum up in the worst case, the maximum truncation error which includes both projections is equal to 2.

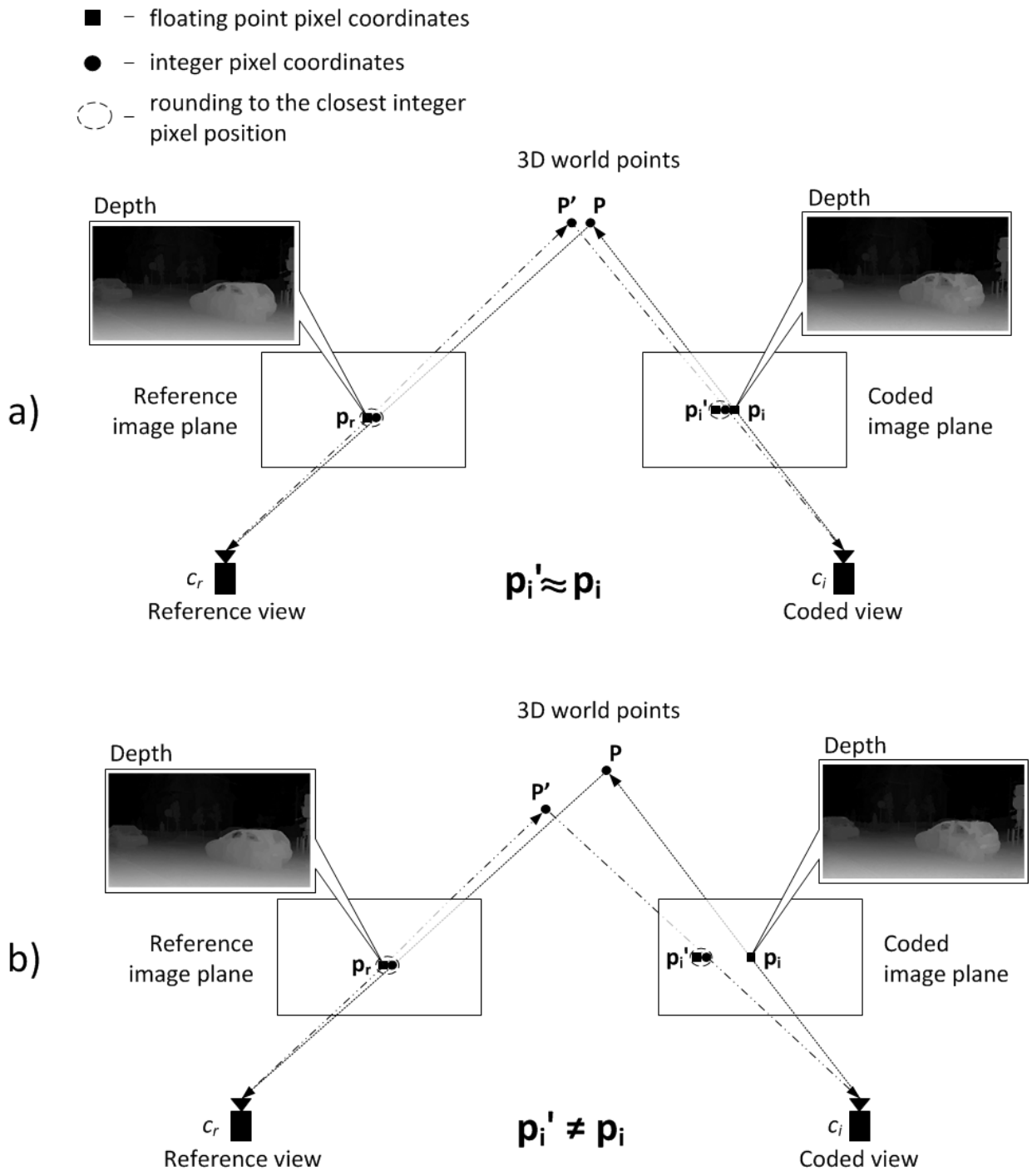


Fig. 3.6. Occlusion detection using back-projection algorithm: a) pixel $\boldsymbol{p}_i$ is visible in reference view, b) pixel $\boldsymbol{p}_i$ is occluded.

Second of the presented occlusion detection algorithms is a *z-test* algorithm, known also from computer graphics and texture synthesis as *z-buffering* [Berg00]. The idea of z-test is to compare the values of depth information assigned to corresponding pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_r$. If depth value D_r assigned to corresponding pixel $\boldsymbol{p}_r$ indicates that the distance of the 3D world point represented by

this pixel is smaller than the distance indicated by depth value D_i of pixel $\boldsymbol{p}_i$, occlusion is detected and pixel $\boldsymbol{p}_i$ is assumed to be not visible in the reference view. Otherwise, pixel $\boldsymbol{p}_i$ is visible and the assignment of the inter-view corresponding pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_r$ is considered to be valid.

As a consequence, the z-test algorithm is very simple and presents low computational complexity. No reverse projection of the pixel position is required and the only operation performed during the occlusion test is simple depth value comparison. However, in case depth information is in form of disparity maps, which is very common, disparity calculated for one view need to be scaled to a value range represented in the other view [ISO10]. As the disparity values in disparity maps are usually quantized to integer values, e.g. in the range of $\{0,255\}$, and the scaled disparity value from the other view is, in general, a floating point number, a small margin of error should be preferably accepted during the comparison of the two disparity values:

$$Occluded\ if: \qquad Disp_r > S_{i\to r}(Disp_i) + E \tag{3.5}$$

where: $Disp_r$ is a disparity value assigned to pixel $\boldsymbol{p}_r$ in the reference view, $S_{i\to r}(Disp_i)$ is a disparity value $Disp_i$ of the pixel $\boldsymbol{p}_i$ in the coded view scaled to range of disparity values of the reference view and E is a value of accepted margin of error. The value of E was set to 1, which is the width of quantization interval used for integer value disparity maps.

Both of the above algorithms perform well for synthetic video sequences, however, produce poor results in case of natural video. The reason is that, in fact, the algorithms check the consistency of depth information available for coded and reference views. As a consequence, the algorithms are very sensitive to inter-view inconsistency of depth information, which can lead to incorrect decisions in occlusion detection. Unfortunately, this is a common case for natural video, for which depth information is obtained using dedicated depth estimation algorithms. Although depth estimation is an intensively growing field, many of the existing algorithms, despite their huge computational complexity, still produce deficient results [Zit04, Lee10]. This happens especially for light-reflecting and transparent objects or texture-deficient regions of the visual scene. Consequently, the resultant depth information is inconsistent not only between subsequent time instances, but also between different views of the multiview sequence, which restricts the range of applications of the occlusion detection algorithms presented so far.

As a result, the above occlusion detection algorithms will not be investigated in detail in presented experimental results (Section 6.1). For further considerations, the *z-test* algorithm was selected as a representative of this group, as it presents similar performance, but much smaller complexity and, consequently, better perspectives for practical applications.

In order to limit the influence of inter-view inconsistency of depth information, another algorithm for occlusion detection, called *o-mask*, have been proposed by the author. The algorithm utilizes a dedicated occlusion mask which is used to determine which pixels in the reference view was already assigned to pixels in the coded view. The information about the pixel coordinates is also stored. The whole procedure is very similar to the pixel visibility check performed in step 3) of the IPDBP algorithm (see Section 3.2.2) and is described as follows: for each pixel $\boldsymbol{p}_r$ in the reference view that is assigned to its inter-view corresponding pixel $\boldsymbol{p}_i$ in the coded view, store the coordinates of pixel $\boldsymbol{p}_i$ and mark pixel $\boldsymbol{p}_r$ as assigned to pixel $\boldsymbol{p}_i$ in the occlusion mask. Also, set pixel $\boldsymbol{p}_i$ as visible in the coded view. When another pixel in the coded view, indicated by $\boldsymbol{p}_i'$, is projected into the location of pixel $\boldsymbol{p}_r$, a simple comparison of depth values assigned to pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$ is performed based on the depth information of the coded view. If pixel $\boldsymbol{p}_i$ represents a 3D world point $\boldsymbol{P}$ that is closer to the camera that the one represented by pixel $\boldsymbol{p}_i'$, pixel $\boldsymbol{p}_i'$ is set to be not visible in the reference view. The assignment between pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_r$ remains unchanged. Otherwise, pixel $\boldsymbol{p}_i'$ occludes pixel $\boldsymbol{p}_i$ and should be set as visible in the coded view. In this case, the assignment between pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_r$ is incorrect. Consequently, pixel $\boldsymbol{p}_i$ should be set as not visible and values stored in occlusion mask for pixel $\boldsymbol{p}_r$ should be appropriately overridden with the data of pixel $\boldsymbol{p}_i'$.

As the algorithm utilizes only depth information of the coded view, the problem of inter-view inconsistency of depth information does not occur. Similarly as z-test algorithm, the above method is not computationally complex, however, requires an additional buffer of size corresponding to the image resolution to store coordinates of assigned pixels from the coded view for each pixel $\boldsymbol{p}_r$ and also information whether the pixel $\boldsymbol{p}_r$ has already been assigned. This obviously results in higher memory consumption of the algorithm when compared to the two previously described.

There are many other methods for occlusion detection known from literature. Among them, methods based on the *Markov Random Fields* or iterative *belief propagation* and *graph cuts* algorithms should be particularly mentioned. These algorithms have recently attracted much interest, especially in applications of stereo matching and depth estimation. In [Sun05] authors propose a symmetric stereo model for occlusion detection and handling which consists in visibility constraint that forces disparity and occlusion to be consistent between the views. On the other hand, in [Fran06] authors describe the probabilistic imaging model, in which visible and occluded regions are generated by two separate processes and the region partitioning is made explicit by the introduction of an hidden binary visibility map. Unfortunately, these methods usually require texture images from both views, which disqualifies their application at the decoder side. Moreover,

the computational complexity of these methods is enormous and often exceeds the one of modern, state-of-the-art video codecs. Obviously, this fact excludes the use of such algorithms in practical implementations, especially if mobile user terminals are concerned. As a consequence, the author of this thesis did not include the abovementioned methods in further considerations.

3.4. Algorithms for unassigned area filling

After the 3D image warping with DIBR algorithm and application of the occlusion detection algorithm, some of the pixels in the coded view may still remain unassigned to any of the inter-view corresponding pixels in the reference views. This is obviously caused by occlusion and the fact, that not all of the pixels in the coded view are visible in any of the available reference views. Fig. 3.7 shows an example of such situation: green pixels indicate image areas that are not visible in the reference view, which, in this case, is located to the right of the current view. However, unassigned pixels may also occur due to wrong decisions of the occlusion detection algorithm or synthesis artifacts related to the problem of undefined pixels. The latter case can appear in the IPDBP algorithm, where positions of corresponding pixels are projected from the reference view into the coded view. As a result of rounding operations performed on floating point pixel coordinates to match the integer pixel grid of the image, some of the pixels in the coded view may not be assigned. Using the nomenclature of computer graphics, such pixels are called the undefined pixels. Examples of undefined pixels can be observed in Fig. 3.7 in form of thin, usually single-pixel lines on the objects with large surface, like walls or a column in presented picture.

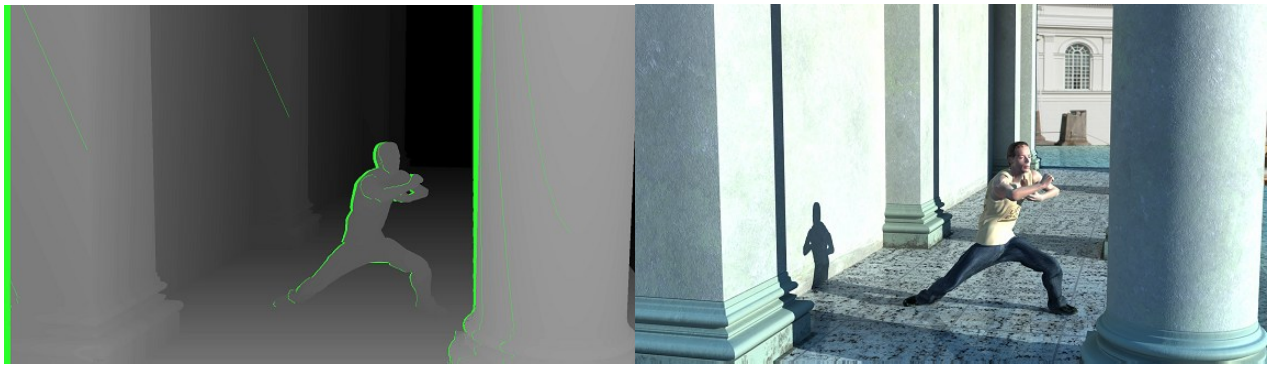


Fig. 3.7. Unassigned pixels due to occlusion and synthesis artifacts: left – depth map with unassigned pixels (green), right – texture image.

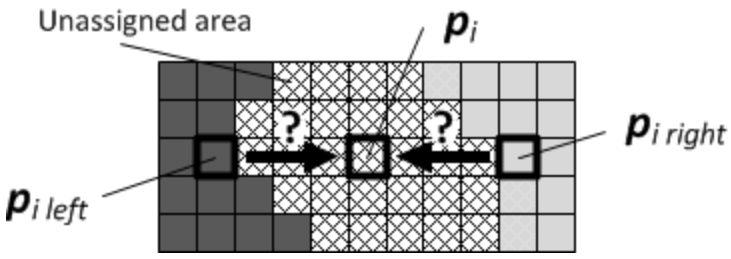


Fig. 3.8. Unassigned are filling based on the nearest left and right neighboring pixels.

To resolve the problem of unassigned pixels in the coded view, dedicated filling algorithms are proposed by the author. Let us denote the unassigned pixel position in the coded view as $\boldsymbol{p}_i$. The positions of the neighboring pixels nearest to the pixel $\boldsymbol{p}_i$, for which a corresponding pixels in the reference view was successfully found are $\boldsymbol{p}_{i\,left}$ and $\boldsymbol{p}_{i\,right}$ for left and right neighbors respectively (see Fig. 3.8). To determine the neighboring pixel of the same view from which the corresponding pixel position for pixel $\boldsymbol{p}_i$ is derived, the following variants of filling algorithms are introduced:

- *FILLmax* – neighboring pixel with larger distance from the camera is selected. This variant of the algorithm favors background pixels as the source for derivation of corresponding pixel position, based on the observation that most of the unassigned pixels $\boldsymbol{p}_i$ are the occluded background pixels. In case of FPDBP algorithm, depth values of pixels $\boldsymbol{p}_{i\,left}$ and $\boldsymbol{p}_{i\,right}$ are available in form of depth information for the coded view. However, for IPDBP algorithm, depth information is not provided for the coded view. To solve this problem, in case of IPDBP algorithm, presented filling algorithms utilize depth information of pixels from the reference views that correspond to pixels $\boldsymbol{p}_{i\,left}$ and $\boldsymbol{p}_{i\,right}$.
- *FILLmin* – neighboring pixel with smaller distance from the camera is selected. In this variant of the algorithm, we include the fact that some of the unassigned pixels are the parts, e.g. side walls, of the foreground objects. Moreover, because the derived position of the corresponding pixel in the reference view can be used to predict e.g. motion information, it is also rational to prefer motion information of the foreground objects as the motion-compensated error for the foreground objects is usually much more annoying.
- *FILLsim* – neighboring pixel with most similar depth is selected. This variant of the algorithm takes into consideration that each of the above, predefined assumptions may not always be accurate. As a result, the decision is made based on the similarity between depth values assigned to pixel $\boldsymbol{p}_i$ and pixel $\boldsymbol{p}_{i\,left}$ or $\boldsymbol{p}_{i\,right}$ respectively. Because this method requires

knowledge of the depth value assigned to pixel $\boldsymbol{p}_i$, it cannot be applied to IPDBP algorithm in which depth information of the coded view is not utilized.

All of the above variants of the filling procedure present similar computational complexity. Each variant requires a search operation to determine the neighboring pixels $\boldsymbol{p}_{i\,left}$ and $\boldsymbol{p}_{i\,right}$ for each unassigned pixel $\boldsymbol{p}_i$. Based on the values of depth assigned to these neighboring pixels, a simple comparison operation is performed. Complexity of this operation depends on selected variant of the algorithm, but does not differ substantially. The most demanding step is obviously related to the search operation, which is performed along the horizontal direction, both left and right. However, as all of the pixels on the same line of an image, belonging to the same unassigned area share the same left and right neighbors $\boldsymbol{p}_{i\,left}$ and $\boldsymbol{p}_{i\,right}$ from which the corresponding pixel position is derived, the number of performed search operations can be reduced. As a result, the proposed filling algorithms are characterized by low computational complexity, which makes them especially useful in practical implementations, including applications for mobile end-user terminals.

A more advanced approaches for occluded area filling for texture images can be found in literature. These methods focus mainly on filling holes in texture images and are commonly known as the *inpainting* algorithms [Bert00]. The basic idea used in inpainting algorithms is to interpolate texture values of the missing pixels based on the values assigned to pixels surrounding them. The fill-in of the missing region is done by prolonging the isophote lines arriving at the boundaries of the region and completing them inside the region. Additionally, the angle of the isophotes is maintained. This way, in each iteration, the algorithm computes missing values for pixels lying closest to the boundaries of the region, until the whole gap is filled in.

Unfortunately, this approach is not suitable for the purpose of inter-view prediction of e.g. motion information, as it is, in fact, a form of more sophisticated interpolation algorithm. As already discussed in Section 3.2.1, utilization of interpolation methods in case of motion field should be avoided. Furthermore, the inpainting algorithms usually present a significant computational complexity, which excludes them from the scope of considered applications.

Chapter 4

Motivation for inter-view motion information prediction

4.1. Introduction

The depth-based inter-view prediction algorithms proposed in Chapter 3 are designed especially for motion information prediction. However, the proposed FPDBP and IPDBP algorithms may be also adopted for inter-view prediction of other syntax elements. In both methods, we assume that each block in analyzed image is encoded using the data derived independently for every pixel of the block, based on the data assigned to corresponding pixel in the reference view. No restrictions on specific types of data are introduced. Consequently, the block can be encoded with various types of data derived from the reference view, including control data, e.g. information about block partitioning or block coding mode, but also transform coefficients and motion information. In the following sections, the arguments for utilization of the proposed algorithms for inter-view prediction of motion information are presented.

4.2. Bitstream structure in advanced video coders

As already mentioned in Section 2.1.1, the output bitstream of the typical modern hybrid video encoder contains three main types of data: control data (prediction modes, block partitioning, etc.), transform coefficients (quantized transform coefficients) and motion information (motion vectors,

reference frame indices). In this section, experimental results showing the participation of the abovementioned components of the visual streams in case of the multiview sequences are presented.

The most significant part of the bitstream produced by a typical hybrid video coder for a single-view video sequence is used to represent transform coefficients [Dom98]. However, this situation changes when the multiview video coding is considered. Tab. 4.1 presents experimental results obtained with MVC codec [Chen09] for the complete set of multiview test sequences (see Annex A.1) using 2-view codec setup. The results are averaged over all test sequences. Detailed results for individual test sequences are presented in Annex C.1. Configuration of MVC coder used in the experiment is described in Annex B.1. The entropy coder utilized in the experiment is Context-Adaptive Binary Arithmetic Coding (CABAC). However, it is extremely difficult to determine the number of bits representing individual syntax elements in the bitstream produced by the CABAC entropy coder. Consequently, the number of bits for individual syntax elements is calculated in the experiment by Context-Adaptive Variable Length Coding (CAVLC), used in MVC codec for rate-optimization purposes. As a result, percentage values presented in Tab. 4.1 may differ from the exact values for the generated output bitstreams, nevertheless, the proportion is very similar.

Tab. 4.1. Bitstream structure for MVC, averaged over all test sequences.

Base view [%]				
Syntax element	**QP**			
	22	**27**	**32**	**37**
Control data	9.9	15.3	22.1	29.9
Transform coefficients	76.8	69.7	62.2	55.1
Motion information	13.3	15.0	15.7	14.9
Side view [%]				
Syntax element	**QP**			
	22	**27**	**32**	**37**
Control data	12.6	21.8	33.7	46.1
Transform coefficients	66.3	49.0	32.1	19.8
Motion information	21.1	29.1	34.2	34.1

As shown in Tab. 4.1 for the base view, transform coefficients are the dominant part of the bitstream for every analyzed value of the quantization parameter (QP). Nevertheless, their share in the bitstream decreases for lower bitrates (larger QP values). At the same time, the share in the bitstream does not change significantly for motion information, with the value below 16 [%] of the bitstream. The part of the bitstream representing control data remains relatively constant for wide range of bitrates. Consequently, its share in the bitstream grows with decreasing bitrate. On the

other hand, in case of the side view of a multiview sequence, the share in the bitstream for transform coefficients is much smaller when compared to the base view. Also, it decreases noticeably, falling below 20 [%] for lower bitrates (QP=37). This is obviously a consequence of the inter-view prediction with multiview disparity compensation used by MVC codec. Simultaneously, share in the bitstream for motion information increases, exceeding the part of the bitstream representing transform coefficients for QP≥32. As a result, motion information, together with control data, becomes a dominant part of the bitstream for side views in multiview sequence, especially when lower bitrates are considered.

Similar tendency can be observed for HEVC video codec, in which the emphasis was mainly placed on more advanced and accurate prediction methods in order to decrease the energy of prediction error and, consequently, reduce the number of transform coefficients to be encoded in the bitstream. This results in larger share in the bitstream representing motion information and control data when compared to coding a video sequence using AVC-based codec with comparable quality of the decompressed video.

4.3. Accuracy of inter-view prediction of motion information

In this section, the possibility of utilizing the similarity between motion fields of the neighboring views of a multiview sequence is discussed. As already stated in Section 2.5, if the distance between neighboring views is small, the content of the video sequences obtained from these views and, consequently, their motion fields are highly correlated [Guo06, Feck05, Merk07, Su06]. However, in order to examine how accurately the motion information can be actually predicted from the neighboring view at the same time instant using the proposed depth-based algorithms, the following experiment is conducted.

The proposed depth-based inter-view prediction algorithms: FPDBP and IPDBP (see Section 3.2) are implemented in MVC reference codec and utilized for motion information coding, as described in Section 5.1. Next, a complete set of 8 multiview test sequences (see Annex A.1) is encoded using the 2-view codec setup (refer to codec configuration in Annex B.1) for values of QP={22,27,32,37} [Su06, Tan08]. In the experiment, the proposed prediction algorithms are utilized in B-frames of the side views only. Consequently, motion information, i.e. motion vectors

and reference picture indices, selected in the coding process for such frames are analyzed. For each picture point of encoded sequences, selected motion vectors are compared with motion vectors predicted using each of the considered inter-view prediction algorithms to assess their prediction accuracy. The evaluation is performed using the accuracy measures of motion field prediction described in Section 2.3: Pearson Correlation Coefficient (PCC) and Vector Similarity Measure (VSIM). In case of PCC, accuracy of motion field prediction is measured separately for horizontal and vertical components of motion vectors. Consequently, values for horizontal (PCCx) and vertical (PCCy) coefficients are presented (refer to Section 2.3). Additionally, an averaged value of the two abovementioned coefficients (PCCavg) is also calculated. The results for both available reference lists of MVC codec, averaged over QP values, are presented in Tab. 4.2. For detailed results obtained for individual QP values, please refer to Annex C.2. The accuracy measures are calculated only for the picture points for which predicted reference picture index is consistent with the one selected during the coding process. The percentage of such points for each predictor is indicated by the "% of points" column in Tab. 4.2.

Tab. 4.2. Accuracy measures for motion information prediction using median, IPDBP and FPDBP predictors, averaged over QP={22,27,32,37}.

	Predictor	median					IPDBP					FPDBP				
Reference list 0	Sequence	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM
	Poznan Street	85	0.96	0.91	0.93	0.97	89	0.96	0.93	0.94	0.97	89	0.96	0.93	0.95	0.97
	Poznan Hall2	84	0.97	0.80	0.88	0.97	90	0.94	0.84	0.89	0.95	90	0.94	0.84	0.89	0.96
	GT Fly	43	0.97	0.96	0.97	0.95	79	0.97	0.97	0.97	0.97	79	0.98	0.97	0.97	0.97
	Dancer	69	0.92	0.80	0.86	0.94	83	0.98	0.83	0.91	0.96	83	0.98	0.83	0.91	0.96
	Kendo	77	0.93	0.77	0.85	0.94	87	0.95	0.88	0.91	0.95	87	0.95	0.89	0.92	0.96
	Baloons	75	0.91	0.92	0.91	0.93	89	0.95	0.96	0.96	0.95	89	0.96	0.96	0.96	0.95
	Lovebird1	85	0.83	0.80	0.82	0.98	93	0.87	0.84	0.85	0.98	93	0.87	0.84	0.86	0.98
	Newspaper	86	0.90	0.79	0.85	0.95	92	0.97	0.88	0.92	0.97	92	0.98	0.89	0.93	0.97
	Average	72	0.95	0.85	0.90	0.95	86	0.96	0.89	0.93	0.96	86	0.96	0.89	0.93	0.96
Reference list 1	Poznan Street	88	0.97	0.92	0.94	0.97	91	0.96	0.92	0.94	0.97	91	0.96	0.93	0.94	0.97
	Poznan Hall2	89	0.97	0.78	0.87	0.97	92	0.92	0.77	0.85	0.94	92	0.92	0.78	0.85	0.94
	GT Fly	48	0.96	0.95	0.96	0.95	87	0.98	0.97	0.97	0.97	87	0.98	0.97	0.97	0.97
	Dancer	71	0.96	0.79	0.88	0.95	88	0.97	0.82	0.90	0.95	87	0.97	0.82	0.89	0.95
	Kendo	83	0.92	0.74	0.83	0.92	91	0.94	0.84	0.89	0.93	91	0.94	0.84	0.89	0.94
	Baloons	82	0.93	0.91	0.92	0.92	92	0.95	0.95	0.95	0.94	92	0.95	0.95	0.95	0.94
	Lovebird1	90	0.86	0.79	0.82	0.98	94	0.84	0.80	0.82	0.98	94	0.84	0.80	0.82	0.98
	Newspaper	89	0.95	0.75	0.85	0.96	94	0.96	0.81	0.88	0.96	94	0.97	0.86	0.92	0.96
	Average	76	0.95	0.84	0.90	0.95	90	0.95	0.86	0.91	0.95	90	0.95	0.87	0.91	0.95

The prediction accuracy measured for FPDBP and IPDBP algorithms are also compared with accuracy of the standard median predictor of AVC, which is the fundamental predictor used in MVC (see Tab. 4.2). Motion vectors resulting from the median prediction are compared with motion vectors selected by the reference MVC codec [Chen09] with the same 2-view configuration (see Annex B.1). For prediction of reference picture indices, a procedure used in standard Direct mode, the most efficient inter-prediction mode of AVC, is utilized.

The analysis of Tab. 4.2 shows that proposed depth-based inter-view prediction algorithms (FPDBP and IPDBP) present higher accuracy of reference picture index prediction than the one used in standard Direct mode of MVC (refer to column "% of points" for median, IPDBP and FPDBP predictors). The average difference is almost 14 [pp] in favor of the proposed inter-view predictors, and is especially noticeable for sequences with complex, non-translational motion, e.g. *GT Fly*. Additionally, the accuracy measures of motion field prediction for FPDBP and IPDBP algorithms are usually similar or higher than for median predictor. This is observed for both of the adopted measures: PCC and VSIM. The results averaged over all inspected QP values and the whole set of test sequences show that proposed inter-view prediction algorithms achieve values of PCCavg=0.93 and VSIM=0.96 for reference list 0 and PCCavg=0.91 and VSIM=0.95 for reference list 1 against PCCavg=0.90 and VSIM=0.95 observed for the median predictor in case of both reference lists.

Analysis of the accuracy measures for individual QP values (see Annex C.2) shows that prediction accuracy grows for lower bitrates regardless of the analyzed predictor. The reason for such situation is the fact that motion vectors assigned to neighboring regions of encoded pictures differ slightly due to predictive coding if the bit budget is limited. Also, motion fields of the neighboring views are more similar in such a case.

Experimental results obtained for individual test sequences indicate also the importance of accuracy of depth information utilized in depth-based inter-view prediction. In case of test sequences with complex motion, for which the ground-truth depth information is not available, motion field obtained using the proposed predictors may be slightly less accurate than for median predictor. Such situation is observed e.g. for the test sequence *Poznan Hall2*.

4.4. Conclusions

Analyses of bitstream structure presented in Section 4.2 show clearly that development of more efficient ways of representing motion information is expedient, as motion information becomes more and more important part of the bitstream produced by the modern hybrid video codecs. This applies also to the multiview video coding, where inter-view prediction of texture is currently much more advanced than motion information prediction. Therefore, improvement in this field is desirable.

Additionally, based on the results presented in Section 4.3, it can be concluded that the proposed depth-based inter-view prediction algorithms exhibit a potential for accurate prediction of motion information. Both FPDBP and IPDBP algorithms present similar performance and, in many cases, can predict motion information more accurately than the state-of-the-art median predictor of AVC and MVC.

As a result, the author of this thesis decided to utilize the proposed depth-based inter-view prediction algorithms for motion information prediction in order to reduce the amount of motion information transmitted in the bitstream and, consequently, increase the coding efficiency of the multiview video codec. With regards to bitstream structure analyses presented above, it can be expected that achieved results should be especially evident for lower bitrates, which, in turn, are of particular importance for emerging and future multiview video codecs [MP11b]. Furthermore, the proposed inter-view prediction algorithms are designed to reduce the need of dividing larger image blocks into smaller units, due to utilization of independent point-to-point prediction. As a result, the number of syntax elements used for signaling the block partitioning in the bitstream should be also reduced, decreasing the amount of control data to be transmitted. Consequently, utilization of the proposed algorithms should increase coding efficiency of the state-of-the-art multiview video codecs, which is further investigated in this dissertation.

Chapter 5

Application of proposed depth-based inter-view prediction methods in motion information coding

5.1. Proposed approach

In this section, utilization of the proposed depth-based inter-view prediction methods in advanced multiview video codecs is discussed. As previously stated, the proposed FPDBP and IPDBP algorithms, introduced in Section 3.2, exhibit a potential to increase the coding efficiency of the multiview video codecs, when adopted for prediction of motion information. Here, methods for utilization of the proposed prediction algorithms for efficient motion information coding in MVC and MV-HEVC multiview video codecs are presented.

As described in Section 3.2, the purpose of the proposed FPDBP and IPDBP algorithms is to find a mapping between coordinates of each point of the coded image with corresponding point in the reference image. Consequently, for every non-base view of encoded multiview sequence, motion information, including motion vectors and reference picture indices, can be predicted from available, already encoded reference views. In the proposed approach, if coder chooses the inter-view predicted motion information as the best option in the rate-distortion optimization process, the predicted motion vectors and reference picture indices are independently derived from the reference view at the same time instant and assigned to each point of encoded block for the purpose of motion

compensation (see Fig. 5.1). As a result, motion information from the reference view is simply reused without the need to retransmit it once again in the bitstream of the coded view. This results in improved compression efficiency of the multiview codec, if the correlation between motion fields of the coded and reference views is high.

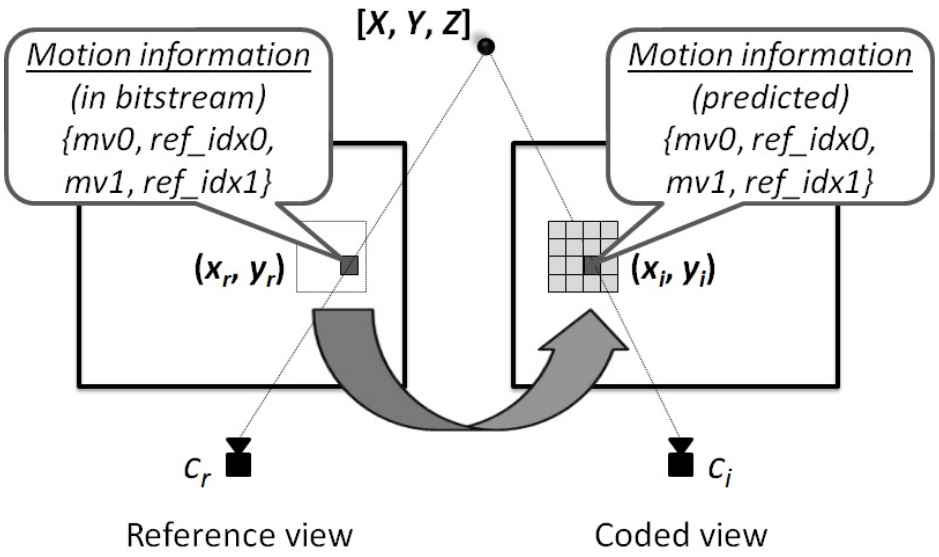


Fig. 5.1. Inter-view derivation of motion information (based on [Kon11]).

Obviously, utilization of the proposed method of motion information prediction for the encoded block must be somehow indicated to the decoder. In the following sections, strategies for signaling, together with details on modifications of syntax and structure of MVC and MV-HEVC advanced video codecs due to application of the proposed approach for motion information coding are presented.

5.2. Motion information coding in MVC

Based on the results presented in Section 4.3, proposed FPDBP and IPDBP predictors should be incorporated into MVC codec in a way that would utilize their potential for efficient motion information prediction as, in many cases, the proposed algorithms perform more efficiently than the median predictor of MVC. Following these observations, the FPDBP and IPDBP prediction algorithms have been adopted as the enhancement of the most effective motion information coding techniques in MVC, the standard Skip and Direct modes of AVC. Consequently, the cost of selecting the proposed depth-based inter-view prediction methods for motion information coding is

possibly low, which should result in maximizing the expected bitrate reduction and improve performance of the multiview video codec.

In the proposed approach, each of the introduced FPDBP and IPDBP prediction algorithms is utilized in the MVC codec one at a time. Following the concept of standard Skip and Direct modes, in which prediction residual signal is transmitted only for the non-skip-coded macroblocks, several variants of utilization of the proposed inter-view prediction algorithms are possible. Consequently, a macroblock coding mode which uses one of the proposed inter-view prediction algorithms for motion information prediction will be referred to as Inter-View Direct (IVD) mode if the prediction residual signal is transmitted for the macroblock and Inter-View Skip (IVS) mode in the opposite case. We will refer to all of these modes together as Inter-View (IV) modes. As a result, regarding the utilized inter-view prediction algorithm, the following new macroblock coding modes are introduced into MVC codec:

- Forward Projection Inter-View Direct (FPIVD),
- Forward Projection Inter-View Skip (FPIVS),
- Inverse Projection Inter-View Direct (IPIVD),
- Inverse Projection Inter-View Skip (IPIVS).

First two modes: FPIVD and FPIVS apply to the case when FPDBP prediction algorithm is used, whereas the latter two: IPIVD and IPIVS implement IPDBP prediction algorithm.

As a consequence, together with the standard macroblock coding modes of MVC, there are four macroblock coding modes for efficient representation of motion information available in the modified MVC codec: Skip, Direct, IVS and IVD. Obviously, in such situation, a number of possible strategies for efficient motion information coding exist. First, the question arises if all of these modes are necessary. On the other hand, there is also the question of what the best way of signaling these modes in a bitstream is. Both question are addressed in Section 6.3.1 and in one of the authors previous works [Kon11a]. Fig. 5.2 presents the possible mode selection strategies and shows necessary syntax modifications to implement the codec [Kon11a].

In Fig. 5.2, eight different variants of syntax are presented. The traditional AVC and MVC codecs use syntax *variant 1*, which is described in detail in Section 2.1.2: Skip mode is signaled with a *skip_flag* only. Direct mode is encoded with a full macroblock header including *mb_type* specific for Direct mode. IVS and IVD modes are not available in *variant 1*, however, they are utilized in all further syntax variants.

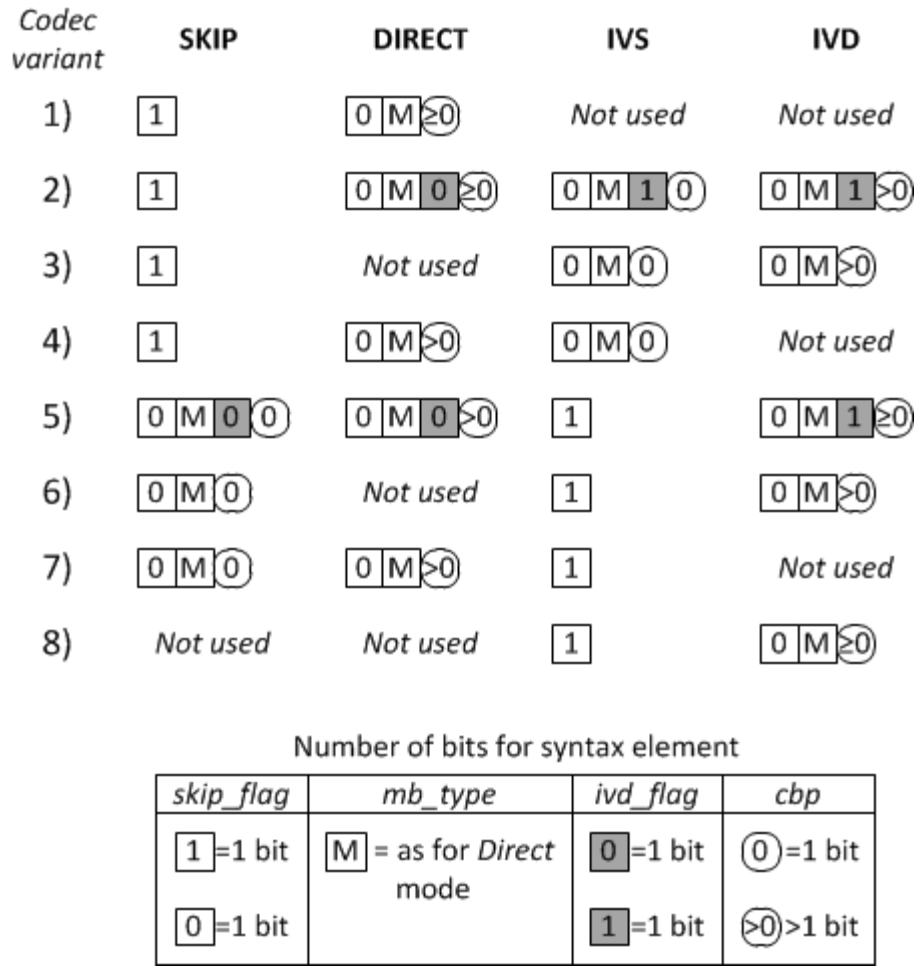


Fig. 5.2. Syntax variants for MVC (based on [Kon11a]).

In *variant 2*, the idea is to modify the existing Direct mode macroblock layer syntax by adding an extra 1-bit flag representing a new syntax element, i.e. *ivd_flag*, when *mb_type* is signaling the Direct mode selection. This bit enables the codec to distinguish IVS and IVD modes from the traditional Direct mode. IVS and IVD modes are discriminated by the value of coded block pattern (*cbp*) element, which indicates if the prediction residual signal is transmitted for the macroblock.

Next two syntax variants (*variant 3* and *4*) are motivated by the observation that Direct and IVD modes - the modes with non-zero prediction residual signal, are selected rarely in encoding process, especially for lower bitrates. If we disable one of these modes, the prediction error will increase for some macroblocks, however, a coding gain may be achieved due to not sending the additional 1-bit flag *ivd_flag*. In *variant 3* Direct mode is not used. Consequently, *ivd_flag* is not necessary to distinguish it from IVS and IVD modes. On the other hand, in *variant 4* IVD mode is disabled and IVS mode is distinguished from Direct mode by testing if *cbp* is set to zero. In this case, *ivd_flag* is also redundant.

As mentioned in Section 4.3, the proposed inter-view prediction algorithms can perform more efficiently than traditional median predictor utilized in Direct and Skip modes of MVC. In order to verify this observation, IVS mode have been encoded with syntax of the Skip mode, resulting in syntax *variants 5-7* (corresponding to *variants 2-4*). Here, IVS mode is signaled with *skip_flag*=1 and other modes are encoded with a full macroblock header including *mb_type* specific for Direct mode. The last syntax variant, *variant 8*, have been proposed to check if the traditional Direct and Skip modes can be completely substituted by the IVD and IVS modes.

For all of the abovementioned syntax variants, the proposed inter-view prediction is disabled in the base view and in all anchor pictures of the side views (see Fig. 5.3). This results from the basic concept of presented inter-view prediction algorithms which reuse motion information from different view and, hence, cannot be applied if no reference view or motion information referring to other time instant are available. This means that the base view remains compatible with AVC and the anchor picture syntax is not modified. Consequently, possible selection of the IVD mode is signaled only in non-anchor pictures of the side views.

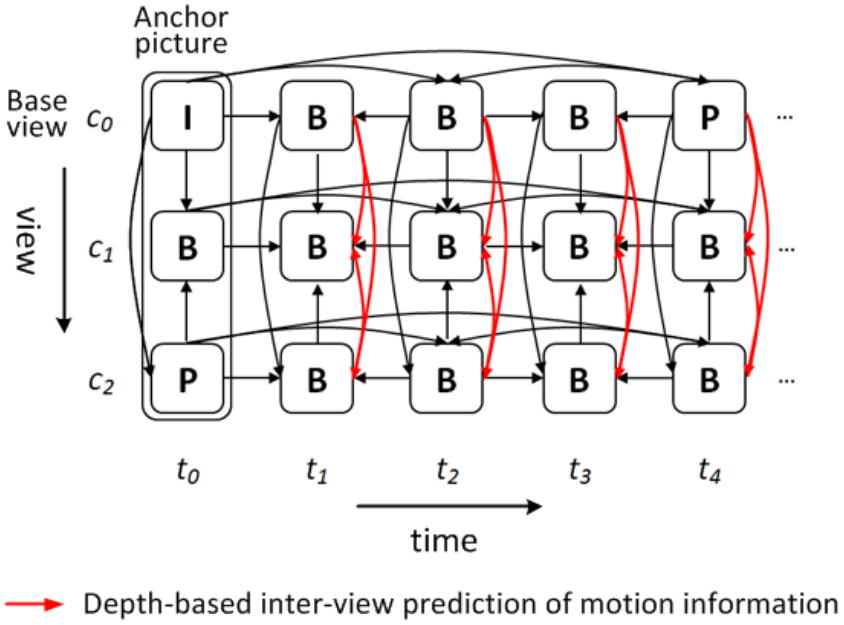


Fig. 5.3. Depth-based inter-view prediction of motion information in exemplary MVC prediction scheme.

When the IVS or IVD mode is selected by the coder in the rate-distortion optimization process, every pixel of the current macroblock derives motion vectors and reference picture indices from a corresponding pixel in the reference view. However, if no motion information can be derived for some pixels of the current macroblock, motion information obtained for this macroblock with the standard spatial predictor available in the codec is used for these pixels (refer to algorithm

description in Section 3.2). In case of MVC codec, the standard Direct mode prediction with median predictor is applied. No motion vectors or reference picture indices are transmitted for the macroblock in IVS or IVD mode. The only additional information signaled in the bitstream is the *ivd_flag* and, in case of IVD mode, the quantized prediction residual signal, which is encoded as in the traditional Direct mode of AVC. In presented implementation of the IVS and IVD modes, no dedicated context model for the *ivd_flag* has been employed in the CABAC encoder. The *ivd_flag* uses context model similar to *skip_flag*. As a consequence, the encoding process of *ivd_flag* is sub-optimal, thus a field for some further improvement exists.

Utilization of the proposed depth-based inter-view prediction algorithms requires availability of depth information for coded (FPDBP algorithm) and reference views (FPDBP and IPDBP algorithms). Originally, MVC codec does not provide any depth information during the coding or decoding process. As a consequence, the appropriate changes have been applied to the structure of MVC codec. In proposed solution, depth information in form of disparity maps is loaded into the codec from an external stream. In this way, the number of modifications introduced to the reference MVC model [Chen09] is reduced. The method of encoding depth information is independent of the MVC structure. Also, the quality of depth information provided to the codec can be flexibly modified without interference with rate-distortion control algorithm of the MVC codec.

Another important issue which refers to utilization of the proposed prediction algorithms in MVC codec is related to the usage of DIBR technique. As discussed in Section 2.6.4, intrinsic and extrinsic camera parameters are indispensable for DIBR. MVC provides these information in form of Supplemental Enhancement Information (SEI) messages called Multiview Acquisition Info SEI [Vet07]. However, in case the depth information is available in form of disparity maps, an additional information, namely the *z_near* and *z_far* coefficients, essential to convert disparity to depth [ISO10], need to be signaled to the decoder for each view. For the purpose of presented solution, this functionality have been implemented by the author [Kon10a, Kon10b] as the extension of Multiview Scene Info SEI message introduced originally in [Yea07] (see Fig. 5.4).

Fig. 5.4. Syntax modifications for camera and depth parameters (based on [Kon11]).

As presented above, the analyzed implementation of the IVS and IVD modes uses the proposed new macroblock modes only in 16×16-pixel macroblock partitioning scheme. This is dictated by the intention to compare the presented concept with the existing Motion Skip coding tool of JMVM, which is applied only to macroblocks of size 16×16 pixels. The 8×8-pixel and further partitioning has not been implemented, however, the possibility of utilizing the IVS and IVD modes in smaller blocks exists and may further improve the performance of multiview video codec.

5.3. Motion information coding in MV-HEVC

Utilization of the proposed FPDBP and IPDBP prediction algorithms in a new generation multiview HEVC-based video codec, named MV-HEVC (see Section 2.4.2), requires a different approach than the one presented previously for MVC. Representation of the proposed depth-based inter-view algorithms for motion information prediction as an extension of the classic Direct and Skip modes known from AVC is no longer possible due to substantial differences in motion-compensated prediction between AVC and HEVC. As the classic Direct and Skip modes do not exist in HEVC, the most efficient modes for encoding a Coding Unit (CU) are based on the concept of block merging, described in Section 2.1.3. All of the available motion information predictors of HEVC are arranged in form of an ordered list of merge candidates. Consequently, the most natural way of introducing the proposed depth-based inter-view motion information predictors into MV-HEVC is extending the existing merge candidate list of the block merging mode. This approach preserves the original structure of MV-HEVC and, also, does not change the number and cost of available CU modes. Additionally, the proposed depth-based inter-view predictors are easily adopted to encoding of wide range of CU sizes without extra syntax modifications.

Fig. 5.5 presents the original merge candidate list of HEVC and its exemplary modifications by adding a new depth-based predictor (*DPB*) at various positions. Similarly as for MVC, only one of the proposed depth-based inter-view motion information predictors is utilized in a single implementation of the MV-HEVC codec. Depending on the prediction algorithm selected for the implementation, *DBP* refers to FPDBP or IPDBP. The position on the candidate list determines the cost of candidate to be chosen for merging process. As a result, this provides a simple but efficient mechanism for manipulating the usage of the proposed FPDBP or IPDBP predictors and their impact on the coding efficiency.

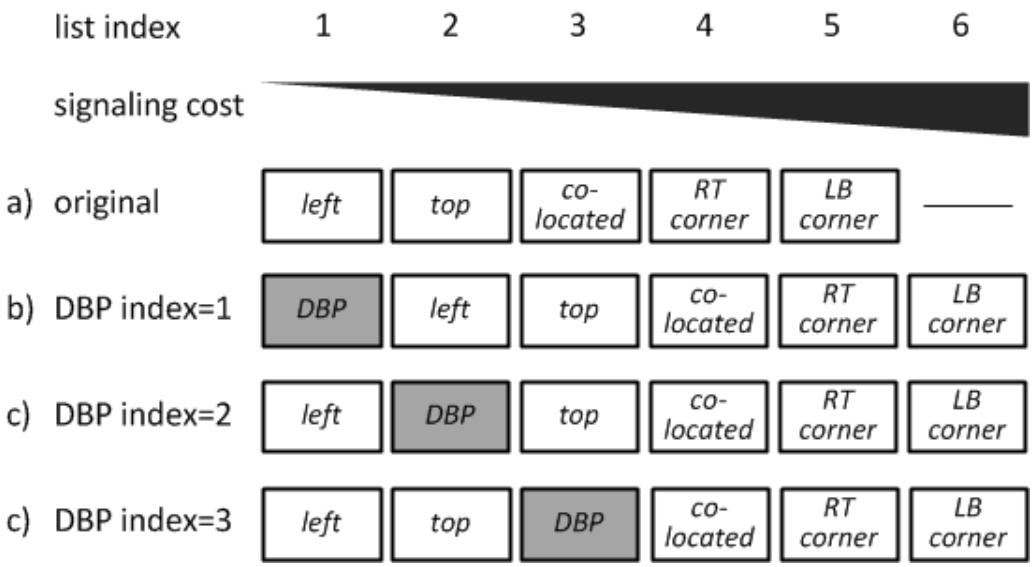


Fig. 5.5. Merge candidate list of MV-HEVC: a) original, b)-d) with additional depth-based predictor (DBP) placed at positions 1-3 respectively.

In the proposed approach, when the *DBP* merge candidate is selected by the coder in the rate-distortion optimization process, every pixel of the current CU derives motion vectors and reference picture indices from corresponding pixel in the reference view. However, if motion information for some of macroblock pixels cannot be derived this way, another predictor available in the codec is utilized to obtain motion information for these pixels (see algorithm description in Section 3.2). In the proposed solution for MV-HEVC, motion information calculated for current CU using the first available merge candidate predictor on the list other than *DBP* is selected. As shown in Fig. 5.5a, the possible merge candidate predictors are in turn: *left*, *top*, *co-located*, *right-top* and *left-bottom*. If no candidate other than *DBP* is available, zero motion vectors and indices of the first available reference picture on each reference picture list are selected to represent the motion information of the missing pixels.

In MV-HEVC, if a merge candidate is chosen to represent encoded CU, motion information for motion-compensated prediction of this CU is obtained from the candidate. As a consequence, no motion vectors or reference picture indices are transmitted in the bitstream for such CU. The only additional information signaled in the bitstream is the *merge_index* that indicates which of the available merge candidates was selected. This procedure is fully compatible with the proposed approach of depth-based inter-view motion information coding. Consequently, the only CU syntax modification is related to extending the range of allowed *merge_index* values. As discussed in Section 5.2 for MVC, the proposed inter-view prediction and all of the resulting syntax modifications are disabled in the base view and all anchor pictures of the side views.

Similarly as in MVC, depth information is required to utilize the proposed depth-based inter-view prediction algorithms in MV-HEVC. In particular, depth information must be available for coded (FPDBP algorithm) and reference views (FPDBP and IPDBP algorithms). In order to introduce such functionality into MV-HEVC codec, an implementation of MV-HEVC codec in which texture and depth information (in form of disparity maps) are encoded simultaneously by a pair of parallel coders has been used. This provides an elegant structure of the multiview codec in which depth information can be compressed with selected quality, but without interfering the encoding process of the texture. Also, this implementation of MV-HEVC provides a mechanism for efficient encoding of camera parameters together with *z_near* and *z_far* coefficients, designed by the author of this thesis. The proposed approach for coding of the abovementioned multiview parameters is an extension of the Multiview Acquisition Info and Multiview Scene Info SEI messages known from MVC (see Section 5.2). However, an additional prediction of parameters' values from neighboring views and time instances is introduced. The detailed description of this method can be found in [Dom11a].

Chapter 6

Experimental results and comparison with existing techniques

6.1. Comparison of proposed occlusion detection algorithms

In this section, performance of the proposed occlusion detection algorithms: *z-test* and *o-mask*, introduced in Section 3.3, has been investigated. These methods are utilized in FPDBP algorithm (see Section 3.2.1) to determine the overlapping pixels and check visibility of pixels from the coded view in the reference view. Similar procedure, called the pixel visibility check (*pvc*), is conducted in step 3) of IPDBP algorithm (Section 3.2.2), however, in this case, the projection of pixel coordinates is made from the reference view into the coded view. The performance of *pvc* algorithm is also investigated in this section. As a result of applying all of the abovementioned methods, the occluded area in the image plane of the coded view is detected. Pixels in the occluded area are not assigned to any corresponding pixel position in available reference views and their motion information have to predicted in other way. Unfortunately, due to inaccuracy and inter-view inconsistency of depth information or synthesis artifacts related to the problem of undefined pixels, unassigned pixels may also occur in not occluded areas of the image plane, which may affect efficiency of the proposed depth-based inter-view prediction algorithms.

In order to evaluate performance of the considered occlusion detection algorithms, the number of such unassigned pixels for each of the algorithms is investigated. Tab. 6.1 presents percentages of unassigned pixels determined by each of the analyzed algorithms, averaged over a complete set

of multiview test sequences (see Annex A.1) using 2-view setup. Detailed results for individual test sequences are presented in Annex C.3. To check the influence of depth quality on performance of the analyzed algorithms, depth information in form of disparity maps compressed with MVC and different quantization parameter values are used. In the experiment, values of {0,20,30,40,50} for depth quantization parameter (QD) are selected. This provides a wide range of QD values to be tested, with a special emphasis on lower bitrates, which results from the approach accepted by the MPEG to treat the depth as a kind of side information to be sent with bitrate much below the target bitrate for texture [MP11b, ISO11a]. QD value equal to zero indicates the usage of original, not compressed disparity maps.

Tab. 6.1. Unassigned pixels for different occlusion detection algorithms and QD values, averaged over all test sequences [% of picture area].

Algorithm	QD					
	0	20	30	40	50	Avg.
pvc	5.010	4.987	4.858	4.695	4.413	4.793
z-test	21.476	21.885	22.284	23.103	20.065	21.763
o-mask	4.803	4.800	4.708	4.675	4.623	4.722

As there are no ground-truth occlusion maps available for the analyzed test sequences, the accuracy of the considered occlusion detection algorithms cannot be objectively evaluated using an independent measure. In this case, the only way is to compare the results obtained by each algorithm and to assess subjectively if the areas determined by the algorithms match the occluded areas of the analyzed visual scenes.

Analysis of experimental results presented in Tab. 6.1 shows that *o-mask* and *pvc* algorithms perform very similarly – average difference between percentages of unassigned pixel areas for the algorithms is usually below 0.2 [pp]. These slight differences result from inconsistency of depth information available for coded and reference views in case of natural video and, consequently, the fact that projection from coded into reference view may sometimes produce different pairs of corresponding points than projection in the opposite direction - from reference into coded view. On the other hand, for synthetic sequences, i.e. *GT_Fly* and *Dancer*, the results are identical (refer to Annex C.3). Moreover, we can also observe similar and small influence of the depth quality on the performance of *o-mask* and *pvc* algorithms. The number of unassigned pixels do not change drastically with the growing QD value. Difference between percentages of unassigned pixels for individual sequences is less than 2.0 [pp] for analyzed range of QD values, but usually does not exceed 10 [%] of the value determined for QD=0 (see Annex C.3). The results show clearly that *o-*

mask and *pvc* algorithms present a practical potential in detection of occluded areas for video coding applications.

The results obtained for *z-test* algorithm differ significantly from the ones presented for *o-mask* and *pvc* algorithms. Values averaged over complete test set of multiview sequence, presented in Tab. 6.1, show that percentages of unassigned pixels determined with *z-test* are much larger. This is caused by the poor results obtained for sequences with inconsistent depth information between the views (Annex C.3). However, in case of synthetic test sequences, i.e. *GT_Fly* and *Dancer*, for uncompressed disparity maps (QD=0), the results are almost identical as for *o-mask* and *pvc* algorithms. Moreover, due to accepted margin of error E, utilized in *z-test* algorithm (see Eq. 3.5) to minimize the error introduced by the rounding operation conducted on disparity value, the determined area of unassigned pixels is even smaller than for concurrent algorithms, but the difference is very slight. Nevertheless, analysis of the experimental results shows clearly that *z-test* algorithm is much more sensitive for depth information inaccuracies and its inconsistency between the views. It is also much more sensitive for depth quality changes. As presented in Annex C.3, differences in size of unassigned pixel areas for individual test sequences due to depth quality changes may exceed even 10.0 [pp]. Consequently, *z-test* algorithm is not suitable for practical usage in occlusion detection applications with corrupted depth information, however, it can be successfully adopted to evaluate the accuracy and consistency of depth information between the views.

In Annex C.4, exemplary pictures used for subjective evaluation of the proposed occlusion detection algorithms are presented. Subjective assessment allows to determine that *o-mask* and *pvc* algorithms usually generate accurate occlusion areas with only a small number of false determined pixels. False detections are mainly caused by depth information inaccuracies for natural video due to imperfect depth estimation algorithms or, in much smaller degree, from synthesis artifacts related to undefined pixels, visible especially for *pvc* algorithm. The influence of depth quality is small and occluded regions are determined properly even for large QD values – for sequences with accurate depth (e.g. *Dancer*), the occluded area is incorrect only for QD=50. As already mentioned, differences between occluded areas detected by *o-mask* and *pvc* algorithms result from inter-view inconsistency of depth information. Consequently, they are visible only for sequences like *Kendo* (natural video with inconsistent and inaccurate depth) or when strongly compressed disparity maps are utilized (e.g. QD=50 for the *Dancer* sequence).

On contrary, subjective evaluation of third of the analyzed algorithms shows that the *z-test* algorithm produces poor results, especially for natural video. The occluded areas determined with z-

test are definitely too large, which is very undesirable regarding the application in the inter-view prediction. The main cause of such situation is the inter-view inconsistency of depth information, which results in incorrectly determined occluded areas even for original, uncompressed disparity maps.

Based on the result presented for the changing quality of disparity maps, we can observe two phenomena related to the influence of depth quality on the size of occluded area determined by the occlusion detection algorithms:

- decreasing depth quality results in smaller occluded area – this is caused by smoothing the depth/disparity values on the object borders,
- decreasing depth quality results in decreasing the accuracy and inter-view consistency of depth which, in turn, may lead to generation of larger occluded area.

As a consequence, the effect of decreasing the quality of disparity maps utilized in occlusion detection depends on which of the two abovementioned phenomena dominates for the analyzed sequence and QD value. A dependency from the 3D structure of the visual scene (e.g. range of disparity values represented in the scene), but also inter-view consistency and accuracy of original, uncompressed disparity maps has been observed. Experiments show also an initial growth of determined occluded area with decreasing depth quality, however, the size of occluded area decreases for the largest QD values. This occurs for all of the analyzed test sequences.

6.2. Comparison of proposed unassigned area filling algorithms

Algorithms for unassigned area filling are used in the proposed depth-based inter-view prediction algorithms FPDBP and IPDBP to resolve the problem of unassigned pixels in the coded view. Here, performance of the unassigned area filling algorithms: *FILLmax*, *FILLmin* and *FILLsim*, introduced in Section 3.4, is analyzed.

The proposed unassigned area filling algorithms are evaluated based on their influence on the coding efficiency of the multiview video codec in which depth-based inter-view prediction is applied as described in Section 5.2. In the experiment, MVC codec [Chen09] with syntax *variant 5* is used (see Section 5.2). This syntax variant specifies a signaling method for macroblock coding modes with depth-based inter-view prediction, called the Inter-View (IV) modes. The choice of

such syntax is a consequence of the experimental results obtained for all of the proposed syntax variants of MVC (refer to Section 6.3.1). In the selected syntax variant, usage of the IV modes is high. Consequently, the influence of the considered filling algorithms on the coding performance is clearly visible. Also, this syntax variant achieves the best coding gains, which makes it the most promising for practical usage. The experiment is conducted for 2-view MVC codec setup with configuration presented in Annex B.1, for values of texture quantization parameter (QP) equal to {22,27,32,37} [Su06, Tan08].

Due to the number of required tests, a reduced set of test sequences is used. *Poznan Street*, *GT Fly*, *Balloons* and *Newspaper* sequences have been selected to assure that both natural and synthetic sequences are included in reduced test set with the same proportion, and also, that all of the video resolutions from complete test set are represented.

For the purpose of evaluating the influence of depth quality on performance of the analyzed unassigned area filling algorithms, depth information in form of disparity maps compressed with MVC (refer to codec configuration in Annex B.1) and various quantization parameter values (QD) is used. In the experiment, QD values of {0,20,40} are selected. The choice of the QD values was dictated by desire to provide a wide range of QD values to be tested, including QD values for lower bitrates, but also requirement to minimize the number of experiments to be conducted. According to convention adopted in this thesis, the value of QD=0 indicates the usage of original, uncompressed disparity maps.

Tab. 6.2. Bitrate change (ΔBitrate [%]) vs. original MVC and usage of Inter-View modes (IV modes usage [%]) for different unassigned area filling algorithms, averaged over all considered test sequences (side view).

Prediction & occlusion detection algorithm	Filling algorithm	ΔBitrate [%]				IV modes usage [%]			
		QD							
		0	20	40	Avg.	0	20	40	Avg.
IPDBP	FILLno	-9.56	-9.47	-10.27	-9.77	68.87	68.78	69.19	68.95
	FILLmax	-15.98	-15.92	-15.73	-15.88	72.65	72.62	72.57	72.61
	FILLmin	-15.92	-15.86	-15.66	-15.81	72.63	72.60	72.55	72.59
FPDBP *z-test*	FILLno	-6.92	-5.89	-5.79	-6.20	66.20	65.24	64.75	65.40
	FILLmax	-12.25	-11.81	-10.91	-11.65	69.36	69.02	68.47	68.95
	FILLmin	-12.64	-12.30	-11.70	-12.21	69.71	69.43	69.04	69.39
	FILLsim	-12.74	-12.39	-11.48	-12.20	69.85	69.54	69.00	69.46
FPDBP *o-mask*	FILLno	-10.12	-9.85	-10.04	-10.00	69.16	68.97	68.95	69.03
	FILLmax	-15.77	-15.76	-15.49	-15.67	72.49	72.48	72.41	72.46
	FILLmin	-15.75	-15.74	-15.49	-15.66	72.49	72.49	72.41	72.46
	FILLsim	-15.75	-15.74	-15.49	-15.66	72.49	72.49	72.41	72.46

Tab. 6.2 shows experimental results obtained for MVC codec with different combinations of prediction, occlusion detection and unassigned area filling algorithms implemented. In the table, values of bitrate change against original MVC and usage of IV modes in analyzed codec are presented, both measured for the side view only. The values are averaged over all considered test sequences. For the results showing bitrate change for individual test sequences, please refer to Annex C.5. The bitrate change is calculated using Bjontegaard metrics (refer to Section 2.2.2) for QP={22,27,32,37}. The IV mode usage indicates the percentage of macroblocks that use one of the IV modes against all macroblocks. The values of IV mode usage are averaged over all considered QP values.

In case of MVC with IPDBP, only two of the proposed unassigned area filling algorithms can be applied: *FILLmax* and *FILLmin*. The reason is that depth information for coded view is not available and *FILLsim* algorithm cannot be used. As presented in Tab. 6.2, utilization of the proposed filling algorithms can reduce bitstream by almost 6.1 [pp] against compression gains achieved by a codec without any unassigned area filling algorithm applied, indicated by *FILLno*. It can be observed that both algorithms: *FILLmax* and *FILLmin* perform very similarly - the difference in bitstream reduction is less than 0.07 [pp] in favor of *FILLmax* algorithm. The usage of IV modes, averaged over all analyzed QD values is equal to 72.61 [%] for *FILLmax* and 72.59 [%] for *FILLmin*, and it is higher by almost 3.5 [pp] when compared to *FILLno*. Decreasing the quality of disparity maps used for depth-base inter-view prediction reduces the coding performance of the codec, however, the change is slight. When compared to the coding performance with uncompressed disparity maps (QD=0), the reduction of bitstream is less than 0.25 [pp] for both filling algorithms. The usage of IV modes also decreases insignificantly, by less than 0.09 [pp]. Consequently, it can be concluded that the influence of depth quality on performance of the codec with *FILLmax* or *FILLmin* unassigned area filling algorithms applied is small. On contrary, if no filling algorithm is utilized (*FILLno*), performance of the codec varies much more for changing depth quality - difference in bitstream reduction is larger than 0.7 [pp]. Also, due to considerably smaller size of the unassigned area detected for QD=40 (refer to Tab. 6.1), the codec performance increases in this case.

In FPDBP depth-based inter-view prediction algorithm, two different occlusion detection algorithms can be used (see Section 3.3). As a consequence, *FILLmax*, *FILLmin* and *FILLsim* unassigned area filling algorithms are analyzed for the cases of utilization of *z-test* or *o-mask* occlusion detection algorithms.

For FPDBP algorithm with *o-mask* occlusion detection, utilization of the proposed filling algorithms can reduce bitstream by almost 5.7 [pp] against *FILLno* (Tab. 6.2). All of the filling algorithms achieve very similar results – the difference between bitstream reduction for each of the analyzed filling algorithms is less than 0.02 [pp] in favor of *FILLmax* algorithm. The usage of IV modes, averaged over all analyzed QD values is equal to 72.46 [%] for *FILLmax*, *FILLmin* and *FILLsim*, and it is higher by almost 3.4 [pp] when compared to *FILLno*. Similarly as for IPDBP, a coding performance reduction for decreasing quality of disparity maps used for depth-base inter-view prediction is observed, but still, the change is slight. When compared to the coding performance with uncompressed disparity maps (QD=0), the reduction of bitstream is less than 0.29 [pp] for every of the analyzed filling algorithms. The usage of IV modes also decreases, however, the difference is insignificant - less than 0.08 [pp]. Consequently, the influence of depth quality on performance of the codec with *FILLmax*, *FILLmin* or *FILLsim* unassigned area filling algorithms applied is small. As for IPDBP, performance of the codec without any filling algorithm applied (*FILLno*) varies much more for changing depth quality, which is evident especially if we refer to the results obtained for single test sequences (see Annex C.5). Also in this case, the codec performance increases for QD=40 due to considerably smaller size of the detected unassigned area.

The results obtained for FPDBP algorithm with *z-test* occlusion detection (see Tab. 6.2) show that utilization of the proposed filling algorithms can reduce bitstream – the difference is almost 5.7 [pp] against *FILLno*. However, in this case, codec performance varies between filling algorithms. The results averaged over all of the considered test sequences and QD values (Tab. 6.2) show that difference between *FILLmin* and *FILLsim* algorithms is slight (less than 0.02 [pp]), while the bitstream reduction for *FILLmax* is almost 0.55 [pp] smaller. Nevertheless, careful analyses of the results obtained for individual test sequences (Annex C.5) show that the above order changes as it depends on accuracy and quality of depth utilized by the proposed algorithms. The usage of IV modes, averaged over all analyzed QD values is equal to 68.95 [%] for *FILLmax*, 69.39 [%] for *FILLmin* and 69.46 [%] for *FILLsim*, and it is higher by almost 3.5-4.1 [pp] when compared to *FILLno*. What is significant, the achieved bitstream reduction and IV mode usage (Tab. 6.2) are clearly smaller then for two previously discussed cases: IPDBP and FPDBP with *o-mask*. The reason is the improperly determined unassigned area which is much larger than the actual occluded area. This disadvantage of *z-test* algorithm was discussed in Section 6.1. Additionally, we can observe that, also in this case, decreasing the quality of disparity maps used for depth-base inter-view prediction reduces the coding performance of the codec. However, in contrast to IPDBP and FPDBP with *o-mask*, differences noted for FPDBP algorithm with *z-test* occlusion detection are

much more evident, exceeding 1.0 [pp] regardless of the analyzed filling algorithm. Similarly, the usage of IV modes varies much more due to depth quality changes when compared with two previously discussed cases – differences equal up to 0.9 [pp] (see Tab. 6.2).

In addition to the experimental results presented above, correlation of coordinates of the corresponding pixels calculated using original and compressed depth information is investigated. The purpose is to check the influence of quality of depth used for inter-view prediction on performance of the proposed unassigned area filling algorithms. In contrast to the previous experiment, in which results obtained for MVC codec are analyzed, utilization of the correlation coefficients provides results that are independent of the multiview codec implementation or even predicted syntax element. In the experiment, we check the correlation between C_{org} and C_{compr}, where C_{org} is a set of coordinates of pixels from the reference view, corresponding to pixels in the coded view and determined using original, uncompressed disparity maps and C_{compr} is the set of corresponding pixel coordinates determined with compressed disparity maps. Quantization parameter values QD from $\{20,30,40,50\}$ set are utilized to compress the disparity maps using MVC. Correlation coefficients are calculated separately for each coordinate of corresponding pixels according to Eq. 2.5. Consequently, pixels which remain unassigned after application of filling algorithms are not analyzed. The results presented in Annex C.6 show correlation coefficients calculated for the horizontal coordinates of corresponding pixels determined with the proposed depth-based inter–view prediction algorithms for different filling algorithms. Also, results for the case where filling algorithm is disabled (*FILLno*) are presented. The results are obtained for 2-view case (one reference view available) and averaged over a complete set of multiview test sequences (see Annex A.1). Vertical correlation coefficients are not presented as their values are equal to 1. This results from the fact that for properly rectified pictures, which is the case, projection between views does not change the image line.

Analyses of the results shown in Annex C.6 lead to similar conclusions as presented for the previous experiment. In particular, values of correlation coefficients calculated for each combination of depth-based inter-view prediction and occlusion detection algorithms differ very slightly between various filling algorithms. On the other hand, correlation for *FILLno* differs from the values calculated for the cases when filling algorithm is enabled and is usually higher. This is because pixels in unassigned area are not considered in the experiment. In fact, corresponding pixel coordinates determined for pixels in unassigned area by means of filling algorithms change the most for decreasing depth quality. Also, we can observe that values of correlation coefficients for FPDBP algorithm with *z-test* deviate visibly from results presented for other algorithms and are evidently

smaller. This confirms the conclusion that *z-test* algorithm is more sensitive to depth quality changes than two other occlusion detection algorithms (refer to Section 6.1), regardless of the filling algorithm utilized. On contrary, FPDBP with *o-mask* and IPDBP algorithms perform very similarly.

Results presented in this section show evidently that application of the discussed unassigned area filling algorithms improve compression performance of the multiview video codec which uses the proposed depth-based inter-view prediction. For all cases, we observe a significant bitstream reduction of approximately 5.5-6.0 [pp] on average when compared to the codec without filling algorithm applied, regardless of the prediction and occlusion detection algorithms utilized in the codec. The choice of the unassigned area filling algorithm is of lesser importance, as obtained results are usually similar for every of the analyzed filling algorithms. Consequently, motivation for applying more advanced approaches for unassigned area filling is small. However, the simple approach to search only the closest available neighbors located to the left and right from unassigned pixel may not always be successful. In some cases, the whole line of image may consist of unassigned pixels only. This was observed for *Kendo* test sequence and FPDBP with *z-test* occlusion detection algorithm. Obviously, the reason of such situation is a poor performance of *z-test* algorithm due to depth information inaccuracy and inter-view inconsistency for this sequence. Nevertheless, such cases may happen and the discussed limitation of the proposed filling algorithms should be noted.

Following the observations mentioned above, only one of the discussed unassigned area filling algorithms will be used in further analyses of the proposed depth-based inter-view prediction algorithms. In this way, the number of examined test cases will be significantly reduced without affecting the general results. *FILLmax* algorithm will be selected for that purpose, because it presents a slightly better results concerning the influence on the coding performance for most of the analyzed syntax variants and, also, it can be utilized for both FPDBP and IPDBP depth-based inter-view prediction algorithms.

6.3. Impact of proposed prediction methods on compression efficiency of MVC

6.3.1. Choice of optimal signaling strategy

In this section, performance of different syntax variants introduced in Section 5.2 is analyzed. These syntax variants are used for signaling utilization of the proposed depth-based inter-view prediction modes in MVC codec. As discussed in Section 5.2, due to introduction of new methods for efficient motion information representation, a number of possible strategies for motion information coding exist. As a consequence, the question of what the best way of signaling the proposed depth-based inter-view modes in a bitstream is should be answered. In the experiment, we have also checked if all of the proposed and existing modes for efficient motion representation in MVC codec are necessary.

Experimental results are obtained with MVC codec [Chen09] in which eight different syntax variants for signaling the proposed depth-based inter-view prediction modes have been implemented (refer to Fig. 5.2 in Section 5.2). In order to reduce the number of performed experiments, only IPDBP prediction algorithm is used. However, as presented in Section 6.3.2, the influence of both proposed depth-based inter-view prediction algorithms FPDBP and IPDBP on the compression efficiency of MVC codec is very similar. Consequently, conclusions drawn for IPDBP should also be true for FPDBP algorithm. Configuration of MVC coder is described in Annex B.1. 2-view codec setup is utilized. In the experiment, five standard multiview test sequences are used (see Tab. 6.3). The selected test sequences were the only sequences approved by MPEG at the time of experiment for which appropriate depth information was available. Due to large number of tests to be conducted, only first 96 frames of each sequence are encoded. As the proposed prediction methods are desired to increase compression efficiency especially for the lower bitrates (refer to Section 4.1), QP values equal to {27,30,33,36} are utilized. To provide depth information to the codec, original, uncompressed disparity maps are used.

Tab. 6.4 presents results for individual test sequences. The improvement in compression performance is measured for syntax variants *2-8* (codecs with proposed IV modes implemented) against *variant 1* (original MVC codec) using Bjontegaard metrics (Section 2.2.2). The measures show average changes of bitrate and luminance PSNR (PSNRY) of encoded side view. Bitrate required for transmission of depth information is not included, as it is assumed to be transmitted with the base layer for other purposes.

Tab. 6.3. Parameters of video sequences used for evaluation of different signaling strategies in MVC.

#	Sequence	Sequence name	Resolution	Frame-rate [FPS]	Number of frames used	Coded views (2-view case)	Source
1.	Champagne tower	Champagne tower	1280×960	29.41	96	39-41	[Tani08]
2.	Pantomime	Pantomime	1280×960	29.41	96	39-41	[Tani08]
3.	Book Arrival	Book Arrival	1024×768	16.67	96	10-8	[Feld08]
4.	Lovebird1	Lovebird1	1024×768	30	96	6-8	[Um08]
5.	Newspaper	Newspaper	1024×768	30	96	4-6	[Ho08]

Tab. 6.4. Performance of MVC codec with syntax *variants 2-8* compared to *variant 1* using Bjontegaard metrics (based on [Kon11a]).

QP={27,30,33,36}	ΔPSNRY [dB]							ΔBitrate [%]						
	Syntax variant													
Sequence	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Book Arrival	0.12	0.13	0.12	0.16	0.15	0.14	0.12	-3.6	-3.8	-3.6	-4.7	-4.4	-4.0	-3.4
Champagne tower	0.25	0.26	0.21	0.31	0.31	0.26	0.28	-5.6	-5.9	-4.8	-7.2	-7.1	-6.1	-6.5
Lovebird1	0.08	0.07	0.06	0.07	0.08	0.06	0.07	-2.5	-2.3	-2.1	-2.4	-2.4	-1.9	-2.4
Newspaper	0.14	0.15	0.15	0.17	0.15	0.15	0.11	-3.6	-3.8	-3.7	-4.3	-3.9	-3.7	-2.9
Pantomime	0.33	0.33	0.30	0.40	0.38	0.35	0.32	-7.7	-7.8	-7.0	-9.3	-9.0	-8.3	-7.5
Average	0.19	0.19	0.17	0.22	0.21	0.19	0.18	-4.6	-4.7	-4.2	-5.6	-5.3	-4.8	-4.5

The results show that modifications of the syntax and mode selection strategy change the compression performance of the codec. All of the proposed syntax variants in which depth-based inter-view prediction is utilize (*variants 2-8*) perform better than the original MVC codec (*variant 1*). Achieved bitrate reductions averaged over considered test sequences are between 4.2 and 5.6 [%]. The largest coding gains are observed for *variant 5* of MVC codec.

The reason of the improved compression efficiency is an increase in frequency of selecting the low-cost macroblock modes of MVC, i.e. Direct, Skip and the proposed IV modes (called IVD and IVS). Fig. 6.1 shows the low-cost mode usage for different syntax variants, averaged over all considered test sequences and QP values. As a result of introducing the proposed depth-based inter-view prediction of motion information, macroblock modes that do not require transmitting motion information in the bitstream are selected more frequently by the coder (difference in usage of low-cost modes is 1.7-5.3 [pp] on average). Consequently, compression performance of the codec is better.

Based on the above results, it can be concluded that the proposed depth-based inter-view prediction performs better than the traditional prediction used in Direct and Skip modes as far as bitrate reduction is concerned. Comparison between syntax variants *1-4* and their counterpart variants *5-8* shows clearly that the low-cost modes are used more frequently for the latter case, which leads to higher compression ratio (see the "All low-cost modes" columns in Fig. 6.1).

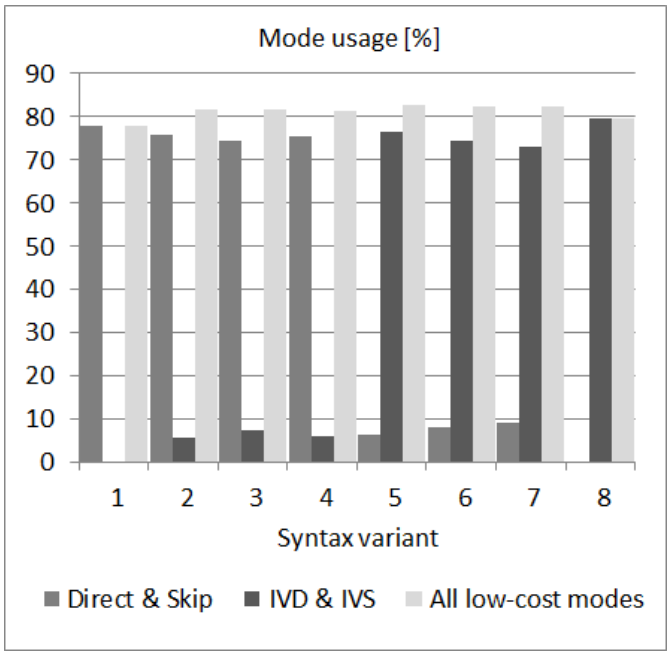


Fig. 6.1. Macroblock mode usage for traditional (Direct & Skip modes) and depth-based inter-view (IVD & IVS) motion information predictors for MVC codec with syntax variants 1-8, averaged over QP={27,30,33,36} and all considered test sequences (based on [Kon11a]).

Let us now refer to the issue of disabling Direct or IVD modes in MVC codec (refer to syntax variants *3*,*4*,*6* and *7*). In case of *variant 3*, a slight bitrate reduction of 0.1 [pp] against *variant 2* can be observed in Tab. 6.4. This results from more frequent usage of macroblocks with depth-based inter-view prediction due to lack of *ivd_flag* for signaling the IV modes. A gain of 1.5 [pp] is noted in Fig. 6.1. However, in *variant 4* we observe a loss of coding performance when compared to *variant 2*. In this case, due to the lack of efficient way of encoding complex motion with low-cost IVD mode, some of macroblocks had to be encoded with extra motion information resulting in higher bitrate. Also, in case of variants *6* and *7*, the gain from not transmitting additional *ivd_flag* is lower than the loss on larger prediction error when compared to *variant 5*. Analysis of the abovementioned syntax variants shows however, that it is always better to disable the Direct mode instead of IVD mode. Consequently, the conclusion is that the best mode selection strategy is to preserve all the macroblock modes to match various cases. Nevertheless, IVS mode should be set as the least expensive mode (*variant 5*).

As a consequence of the experimental result presented above, *variant 5* of the MVC codec has been selected to be used in further research on MVC as it achieves the best coding performance from all of the analyzed syntax variants.

6.3.2. Coding efficiency analysis

In this section, influence of the proposed depth-based inter-view prediction algorithms: FPDBP and IPDBP (see Section 3.2) on performance of MVC codec is investigated experimentally. Based on the conclusions presented in previous section, the syntax variant *5* of the MVC codec is analyzed (refer to Section 6.3.1). In both of the considered algorithms: FPDBP and IPDBP, unassigned area filling algorithm called *FILLmax* is used, as a consequence of experimental results presented in Section 6.2. Additionally, in case of FPDBP algorithm, the *o-mask* occlusion detection algorithm is utilized. As discussed in Section 6.1, this choice results from the fact that the second occlusion detection method introduced for FPDBP, namely *z-test*, is much less suitable for practical usage in occlusion detection applications due to its large sensitivity for inaccurate or inter-view inconsistent depth information. Unfortunately, such corrupted depth information may occur in case of multiview video coding, e.g. when depth information is provided in form of compressed disparity maps. Consequently, the variant of FPDBP algorithm using *z-test* for occlusion detection may perform much worse if utilized as a new prediction method in multiview video codec. An example of such situation can be observed in Tab. 6.2 (see Section 6.2) where performance of MVC codec for different unassigned area filling algorithms is discussed. Average bitstream reduction achieved for codecs with *z-test* algorithm applied is almost 2.5-3.0 [pp] worse than for the corresponding codec variants using *o-mask* algorithm, in case of utilization of original, uncompressed disparity maps (QD=0). For compressed disparity maps, the difference in bitstream reduction increases even to 4.5 [pp]. Similarly, the average usage of the proposed inter-view macroblock modes (IV modes) for all codecs with *z-test* algorithm is approximately 3-4 [pp] smaller than for the corresponding codec variants with *o-mask* algorithm.

In the experiment, MVC codecs with FPDBP or IPDBP algorithms implemented (denoted as JMVC+FPDBP and JMVC+IPDBP respectively) are compared with the original MVC codec [Chen09] (JMVC) and a reference codec model for MVC called the joint multiview model [MP08] with Motion Skip coding tool enabled (JMVM+MS). JMVM contains a number of additional multiview coding tools, including Motion Skip (Section 2.5), that were not included in the final MVC standard, but can achieve some additional coding gains compared to MVC (refer to Section 2.4.1). As a result, performance of the proposed depth-based inter-view prediction algorithms is

evaluated in comparison to the latest and most promising solutions of the state-of-the-art multiview video coding standard.

The experiment is conducted for 2-view and 3-view MVC codec setup with configuration presented in Annex B.1. JMVM+MS codec also uses the default configuration of MVC (Annex B.1), however, the Motion Skip coding tool must be additionally turned on, as it is disabled by default. Performance of the analyzed codecs is tested for the complete set of multiview test sequences (see Annex A.1) and QP values equal to $\{22,27,32,37\}$ [Su06, Tan08]. Depth information for JMVC+FPDBP and JMVC+IPDBP codecs is provided in form of original, uncompressed disparity maps to make the obtained results independent of the influence of additional depth compression artifacts (the effect of depth compression is analyzed separately in Section 6.3.3).

The results are presented for side views only. In the 2-view case, only one side view is available. According to the view coding order in MVC configuration (Annex B.1), this view is denoted as *view 1*. On the other hand, for 3-view camera setup, more than one view coding order is applicable. In the experiment, a typical view coding order presented in Fig. 2.10b is utilized. Consequently, according to Annex B.1, the analyzed side views are denoted as *view 1* for the central view (c_1 in Fig. 2.10b) and *view 2* for the outermost view (c_2 in Fig. 2.10b).

Tab. 6.5 and Tab. 6.6 show results obtained for individual test sequences encoded using 2-view and 3-view case respectively. The improvement in compression performance is measured for JMVC+FPDBP, JMVC+IPDBP and JMVM+MS codecs against JMVC using Bjontegaard metrics (Section 2.2.2). The measures indicate average changes of bitrate and luminance PSNR (PSNRY) of encoded side views. The depth information is assumed to be transmitted with the base layer for other purposes. Consequently bitrate required for transmission of depth information is not included in the results.

The results presented in Tab. 6.5 and Tab. 6.6 show that JMVC+FPDBP and JMVC+IPDBP codecs always perform better than JMVC. In case of JMVC+FPDBP, the average bitrate reduction for side view is equal to 14.5 [%] in 2-view case and 14.3 [%] for *view 1* and 7.7 [%] for *view 2* in 3-view case. Even better results are observed for JMVC+IPDBP: the average bitrate reduction is equal to 14.7 [%] in 2-view case and 15.2 [%] for *view 1* and 9.8 [%] for *view 2* in 3-view case. In contrast, the concurrent solution for inter-view motion prediction called Motion Skip, implemented in JMVM+MS codec, performs less effectively than the proposed methods. The average bitrate reductions achieved by JMVM+MS for side view are visibly smaller and equal to 6.8 [%] in 2-view case and 7.3 [%] for *view 1* and 5.6 [%] for *view 2* in 3-view case.

Tab. 6.5. Bjontegaard metrics for JMVC+FPDBP, JMVC+IPDBP and JMVM+MS vs. JMVC codec, calculated for side view with QP={22,27,32,37} (2-view case, *view 1*).

Sequence	ΔPSNRY [dB]			ΔBitrate [%]		
	JMVC +FPDBP	JMVC +IPDBP	JMVM +MS	JMVC +FPDBP	JMVC +IPDBP	JMVM +MS
Poznan Street	0.12	0.12	0.00	-6.2	-6.1	0.1
Poznan Hall2	0.34	0.34	0.22	-22.1	-22.4	-12.4
Dancer	0.39	0.41	0.09	-12.7	-13.3	-2.4
GT Fly	0.67	0.68	0.02	-22.8	-23.2	-0.6
Kendo	0.55	0.55	0.53	-15.4	-15.2	-13.0
Balloons	0.89	0.92	0.75	-22.8	-23.7	-17.6
Lovebird1	0.10	0.10	0.05	-2.8	-2.7	-1.3
Newspaper	0.40	0.39	0.28	-11.3	-11.0	-7.3
Avg.	0.43	0.44	0.24	-14.5	-14.7	-6.8

Tab. 6.6. Bjontegaard metrics for JMVC+FPDBP, JMVC+IPDBP and JMVM+MS vs. JMVC codec, calculated for side views with QP={22,27,32,37} (3-view case).

View 1 (central)						
Sequence	**ΔPSNRY [dB]**			**ΔBitrate [%]**		
	JMVC +FPDBP	**JMVC +IPDBP**	**JMVM +MS**	**JMVC +FPDBP**	**JMVC +IPDBP**	**JMVM +MS**
Poznan Street	0.18	0.17	-0.03	-8.2	-8.1	1.5
Poznan Hall2	0.40	0.40	0.30	-22.2	-22.3	-14.3
Dancer	0.43	0.44	0.12	-13.1	-13.8	-3.5
GT Fly	0.55	0.56	-0.03	-18.4	-18.5	0.5
Kendo	0.55	0.60	0.54	-15.5	-17.0	-13.1
Balloons	0.93	1.05	0.89	-23.2	-26.7	-20.0
Lovebird1	0.11	0.14	0.03	-3.0	-3.6	-0.7
Newspaper	0.42	0.44	0.35	-11.1	-11.6	-8.7
Avg.	0.45	0.48	0.27	-14.3	-15.2	-7.3
View 2 (outer)						
Sequence	**ΔPSNRY [dB]**			**ΔBitrate [%]**		
	JMVC +FPDBP	**JMVC +IPDBP**	**JMVM +MS**	**JMVC +FPDBP**	**JMVC +IPDBP**	**JMVM +MS**
Poznan Street	0.09	0.08	0.02	-4.0	-3.8	-0.8
Poznan Hall2	0.32	0.32	0.24	-17.2	-17.4	-11.4
Dancer	0.19	0.25	0.09	-5.9	-7.7	-2.4
GT Fly	0.51	0.52	0.22	-16.8	-17.1	-6.3
Kendo	0.21	0.37	0.38	-5.4	-9.8	-8.9
Balloons	0.40	0.68	0.63	-9.7	-16.4	-14.2
Lovebird1	0.02	0.05	-0.03	-0.5	-1.6	0.7
Newspaper	0.09	0.18	0.07	-2.2	-4.4	-1.6
Avg.	0.23	0.31	0.20	-7.7	-9.8	-5.6

Coding gains achieved by JMVC+FPDBP and JMVC+IPDBP codecs in 2-view case differ very slightly – difference in bitrate reduction of side view is equal to approximately 0.2 [pp] on average, and do not exceed 0.7 [pp] for individual test sequences. In fact, our experiments show that motion information is predicted for both of the considered algorithms from the same 4×4-pixel blocks for more than 75 [%] of pixels and from the same 16×16-pixel blocks for almost 90 [%] of pixels. Consequently, as the motion vectors for neighboring blocks are similar because of the predictive coding of the motion vector residuum, the motion field predicted by FPDBP and IPDBP algorithms differs slightly. In turn, this results in similar influence on compression performance of the codec. However, in 3-view case, the difference is more noticeable, especially for test sequences with inter-view inconsistent depth information, e.g. *Kendo*, *Balloons*, *Lovebird1* or *Newspaper*, and for encoding of the most outer view, i.e. *view 2*. In case of projection between distant views, the inter-view depth information inconsistency causes larger differences in the results of projection from left-to-right and right-to-left direction. Consequently, coordinates of the corresponding pixels in reference view calculated with FPDBP and IPDBP differ more significantly. This obviously results in larger differences in coding gains achieved by the two analyzed codecs – the average bitrate reductions for JMVC+FPDBP and JMVC+IPDBP codecs differ by 2.1 [pp] for *view 2*. In case of *view 1*, the differences between achieved coding gains are much smaller – the average bitrate reductions differ by 0.9 [pp]. Nevertheless, for individual test sequences, especially the ones with inter-view inconsistent depth information, the differences exceed 1.0 [pp] in some cases.

As presented in Fig. 6.2, coding gains achieved by the analyzed codecs are bigger for lower bitrates. The reason is that the number of bits saved by not sending motion vectors due to utilization of inter-view prediction increases slower than the number of bits for representing transform coefficients that must be transmitted for growing quality of the video.

The reason for better compression efficiency of JMVC+FPDBP and JMVC+IPDBP codecs is a significantly higher usage of macroblock modes that do not require transmission of motion information (low-cost modes) observed for these codecs. Fig. 6.3-6.5 show the usage of low-cost modes for all of the analyzed codecs. The results are averaged over all considered test sequences and QP values. For results showing low-cost modes usage for individual QP values, please refer to Annex C.7. In 2-view case, the average usage of all considered low-cost modes is higher by 10.3 and 10.4 [pp] in JMVC+FPDBP and JMVC+IPDBP respectively when compared with JMVC codec. Similarly, in 3-view case the values are: 10.4 and 10.6 [pp] for *view 1* and 8.4 and 9.4 [pp] for *view 2*. In contrast, the effect of higher usage of low-cost modes does not occur in JMVM+MS codec. The same tendency is observed for individual QP values (Annex C.7).

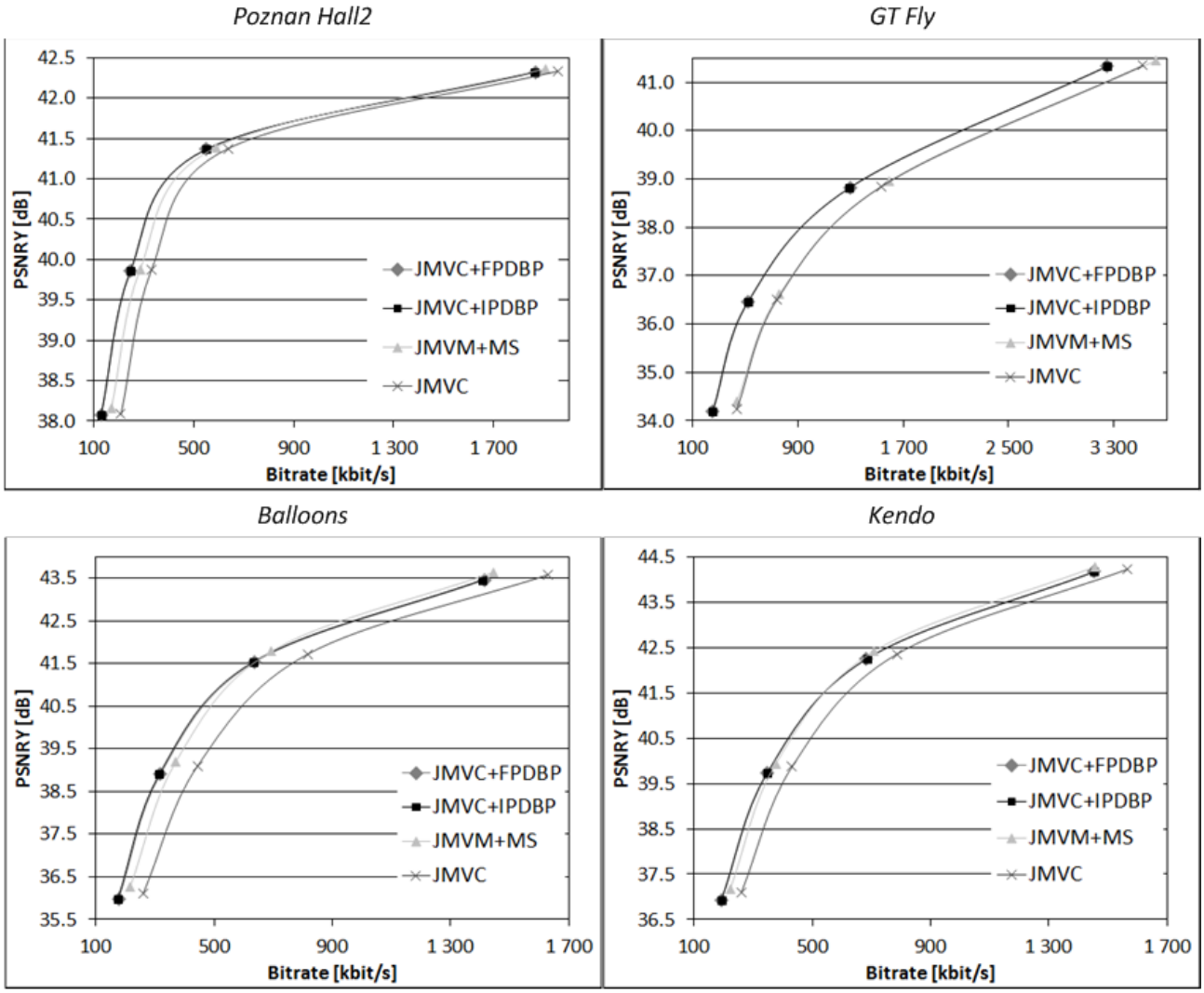


Fig. 6.2. Exemplary rate-distortion curves for JMVC+FPDBP, JMVC+IPDBP, JMVM+MS and JMVC codecs (2-view case, *view 1*).

The analysis of the number of bits representing individual syntax elements in the output bitstreams of considered multiview coders shows another reason for better performance of the proposed JMVC+FPDBP and JMVC+IPDBP codecs, when compared with the reference JMVC codec. In the experiment, number of bits representing control data, transform coefficients and motion information in side views is analyzed. All of the considered codecs use CABAC entropy coder, however, the number of bits representing individual syntax elements is determined based on CAVLC, utilized in MVC for rate-optimization purposes. As discussed in Section 4.1, a consequence of such approach is that presented percentage values may differ from the exact values for the generated output bitstreams, but the proportion is very similar.

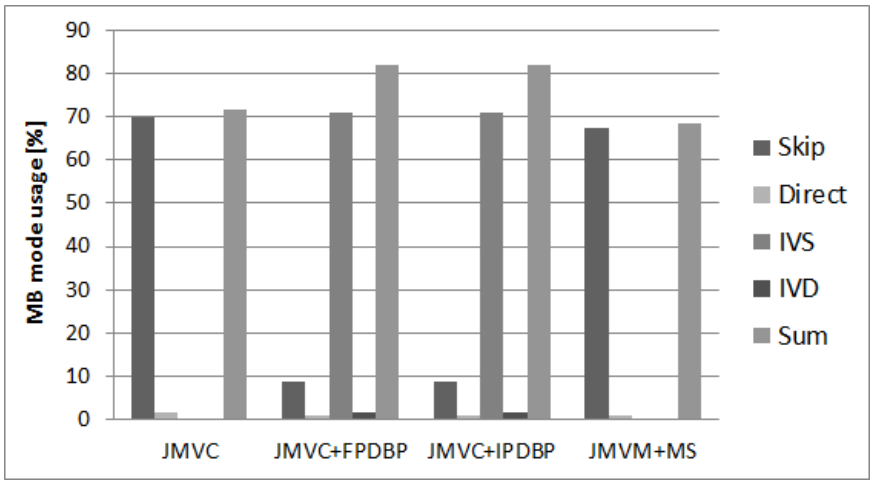


Fig. 6.3. Average usage of low-cost modes for different codecs, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

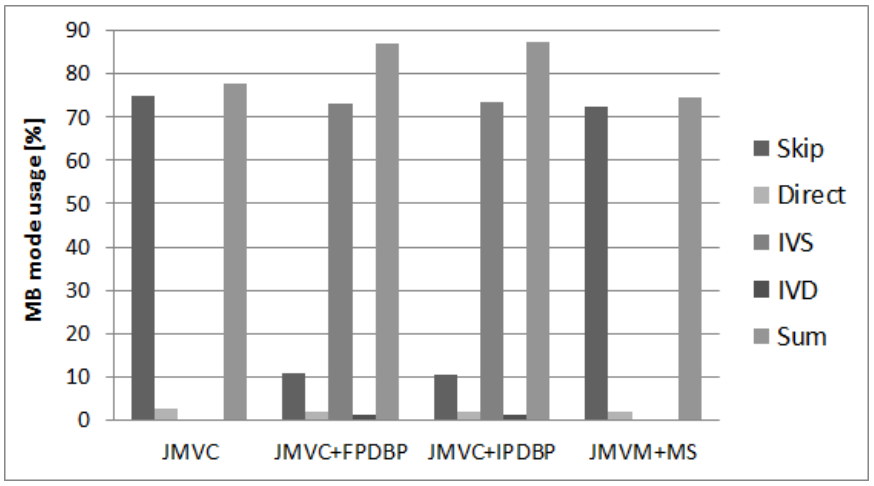


Fig. 6.4. Average usage of low-cost modes for different codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

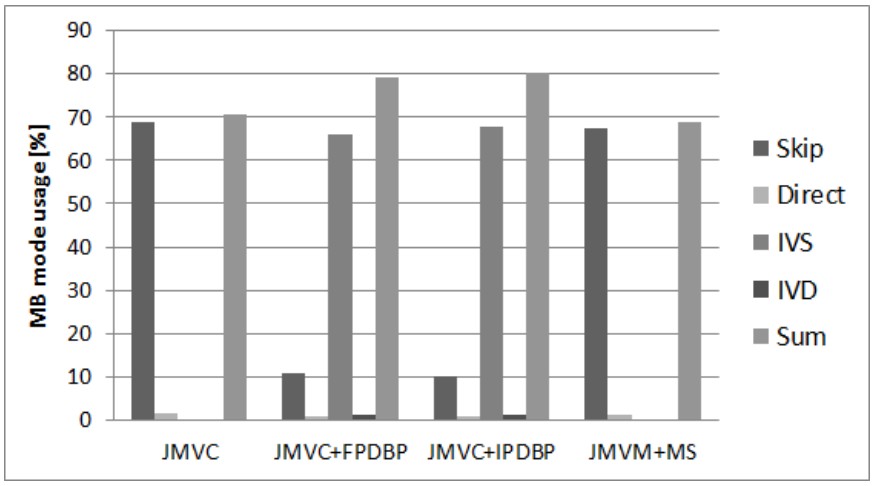


Fig. 6.5. Average usage of low-cost modes for different codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

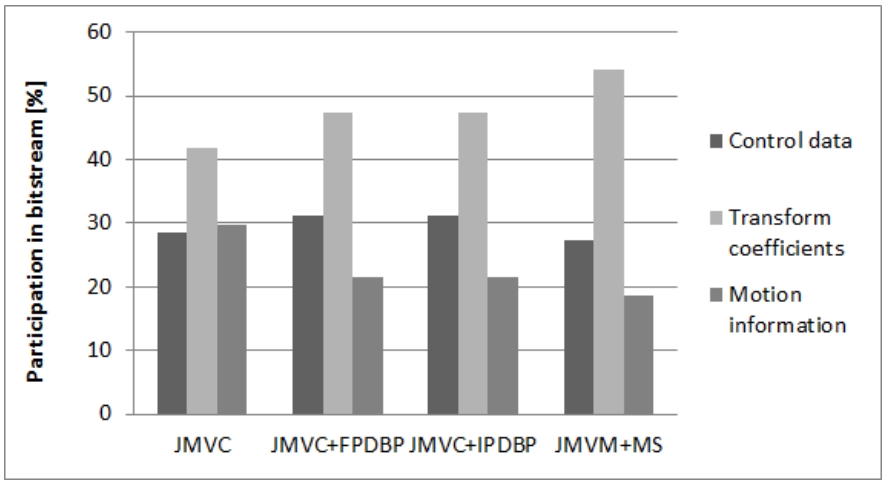


Fig. 6.6. Percentage participation in bitstream for each syntax element in different codecs, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

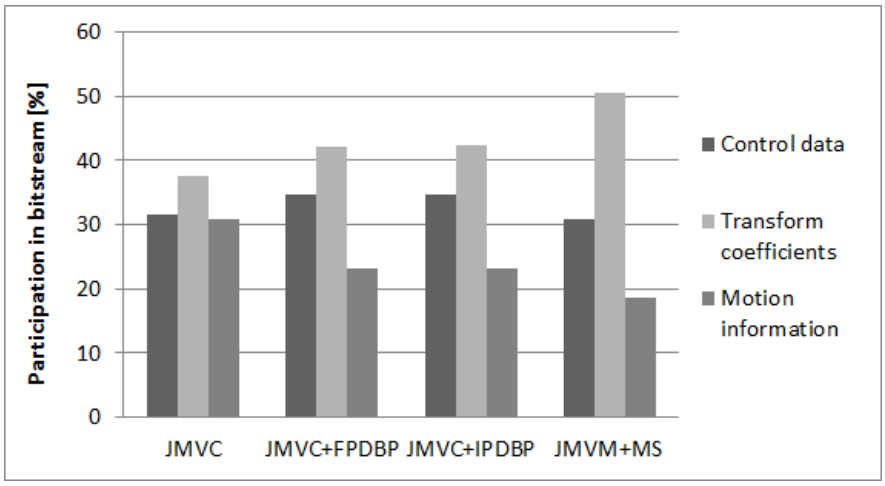


Fig. 6.7. Percentage participation in bitstream for each syntax element in different codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

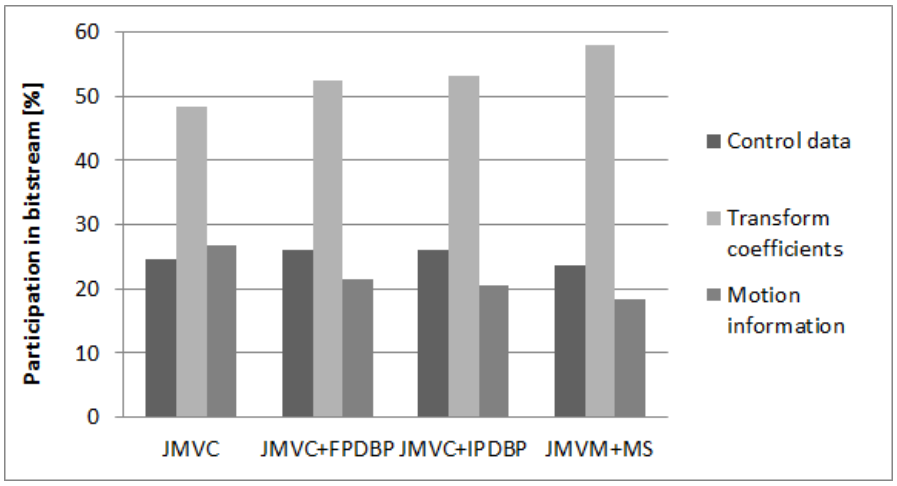


Fig. 6.8. Percentage participation in bitstream for each syntax element in different codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

Fig. 6.6-6.8 show bitstream structure for analyzed codecs. The values are averaged over all considered test sequences and QP values. Detailed results for individual QP values are presented in Annex C.8. The percentage of the bitstream for motion information for JMVC+FPDBP and JMVC+IPDBP codecs is significantly smaller than for JMVC codec. The difference observed for each of the proposed codecs in 2-view case is between 4-11 [pp] and for *view 1* in 3-view case. In 3-view case, the difference observed for *view 2* is between 3-8 and 3-9 [pp] for JMVC+FPDBP and JMVC+IPDBP respectively. Also, we observe that for JMVM+MS codec, the percentage of the bitstream for motion information is even smaller, however, at the same time, percentage of bitstream for transform coefficients increases. Based on these results, it can be concluded that prediction in Motion Skip generates larger prediction residuum in this case, which obviously limits the achieved coding gains.

The conclusion from analyses of the influence of the proposed depth-based inter-view prediction algorithms on MVC codec performance is that the proposed FPDBP and IPDBP algorithms can limit the number of macroblocks in the bitstream, for which motion information need to be transmitted. This results in increased performance of MVC codec. Also, results show that the proposed prediction methods provide larger coding gains that the concurrent Motion Skip coding tool. As a consequence, we have shown that the proposed depth-based inter-view prediction algorithms can be successfully adopted for motion information prediction in MVC, increasing the performance of multiview video codec.

6.3.3. Influence of depth quality on coding efficiency of MVC

The influence of depth quality on coding efficiency of MVC is an important issue in case of practical applications of the proposed depth-based inter-view prediction algorithms. As discussed in Section 1.1, depth information utilized in 3D video systems is usually transmitted to decoder using some lossy compression methods. As a result, the inter-view projection conducted in the proposed algorithms is made with modified, not original depth information. In texture synthesis applications, a slight change in luminance value of a pixel in texture image does not affect the quality of rendered image significantly, especially if the subjective quality evaluation is concerned [Yan05]. However, the same slight change of a depth image value may result in very annoying changes of luminance value in texture image rendered with this depth [Merk08]. The reason is that position of a pixel calculated using the modified depth value can differ substantially. Obviously, this property of depth may also affect the accuracy of motion information predicted with the proposed depth-based inter-view prediction algorithms and, consequently, performance of a multiview codec utilizing these

algorithms. In this section, the impact of depth quality on performance of MVC with the proposed prediction algorithms applied is analyzed.

The experiment setup is the same as described in Section 6.3.2, however, this time only the proposed inter-view prediction algorithms FPDBP and IPDBP implemented in MVC are considered. As a result, performance of JMVC+FPDBP and JMVC+IPDBP codecs is compared with the original MVC codec [Chen09] (JMVC). The experiment is conducted for 2-view and 3-view setup (see MVC configuration in Annex B.1). Also, the same complete set of multiview test sequences (see Annex A.1) is encoded using QP values equal to {22,27,32,37} [Su06, Tan08].

To check the influence of depth quality on performance of the analyzed algorithms, depth information for JMVC+FPDBP and JMVC+IPDBP codecs is provided in form of disparity maps compressed using MVC codec for QD values equal to {0,20,30,40,50}. This way, a wide range of QD values is tested, with special emphasis on lower bitrates, for which compression artifacts are most annoying. Similarly as in previous experiments, the value of QD=0 indicates the usage of original, uncompressed disparity maps. The approach of coding disparity maps by means of a texture coder has been proposed by MPEG [MP11b, ISO11a] and is commonly used in research on the impact of disparity maps compression on the quality of synthesized texture.

The results are presented for side views only. Similarly as in Section 6.3.2, in the 2-view case this view is denoted as *view 1*. Also in 3-view camera setup, the same notation for the analyzed side views is used: *view 1* and *view 2* indicate central and outermost views respectively. The position of the cameras in a multiview acquisition system is illustrated in Fig. 2.10b (refer to view c_1 and c_2 respectively).

Fig. 6.9-6.11 present experimental results obtained for each of the analyzed QD values using 2-view and 3-view codec setup. The values are averaged over all considered test sequences and show improvement in compression performance as a function of QD value for JMVC+FPDBP and JMVC+IPDBP codecs against original JMVC codec. For that purpose, Bjontegaard metrics (Section 2.2.2) indicating the average changes of bitrate for encoded side views are utilized. Bjontegaard metrics for individual test sequences, showing the average changes of luminance PSNR (PSNRY), together with numerical values for bitrate, are presented in Annex C.9. As previously, bitrate required for transmission of depth information is not included in the results.

The results show that lossy compression of depth information generally decreases the performance of the proposed JMVC+FPDBP and JMVC+IPDBP codecs. However, the change in coding gains is only slight. For analyzed range of QD values, the value of average bitrate reduction decreases by less than 1.0 [pp] when compared with the results for QD=0. Additionally, for a wide

range of QD values, the difference usually does not exceed 0.1-0.3 [pp]. A greater decrease is observed for the largest QD values: for QD=50 the difference exceeds 0.5 [pp]. Also, for the outermost view (*view 2*) in 3-view camera setup, a noticeable decrease of coding gain is observed between values for QD=20 and QD=0. In this case, the large distance between coded and reference views results in less accurate inter-view prediction due to only slight changes in depth quality.

On the other hand, compression of disparity maps utilized in depth-based inter-view prediction may increase the coding efficiency. Such situation is observed especially for test sequences with inaccurate depth information, e.g. *Newspaper* or *Balloons*. Here, false edges occurring in disparity maps are usually smoothened as a result of the lossy compression. Consequently, their impact on the inaccurate inter-view prediction is being reduced to some extent.

In the experiment, the influence of depth compression on the usage of low-cost macroblock modes is also analyzed. Fig. 6.12-6.14 show the average usage of all low-cost modes available in the analyzed multiview codecs for different QD values. The results are averaged over all test sequences and QP values used in the experiment. Detailed results for individual low-cost macroblock modes are presented in Annex C.10. We can observe that decreasing the depth quality changes the average usage of low-cost modes only slightly. The difference against values obtained for original, uncompressed disparity maps (QD=0) does not exceed 0.5 [pp]. The usage of low-cost modes that utilize the proposed depth-based inter-view prediction algorithms (IVS and IVD modes) is smaller for lower depth quality, while the usage of traditional low-cost modes grows slightly. Nevertheless, the total usage of all of available low-cost modes decreases which results in limited compression efficiency of the multiview codec.

Now, let us consider the causes of the observed impact of decreasing the depth quality on the performance of the depth-based inter-view prediction algorithms. As discussed in Section 6.1, lossy compression of disparity maps changes the size of occluded area determined by the proposed algorithms. It can also improve the inter-view consistency of depth information (refer to Section 6.1). Additionally, the effect of smoothing the sharp edges indicating object borders in the disparity maps occurs. On the one hand, this results in less accurate inter-view prediction, but also may reduce the influence of errors introduced by imperfect depth estimation algorithms, that occur in original, uncompressed depth information. Nevertheless, decreasing the depth quality has generally a negative impact on performance of the proposed prediction algorithms, reducing the compression efficiency of a codec. Fortunately, as presented in this section, the coding gains achieved for compressed and original disparity maps differ only slightly which makes the proposed depth-base inter-view prediction algorithms suitable for practical applications of multiview video coding.

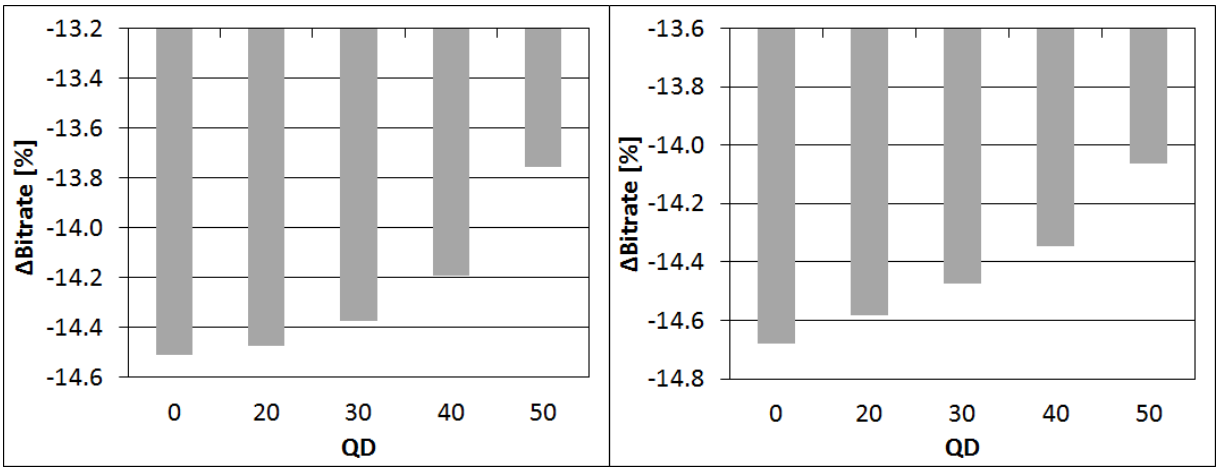


Fig. 6.9. Bjontegaard metrics for JMVC+FPDBP (left) and JMVC+IPDBP (right) vs. JMVC codec calculated for QP={22,27,32,37} and different QD values (2-view case, *view 1*).

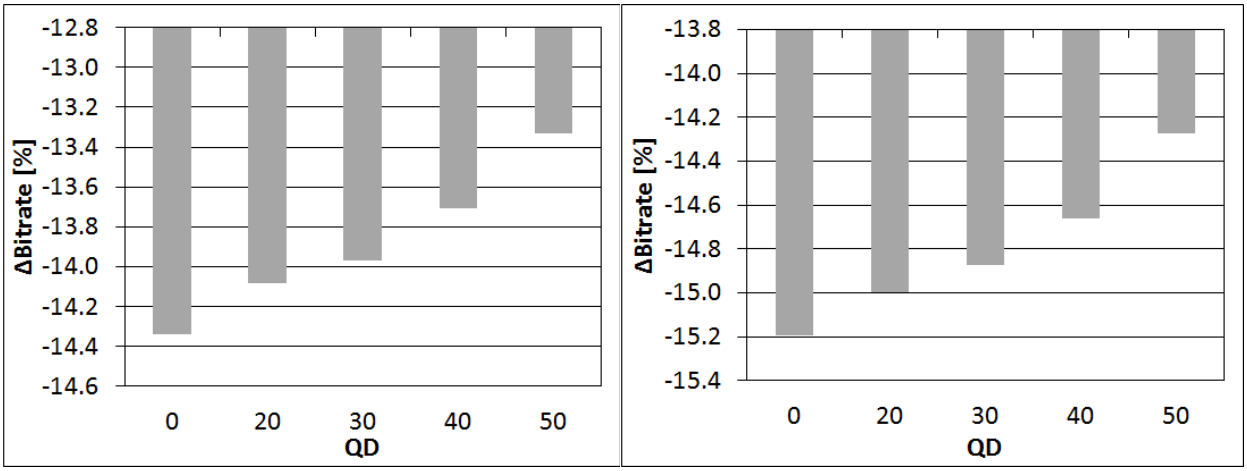


Fig. 6.10. Bjontegaard metrics for JMVC+FPDBP (left) and JMVC+IPDBP (right) vs. JMVC codec calculated for QP={22,27,32,37} and different QD values (3-view case, *view 1*).

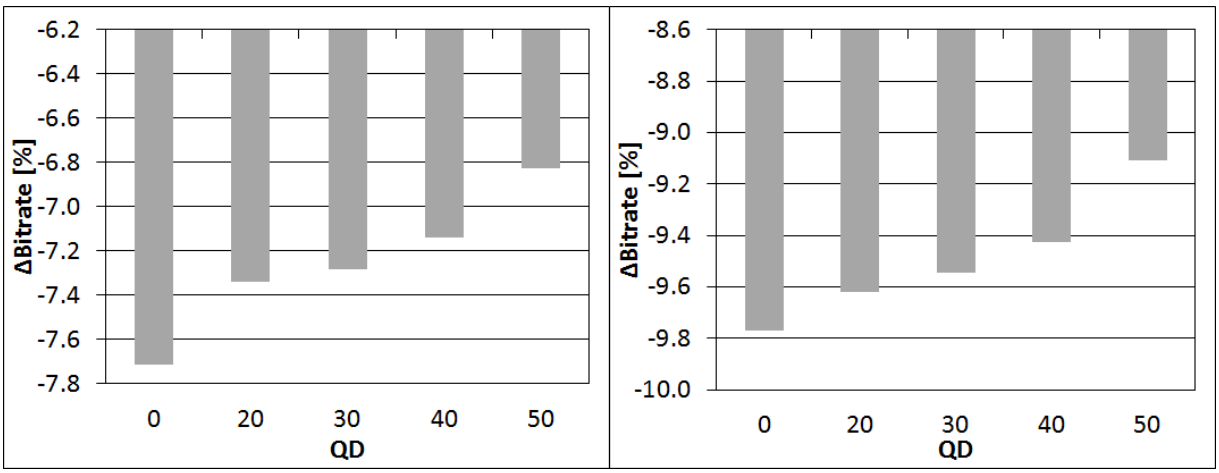


Fig. 6.11. Bjontegaard metrics for JMVC+FPDBP (left) and JMVC+IPDBP (right) vs. JMVC codec calculated for QP={22,27,32,37} and different QD values (3-view case, *view 2*).

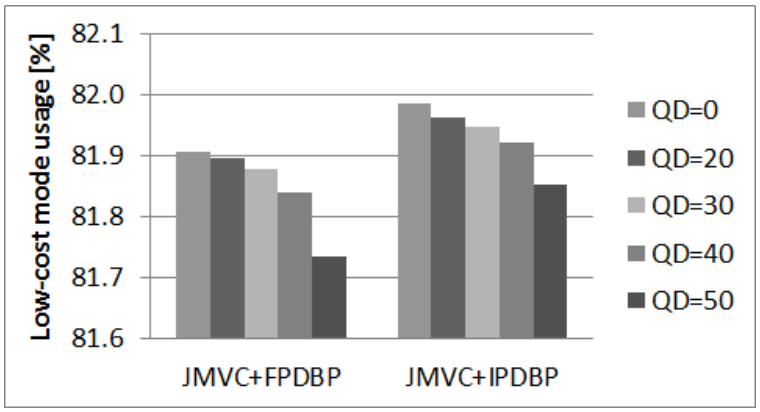


Fig. 6.12. Average usage of all low-cost modes in proposed codecs for different QD values, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

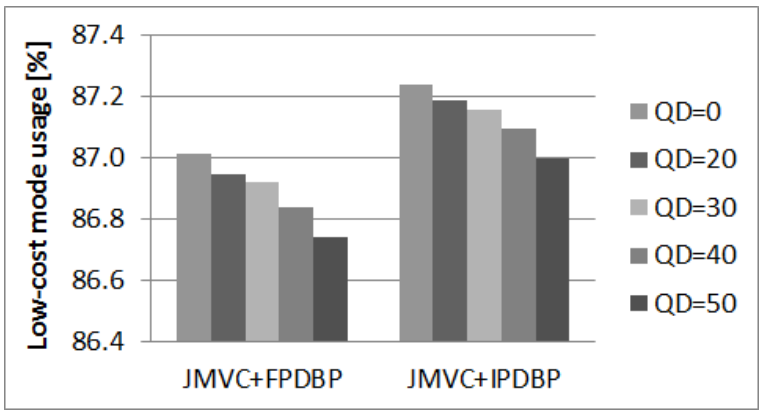


Fig. 6.13. Average usage of all low-cost modes in proposed codecs for different QD values, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

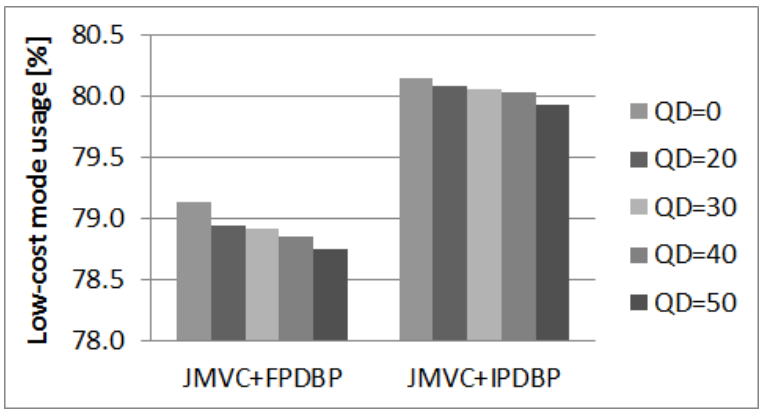


Fig. 6.14. Average usage of all low-cost modes in proposed codecs for different QD values, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

6.4. Impact of proposed prediction methods on compression efficiency of MV-HEVC

In this section, influence of the proposed depth-based inter-view prediction on performance of MV-HEVC codec is analyzed. Previously, the proposed algorithms have been experimentally evaluated in the state-of-the-art multiview video codec called MVC. Here, the impact on compression efficiency in a new generation multiview HEVC-based video codec, named MV-HEVC (see Section 2.4.2) is investigated. As discussed in Section 5.3, utilization of the proposed prediction algorithms in MV-HEVC requires a different approach to motion information coding than the one proposed for MVC codec. The reasons are the substantial changes in motion-compensated prediction of the HEVC core when compared to AVC, especially the utilization of a new advanced motion information predictors arranged in form of an ordered list of merge candidates and introduction of the concept of block merging (refer to Section 2.1.3). Consequently, the question arises if the proposed depth-based inter-view prediction algorithms can still achieve good performance and provide additional coding gains to the new generation MV-HEVC multiview video codec.

In the experiment, performance of MV-HEVC codec with IPDBP algorithm implemented (denoted as MV-HEVC+IPDBP) is compared with the original MV-HEVC codec [Dom11, Sta12]. As discussed in Section 6.3.2, impact of the proposed FPDBP and IPDBP algorithms (see Section 3.2) on performance of the state-of-the-art multiview video codec is very similar. As a consequence, in order to reduce the number of tests to be conducted in the experiment, only one of the algorithms is investigated. The selection of IPDBP results from the fact that this algorithm does not require depth information of the coded view, which makes it applicable to most of the multiview video coding scenarios. Also, coding gains reported in Section 6.3.2 are slightly better for IPDBP than for FPDBP. Similarly as in previous experiments on coding performance of MVC, unassigned area filling algorithm called *FILLmax* is used in IPDBP.

As discussed in Section 5.3, the proposed depth-based inter-view predictor is adopted in MV-HEVC as an additional merge candidate, called depth-based predictor (*DBP*). Position on the candidate list determines cost of selecting the candidate and affects coding efficiency. In order to analyze this dependency, the proposed *DBP* candidate is implemented at three different positions on the list – for *list index* 1, 2 and 3 (Fig. 5.5b, c and d respectively). We will refer to these syntax variants of MV-HEVC+IPDBP codec as: MV-HEVC+IPDBP1, MV-HEVC+IPDBP2 and MV-HEVC+IPDBP3 respectively.

The experiment is conducted for 2-view and 3-view MV-HEVC codec setup with configuration presented in Annex B.2. Analyzed codecs are investigated using a complete set of multiview test sequences (see Annex A.1) and QP values equal to {22,27,32,37} [Bos11].

Depth information for MV-HEVC+IPDBP codec is provided in form of disparity maps compressed using MV-HEVC codec with QD values equal to quantization parameter for texture (QD=QP). As discussed in Section 6.3.3, the approach of coding disparity maps by means of a texture coder is commonly used in MPEG [MP11b, ISO11a]. In such a case, QD values are usually selected by group of experts during subjective tests in order to obtain the {QP, QD} pair that provides the best quality of the synthesized view. However, for the purpose of practical utilization in multiview video coding, the procedure described above is not applicable. Therefore, in the experiment, a simple solution of using QD value that corresponds to value of quantization parameter for texture is adopted.

The results are presented for side views only. Similarly as in previous experiments, the same notation for the analyzed side views is used. In the 2-view case dependent view is denoted as *view 1*. In 3-view camera setup, *view 1* and *view 2* indicate central and outermost views respectively (refer to position of the cameras in an exemplary multiview acquisition system illustrated in Fig. 2.10b: view c_1 and c_2 respectively).

Tab. 6.7. Bjontegaard metrics for different syntax variants of MV-HEVC+IPDBP codec vs. MV-HEVC codec, calculated for side view with QP={22,27,32,37} (2-view case, *view 1*).

QP={22,27,32,37}	ΔPSNRY [dB]			ΔBitrate [%]		
	***DBP* candidate position**					
Sequence	**1**	**2**	**3**	**1**	**2**	**3**
Poznan Street	0.03	0.03	0.03	-2.4	-2.3	-2.2
Poznan Hall2	0.03	0.03	0.03	-4.1	-4.2	-3.7
Dancer	0.06	0.06	0.06	-2.3	-2.3	-2.3
GT Fly	0.30	0.29	0.29	-12.4	-12.3	-12.1
Kendo	0.15	0.15	0.15	-5.7	-5.7	-5.5
Balloons	0.32	0.32	0.31	-11.8	-11.6	-11.4
Lovebird1	0.01	0.01	0.01	-0.5	-0.4	-0.6
Newspaper	0.13	0.13	0.13	-4.5	-4.4	-4.4
Avg.	0.13	0.13	0.13	-5.5	-5.4	-5.3

In Tab. 6.7 and Tab. 6.8 results obtained for individual test sequences encoded using 2-view and 3-view case respectively are presented. The improvement in compression performance is measured for three syntax variants of MV-HEVC+IPDBP codec (three different *DBP* positions on

the merge candidate list) and compared with MV-HEVC using Bjontegaard metrics (Section 2.2.2). Average changes of bitrate and luminance PSNR (PSNRY) of encoded side views are shown. Following the approach applied in previous experiments, bitrate required for transmission of depth information is not included in the results.

Tab. 6.8. Bjontegaard metrics for different syntax variants of MV-HEVC+IPDBP codec vs. MV-HEVC codec, calculated for side views with QP={22,27,32,37} (3-view case).

View 1 (central)						
QP={22,27,32,37}	**ΔPSNRY [dB]**			**ΔBitrate [%]**		
	***DBP* candidate position**					
Sequence	**1**	**2**	**3**	**1**	**2**	**3**
Poznan Street	0.04	0.03	0.03	-2.6	-2.5	-2.3
Poznan Hall2	0.03	0.03	0.02	-3.6	-3.7	-3.2
Dancer	0.07	0.07	0.07	-2.5	-2.5	-2.5
GT Fly	0.19	0.18	0.18	-8.1	-7.9	-7.8
Kendo	0.12	0.12	0.11	-5.1	-5.0	-4.7
Balloons	0.35	0.35	0.34	-13.7	-13.5	-13.1
Lovebird1	0.00	0.00	0.01	-0.3	-0.2	-0.3
Newspaper	0.12	0.12	0.12	-4.6	-4.6	-4.5
Avg.	0.12	0.11	0.11	-5.1	-5.0	-4.8
View 2 (outer)						
QP={22,27,32,37}	**ΔPSNRY [dB]**			**ΔBitrate [%]**		
	***DBP* candidate position**					
Sequence	**1**	**2**	**3**	**1**	**2**	**3**
Poznan Street	0.02	0.01	0.02	-1.2	-1.0	-1.2
Poznan Hall2	0.02	0.02	0.02	-2.7	-2.6	-2.5
Dancer	0.02	0.01	0.02	-0.7	-0.5	-0.8
GT Fly	0.18	0.17	0.18	-7.1	-6.9	-7.0
Kendo	0.07	0.07	0.07	-2.5	-2.4	-2.4
Balloons	0.19	0.18	0.18	-6.4	-6.2	-6.1
Lovebird1	0.00	-0.01	0.00	0.0	0.1	-0.1
Newspaper	0.03	0.03	0.03	-1.1	-1.0	-1.1
Avg.	0.07	0.06	0.06	-2.7	-2.6	-2.7

The results show that the proposed MV-HEVC+IPDBP achieves coding gains when compared with original MV-HEVC. The bitrate reduction averaged over all considered test sequences in 2-view case is equal to 5.3-5.5 [%] depending on the position of *DBP* candidate on the merge candidate list. In 3-view case, the average bitrate reductions are equal to 4.8-5.1 [%] for *view 1* and 2.6-2.7 [%] for *view 2*.

We can also notice, that the position of *DBP* candidate on the list affects the compression performance of the codec. The largest coding gains are observed for *DBP* placed at the front of the

list regardless of which of the considered coding scenarios (2-view or 3-view) is analyzed. However, the difference in average bitstream reduction against other syntax variants is usually less than 0.5 [pp]. Moreover, it can be observed that selecting the position of *DBP* candidate closer to the front of the list is not always the best strategy. For *view 2* in 3-view case (Tab. 6.8), performance of MV-HEVC+IPDBP2 codec is worse than for MV-HEVC+IPDBP3. The reason is a large distance between the coded *view 2* and the only available reference view, the base view. As a consequence, the proposed inter-view prediction of motion information is less accurate and some other predictors available on the candidate list perform better in this case.

Fig. 6.15 shows exemplary rate-distortion curves obtained for the analyzed codecs. Similarly as for MVC codec, bitrate savings achieved by MV-HEVC+IPDBP grow for lower bitrates, as the cost of encoding motion information is higher for small bitstreams.

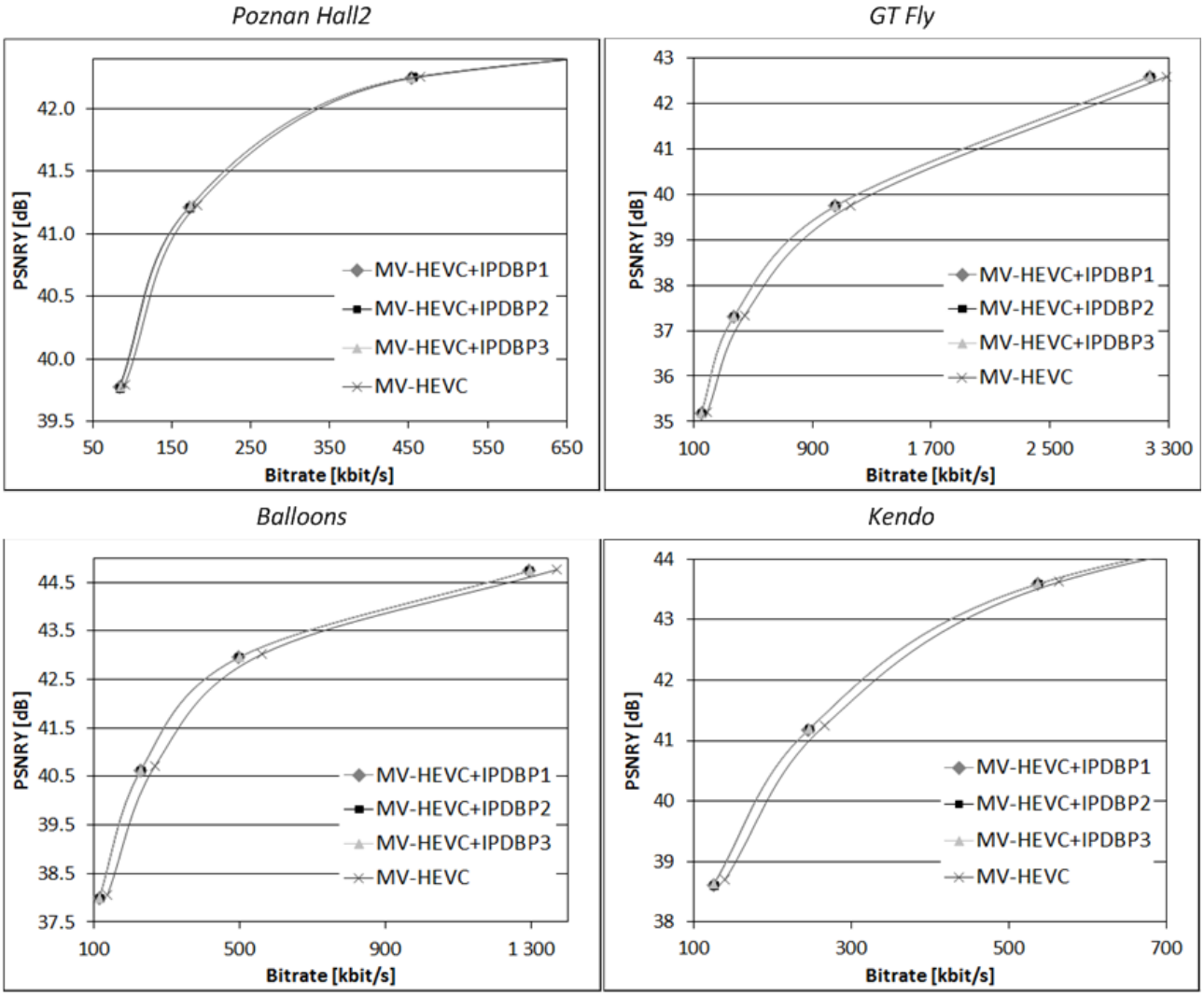


Fig. 6.15. Exemplary rate-distortion curves for MV-HEVC+IPDBP and MV-HEVC codecs (2-view case, *view 1*).

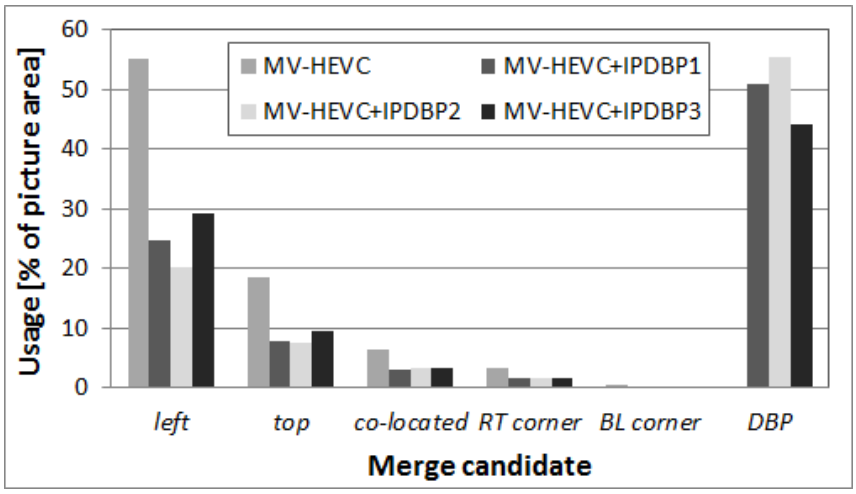


Fig. 6.16. Usage of merge candidate predictors in MV-HEVC+IPDBP and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

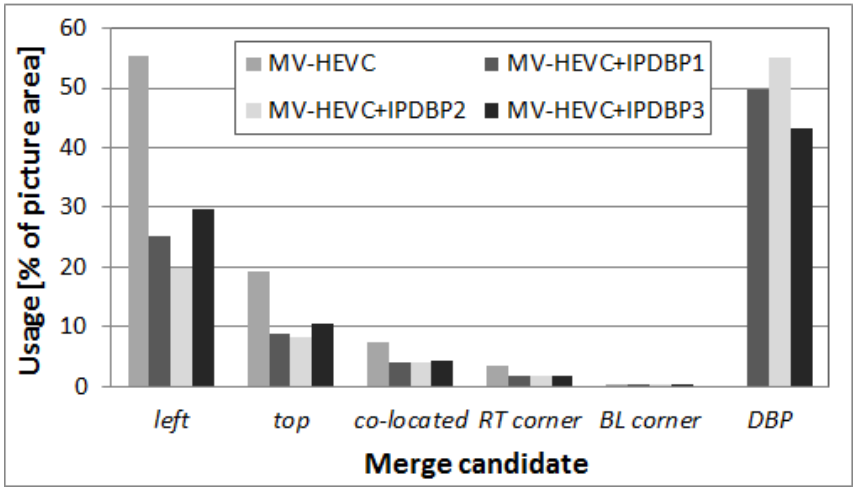


Fig. 6.17. Usage of merge candidate predictors in MV-HEVC+IPDBP and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

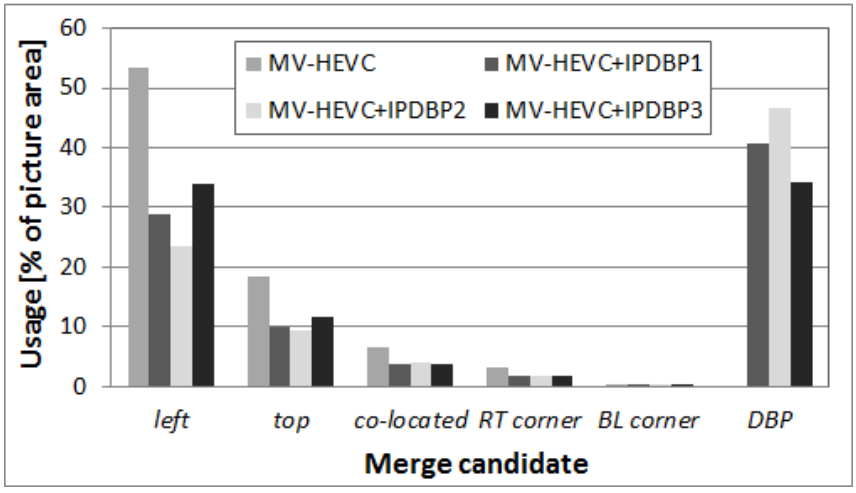


Fig. 6.18. Usage of merge candidate predictors in MV-HEVC+IPDBP and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

Fig. 6.16-6.18 show usage of all available merge candidate predictors in the analyzed codecs for 2-view and 3-view coding scenarios. The values are averaged over all test sequences and QP values utilized in the experiment. Results for individual QP values are presented in Annex C.11. In HEVC-based codec, each of the motion information predictors can be assigned to represent blocks with different size. The size of a block may vary from 64×64 to 4×4 pixels. Therefore, in the experiment, we analyze the percentage of picture area to which every of the considered predictors is assigned.

According to presented results, introduction of a new *DBP* prediction candidate into merge candidate list increases the overall usage of merge candidate predictors by 2.8-5.3 [pp] depending on the MV-HEVC+IPDBP syntax variant and coding scenario (see Annex C.11). However, the influence of the position of *DBP* candidate on the list is small. The differences between corresponding values of the overall usage of merge candidate predictors for the three considered MV-HEVC+IPDBP syntax variants are less than 0.2 [pp]. On the other hand, the usage of individual merge candidates varies noticeably for these syntax variants and relates especially to the proportion between *DBP* and *left* candidates, the two most frequently used predictors.

In Fig. 6.19-6.21 usage of different types of coding units available in HEVC-based codec is presented. Each coding unit (CU) can be encoded with intra (denoted as *INTRA*) or inter prediction. Among inter modes, we can specify ones that require transmission of:

- transform coefficients and motion vectors prediction error (*INTER*),
- transform coefficients only (*INTER-MERGE*),
- no transform coefficients nor motion vectors prediction error (*MERGE*).

As the results for all considered MV-HEVC+IPDBP syntax variants are similar, we present figures for MV-HEVC+IPDBP1 and MV-HEVC codecs only. Presented values are averaged over all test sequences and QP values utilized in the experiment. Detailed results for individual QP values and all of the analyzed codecs can be found in Annex C.12. For the same reasons as above, we analyze the percentage of picture area to which every of the considered CU mode is assigned.

In Fig. 6.19-6.21, we can observe that presence of the new *DBP* predictor on the merge candidate list results in higher usage of *MERGE* modes to represent encoded CUs. Depending on the MV-HEVC+IPDBP syntax variant and QP value, the increase is by approximately 3.6-6.6 [pp] (refer to Annex C.12). At the same time, usage of modes that require sending of additional motion vectors prediction error or transform coefficients (*INTER* and *INTER-MERGE*) decreases, which results in improved compression efficiency of the MV-HEVC+IPDBP codec.

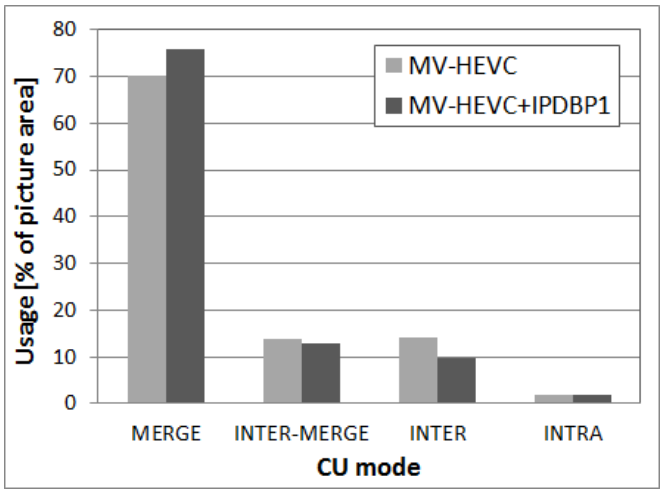


Fig. 6.19. Usage of coding units (CUs) with different modes in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

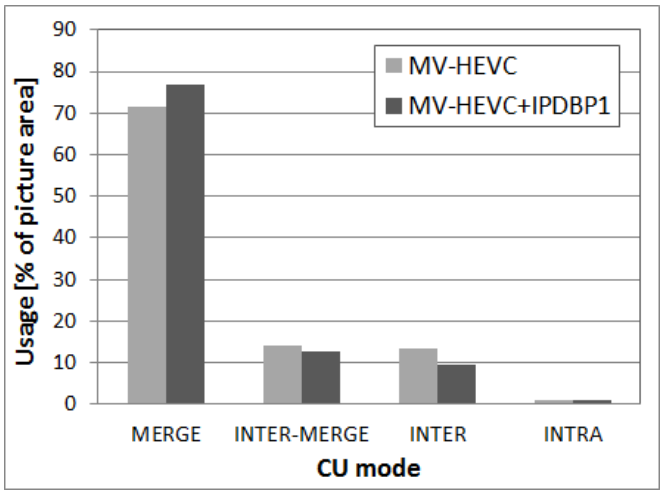


Fig. 6.20. Usage of coding units (CUs) with different modes in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

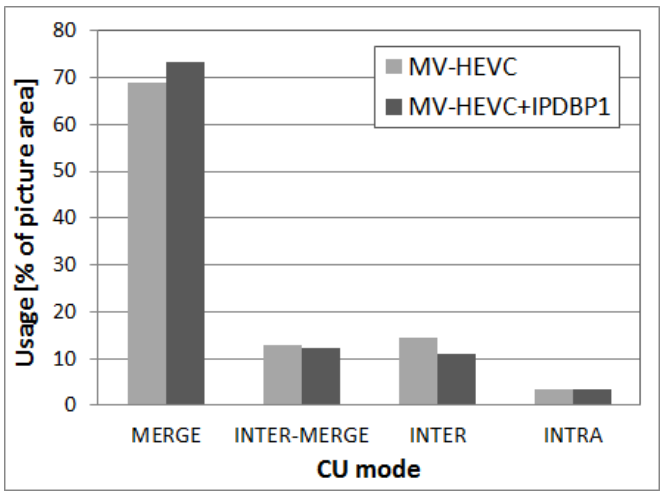


Fig. 6.21. Usage of coding units (CUs) with different modes in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

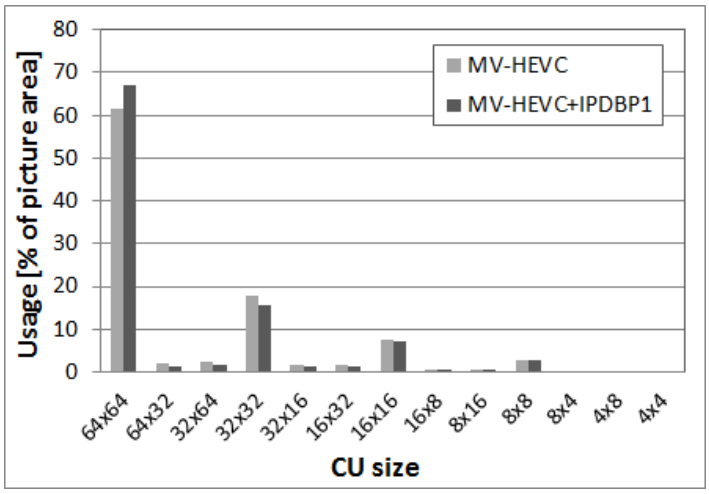


Fig. 6.22. Usage of coding units (CUs) with different size in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (2-view case, *view 1*).

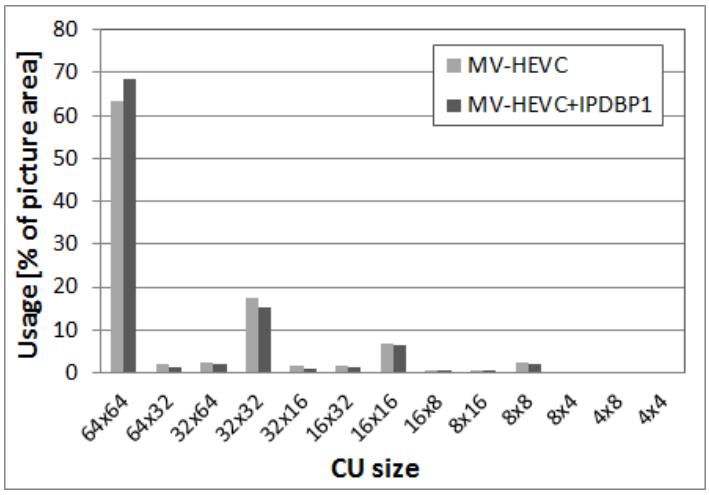


Fig. 6.23. Usage of coding units (CUs) with different size in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 1*).

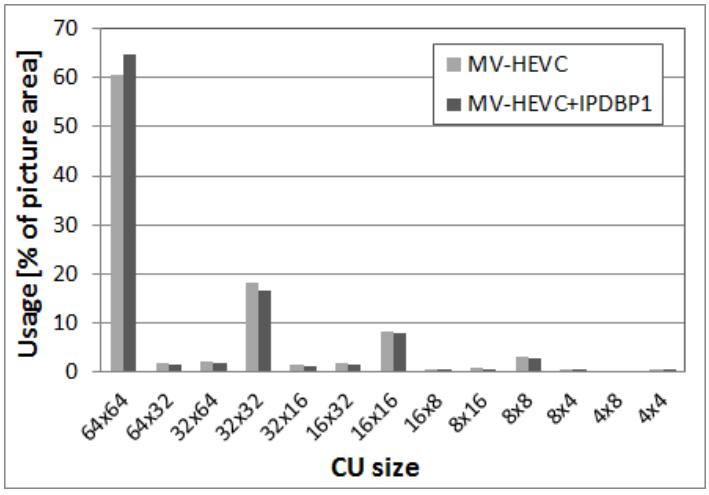


Fig. 6.24. Usage of coding units (CUs) with different size in MV-HEVC+IPDBP1 and MV-HEVC codecs, averaged over QP={22,27,32,37} and all test sequences (3-view case, *view 2*).

Another reason for coding gains achieved by the proposed MV-HEVC+IPDBP codec is illustrated in Fig. 6.22-6.24 which show the usage of CUs with different size for analyzed codecs. The values show a percentage of picture area covered by CUs of each available size. Again, as the results for all considered MV-HEVC+IPDBP syntax variants are similar, figures for MV-HEVC+IPDBP1 and MV-HEVC codecs are presented. Presented values are averaged over all test sequences and QP values used in the experiment. For results showing all of the analyzed codecs and QP values, please refer to Annex C.13.

The results show clearly that the usage of CUs with the largest available size (64×64 pixels) is higher for MV-HEVC+IPDBP codec. The increase is by approximately 2.6-6.1 [pp] depending on MV-HEVC+IPDBP syntax variant and QP value (see Annex C.13). This means that the proposed depth-based inter-view predictor can preserve some CUs from further division into smaller parts. It also reveals an important feature of the proposed prediction algorithms in which motion information is predicted independently for each pixel of encoded CU (refer to Section 3.1). As a consequence, bits for signaling the further CU partitioning are not transmitted in the bitstream which leads to coding gains achieved by MV-HEVC+IPDBP codec.

To conclude, the experimental results presented in this section show that, despite the advanced motion prediction methods used in HEVC-based codec, the proposed depth-based inter-view prediction algorithms can still improve the compression efficiency of a new generation multiview video codec that implements the HEVC core, i.e. MV-HEVC. Introduction of the proposed *DBP* predictor on the merge candidate list increases the overall usage of merge candidate predictors. It also results in higher usage of low-cost modes that do not require transmission of additional motion vectors prediction error or transform coefficients. Additionally, the proposed depth-based inter-view predictor preserves some CUs from further division into smaller parts. As a consequence, number of bits transmitted in the bitstream is reduced, which leads to coding gains when compared to original MV-HEVC codec.

6.5. Complexity analysis of proposed algorithms on multiview video codec

In this section, computational complexity of the proposed algorithms for depth-based inter-view motion information prediction and coding is evaluated. A method proposed by JCT-VC,

described in Section 2.2.3, is adopted for that purpose. According to this method, a rough computational complexity analysis of a video codec can be conducted by measuring a single runtime of encoder or decoder using the same machine for each tested codec. In this way, the impact of the proposed FPDBP and IPDBP algorithms on performance of MVC and MV-HEVC multiview video codecs is investigated.

Experiment is conducted using a PC computer with Intel Core i7 CPU, 12 GB RAM and Windows 7 Professional platform. For each of the analyzed codecs, appropriate codec configuration is applied (see Annex B.1 and Annex B.2 for MVC and MV-HEVC respectively). To simplify the experiment, only 2-view setup is used. Selected test sequences are encoded and decoded with QP values equal to $\{22,27,32,37\}$. In order to limit the number of conducted tests, a sub-set of four test sequences is used: *Poznan Street*, *GT Fly*, *Balloons* and *Newspaper* (refer to Annex A.1). To further reduce the number of tests, only first 24 frames of each test sequence are encoded/decoded. In case of MVC codecs that utilize the proposed depth-based inter-view prediction algorithms, depth information in form of original, uncompressed disparity maps is used. For MV-HEVC codecs, disparity maps compressed with QD value equal to QP for texture are used. Encoding and decoding times are measured based on the number of CPU cycles reported by the operating system.

Unfortunately, the results obtained with the utilized evaluation method suffer from inaccuracies caused by the CPU load due to other tasks and varying hard drive access time. As a consequence, the experimental results are not repeatable and may be affected with significant measurement errors. In order to assess the accuracy of the obtained experimental results, the encoding and decoding time of an exemplary test sequence have been measured using the considered computational complexity evaluation method. The first 24 frames of *Balloons* sequence have been encoded and decoded 30 times using MVC and MV-HEVC codecs in order to obtain population samples for statistical analysis. Each of the samples has been tested using Lillefors test to confirm the null hypothesis that the samples come from a distribution in the normal family at significance level 0.05. Next, standard deviation and 99 [%] confidence interval for each sample have been computed. Analysis of the measured encoding and decoding times shows that the ratios of confidence interval to average value for each sample do not exceed 0.4 [%] in case of coding time and are less than 2.7 [%] for decoding time (see Annex C.14). Consequently, deviation of the single test result from the average value is negligibly small and can be omitted during a rough evaluation of computational complexity of the analyzed algorithms. As a result, in the experiment, only single encoding or decoding will be performed for each input data and codec configuration in order to significantly reduce the number of conducted tests.

In order to analyze the computational complexity of the proposed algorithms in a more comprehensive way, two main tasks performed in the proposed depth-based inter-view coding methods are investigated in the experiment:

- *DBP task*: depth-based projection of pixel coordinates, occluded area detection and unassigned area filling (refer to steps 1) to 4) in Section 3.2.1 or steps 1) to 5) in Section 3.2.2 for FPDBP and IPDBP algorithms respectively),
- *MCP task*: point-to-point motion-compensated prediction using predicted motion information - an additional block coding mode available in the codec (refer to step 5) in Section 3.2.1 or step 6) in Section 3.2.2 for FPDBP and IPDBP algorithms respectively).

Increase of encoding/decoding time due to each task is analyzed in presented results. We will refer to these values as: Inc_{DBP} for the first task and Inc_{MCP} for the latter one. Inc_{DBP} and Inc_{MCP} are calculated using the following equations:

$$Inc_{DBP} = \frac{T_{DBP}}{T_{ref}} \cdot 100\% \tag{6.1}$$

$$Inc_{MCP} = \frac{T - T_{DBP} - T_{ref}}{T_{ref}} \cdot 100\% \tag{6.2}$$

where: T is the time of encoding/decoding measured for the analyzed codec that utilize one of the proposed prediction algorithms, T_{ref} is the time of encoding/decoding measured for the reference codec and T_{DBP} is the time of depth-based projection of pixel coordinates, occluded area detection and unassigned area filling performed in the analyzed codec.

However, the evaluated algorithms are implemented without any special intention to assure their efficient execution and using non-optimal reference software. Also, in practical applications, a software optimized for target platform, utilizing all possible hardware acceleration would be used. Consequently, due to sub-optimal implementation of the proposed algorithms, the measured values are exaggerated.

In case of MVC codec, computational complexity of FPDBP and IPDBP algorithms is analyzed. JMVC+FPDBP and JMVC+IPDBP codecs, investigated previously in Section 6.3.2, are analyzed and compared with original JMVC codec. In case of FPDBP algorithm, the *o-mask* occlusion detection algorithm is utilized. The other occlusion detection method introduced for FPDBP, called *z-test*, is not evaluated in the experiment as the coding gains achieved by this method are significantly worse (see Tab. 6.2). However, the number of operations performed in *z-test* are much smaller, so it can be concluded that its computational complexity is lower than the one measured for *o-mask* algorithm.

In the experiment, an increase of coding time is analyzed in relation to encoding time of side view only and, also, in relation to overall encoding time of all views. This values will be referred to as *single view* and *all views* respectively. In case of decoding time analyses, relation to all views is analyzed, as the considered implementation of MVC decoder enables processing of all decoded views at the same time only.

As presented in Fig. 6.25, the values of Inc_{DBP} averaged over all considered test sequences vary from 3 to 4 [%] for both FPDBP and IPDBP algorithms in relation to single view encoding time. In relation to encoding time of all views, the averaged value of Inc_{DBP} falls to approximately 2 [%] for both analyzed algorithms (see Fig. 6.26). In case of decoder, due to suboptimal implementation of DBP task in MVC codec, decoding time of JMVC+FPDBP and JMVC+IPDBP increases approximately 20 times when compared with JMVC, regardless of the analyzed algorithm or QP value (refer to Annex 0). Experimental results show also that time required for DBP task is almost the same for every QP value, however its proportion to overall processing time changes as the time of coding/decoding depends from QP value. As a consequence, also Inc_{DBP} changes for different QP values (refer to Fig. 6.25 and Fig. 6.26 for coder and Annex 0 for decoder). Additionally, we can observe that Inc_{DBP} is very similar for both analyzed depth-based inter-view prediction algorithms. Nevertheless, values measured for JMVC+FPDBP are usually larger which results from the characteristics of the *o-mask* algorithm utilized in FPDBP. In particular, both FPDBP and IPDBP algorithms use very similar procedure for occlusion detection, however, FPDBP with *o-mask* requires an extra operation to be performed for overlapping pixels in case an incorrect assignment to pixel in reference view occurs (refer to Section 3.3).

In Fig. 6.27, the values of Inc_{MCP} averaged over all considered test sequences are presented. The values vary from 15 to 19 [%] for FPDBP and 15 to 18 [%] for IPDBP algorithm in relation to encoding time of a single view. In relation to encoding time of all views, the averaged value of Inc_{MCP} is equal to approximately 9-11 [%] for both analyzed algorithms (see Fig. 6.28). Again, due to suboptimal implementation of MCP task in MVC codec, decoding time of JMVC+FPDBP and JMVC+IPDBP increases approximately 50-70 times when compared with JMVC (refer to Annex 0). The values of Inc_{MCP} measured for JMVC+FPDBP and JMVC+IPDBP codecs differ only slightly which results from unequal usage of macroblock modes that utilize the proposed prediction algorithms (IVS and IVD modes) in these codecs. The results show also that value of Inc_{MCP} increases for large QP values which is obviously caused by the growing usage of IVD and IVS modes for smaller bitrates.

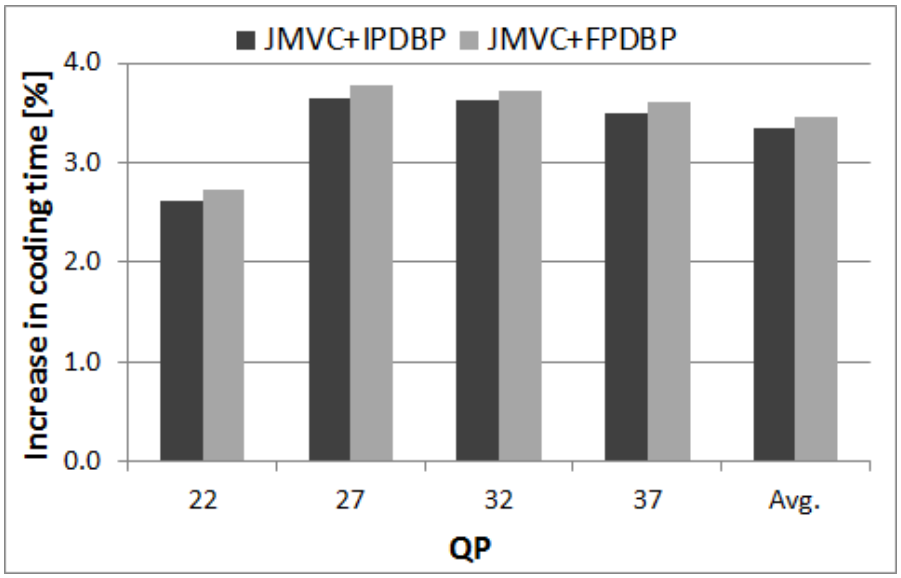


Fig. 6.25. Inc_{DBP} for JMVC+FPDBP and JMVC+IPDBP coders and different QP values, averaged over all test sequences (single view).

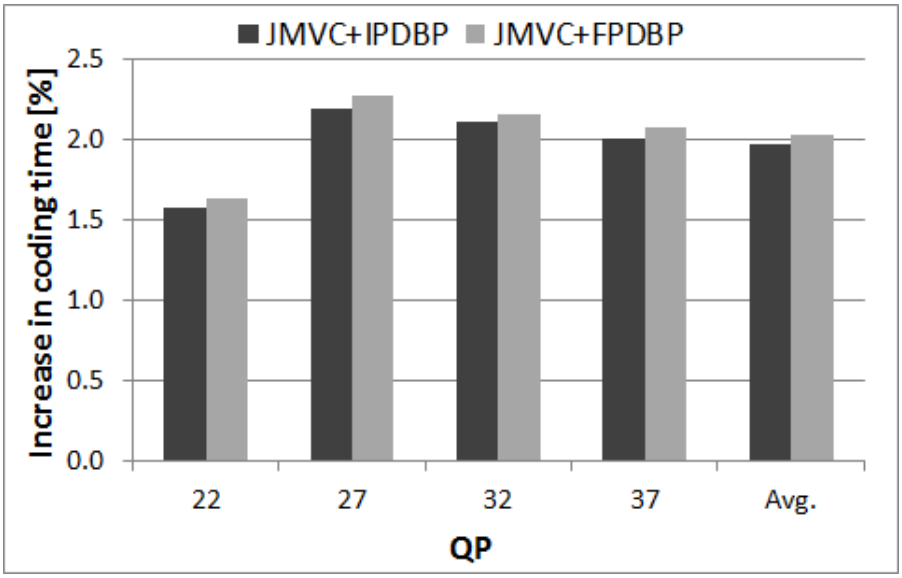


Fig. 6.26. Inc_{DBP} for JMVC+FPDBP and JMVC+IPDBP coders and different QP values, averaged over all test sequences (all views).

Based on the experimental results it can be concluded that, despite the utilization of not optimized implementation of the proposed algorithms, the increase of encoding time is not very large having regard to the fact that two additional macroblock coding modes, i.e. IVS and IVD, have been introduced into codec. Moreover, the observed extra computational overhead is also few times smaller than computational complexity of motion estimation.

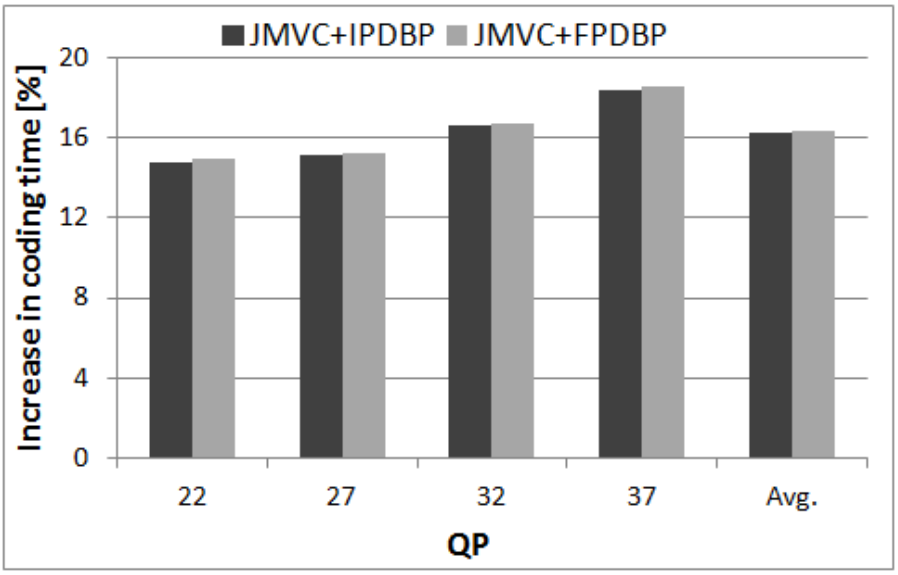


Fig. 6.27. Inc_{MCP} for JMVC+FPDBP and JMVC+IPDBP coders and different QP values, averaged over all test sequences (single view).

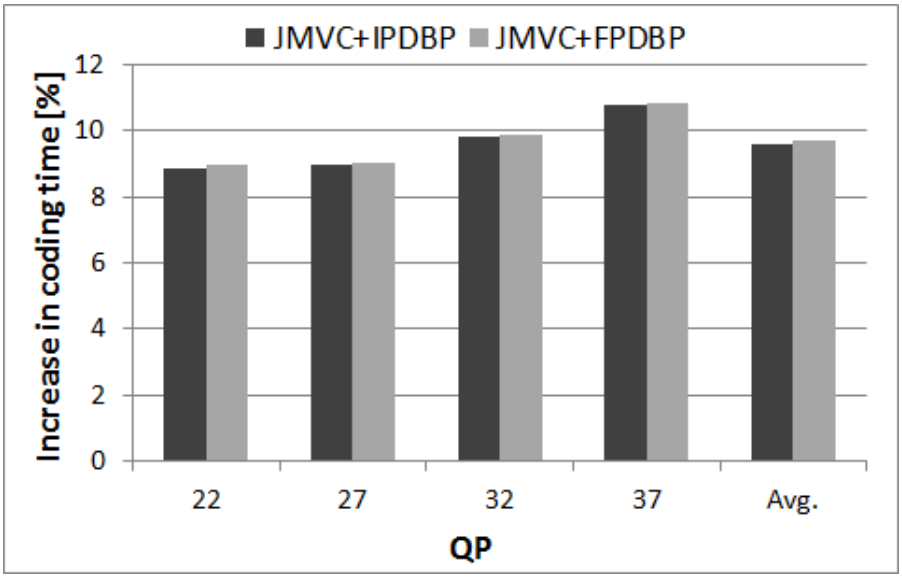


Fig. 6.28. Inc_{MCP} for JMVC+FPDBP and JMVC+IPDBP coders and different QP values, averaged over all test sequences (all views).

Now, let us refer to the main reason for large computational complexity of JMVC+FPDBP and JMVC+IPDBP decoders. Both DBP and MCP tasks performed in the analyzed codecs need to read and write data many times. Unfortunately, in case of the considered MVC implementation, this data is stored in form of external files on the hard drive. As a consequence, delays related to hard drive access time result in significant increase in processing time.

A comparison between the analyzed measures show also a huge disproportion in values obtained for coder and decoder. Unlike in the inter-predicted macroblock modes that use a complex motion estimation at the encoder side and only a simple motion-compensated prediction with a given motion vector at the decoder, the number of operations required in encoder and decoder for IVD and IVS modes is much more symmetric. This obviously results in larger increase of operating time for the decoder, as the encoding time is usually much bigger.

In case of MV-HEVC, computational complexity of IPDBP algorithms is analyzed for three different syntax variants: MV-HEVC+IPDBP1, MV-HEVC+IPDBP2 and MV-HEVC+IPDBP3 (see Section 6.4). For each variant, the proposed merge candidate predictor, called *DBP*, is located at different position on the candidate list. In the experiment, an increase of coding and decoding time measured for each of these codecs is compared with original MV-HEVC. As the considered implementation of MV-HEVC processes all views at the same time, the codecs are analyzed in relation to overall processing time of all views.

In Fig. 6.29, the values of Inc_{DBP} averaged over all considered test sequences are shown. As the DBP task is exactly the same for each of the analyzed syntax variants, the results for only one codec, labeled as MV-HEVC+IPDBP, are presented. All values for coder, presented in Fig. 6.29, are similar and do not exceed 1 [%]. In case of decoder, the value of Inc_{DBP} averaged over all considered test sequences varies from approximately 60 to 90 [%] (refer to Annex C.16). The increase is large due to suboptimal implementation of DBP task in MV-HEVC codec. However, it is also significantly lower than the one measured for MVC codec, because MV-HEVC software is better optimized. Also in case of MV-HEVC codec, conducted experiments show that time required for DBP task differs only slightly for all analyzed QP values. At the same time, its proportion to overall processing time changes as the time of coding/decoding depends on the bitrate. Consequently, the value of Inc_{DBP} also changes with the bitrate (refer to Fig. 6.29 for coder and Annex C.16 for decoder).

As presented in Fig. 6.25, the values of Inc_{MCP} averaged over all considered test sequences vary from 9 to 13 [%] for all MV-HEVC+IPDBP syntax variants. Similarly as for MVC, the measured increase of encoding time is not significant, despite the utilization of not optimized software implementation of the proposed algorithms. In case of decoder, the value of Inc_{MCP} averaged over all considered test sequences varies from approximately 80 to 260 [%] and depends on analyzed syntax variant and QP value (refer to Annex C.16). Obviously, the implementation of MCP task is suboptimal. Nevertheless, as procedures used for point-to-point motion-compensated

prediction are designed more carefully in MV-HEVC software, the values of Inc_{MCP} are significantly smaller than for MVC. In particular, MV-HEVC keeps all the necessary data cached, unlike MVC, which stores it on the hard drive. This obviously reduces required access time resulting in much faster decoding of the bitstream.

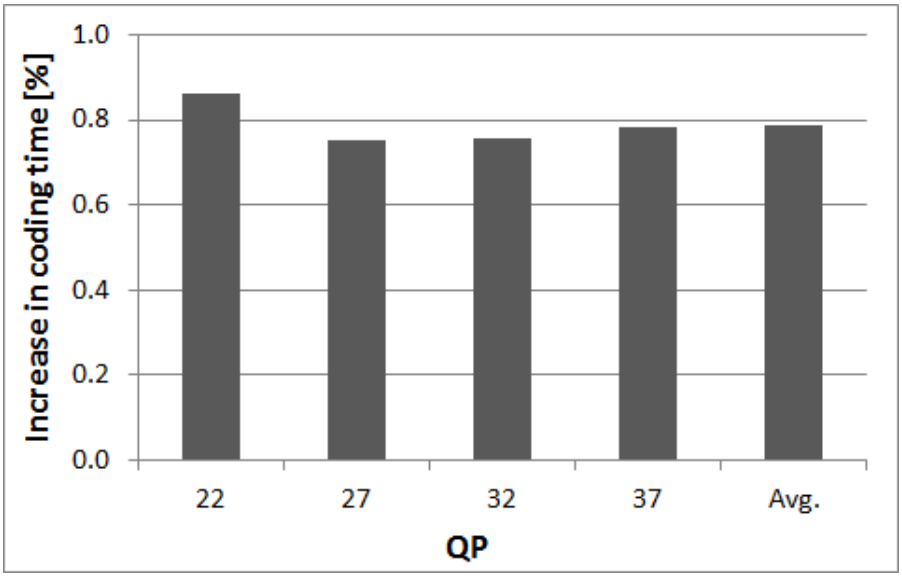


Fig. 6.29. Inc_{DBP} for MV-HEVC+IPDBP coder and different QP values, averaged over all test sequences (all views).

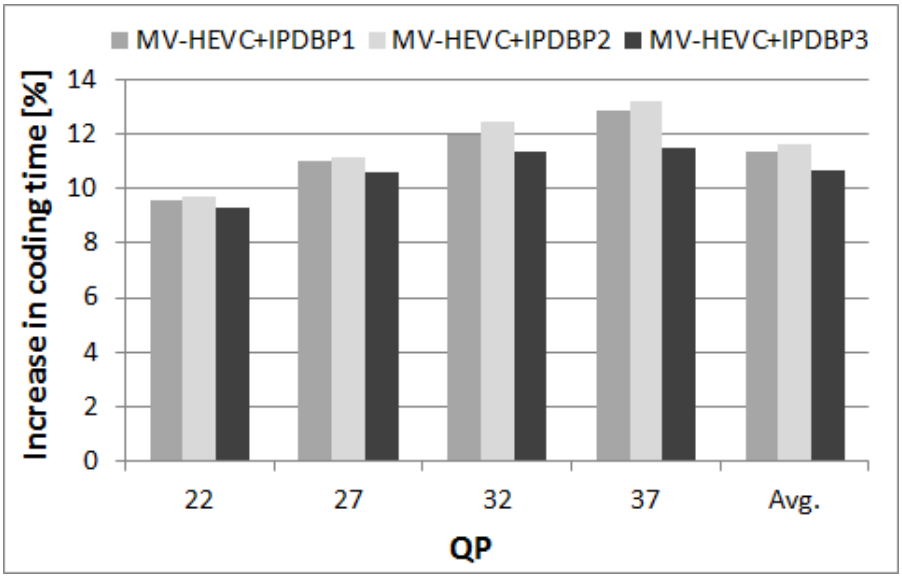


Fig. 6.30. Inc_{MCP} for different syntax variants of MV-HEVC+IPDBP coder and QP values, averaged over all test sequences (all views).

Among all analyzed syntax variants, the value of Inc_{MCP} is largest for MV-HEVC+IPDBP2 (see Fig. 6.30). According to Fig. 6.16, the usage of *DBP* merge candidate is highest for this syntax variant, which explains this relation. We can also notice a regular increase of Inc_{MCP} with the growing value of QP which results from the fact that the usage of *DBP* merge candidate is higher for smaller bitrates. Similarly as for MVC, a huge disproportion in values measured for coder and decoder is observed as well.

The obtained experimental results show that computational complexity of the proposed algorithms, observed especially for the decoder, are not negligible. However, as already stated, presented results are exaggerated due to utilization of not optimized implementations of the proposed methods. Moreover, no additional hardware acceleration is used.

As a consequence, a number of possible improvements exits which should considerably reduce the computational overhead. First of all, current implementation of the proposed point-to-point motion compensation is performed using very inefficient procedure which requires multiple function callbacks. On the other hand, dedicated 3D point projection methods can be applied. One of the possible solutions is a relief texture mapping algorithm [Oliv00, Wol90] developed for rendering uneven surfaces of synthetic computer graphics objects. In this approach, the 3D image warping equation is factorized into a combination of simpler 2D texture mapping operations. As a result, a hardware implementation available in a standard Graphic Processor Unit (GPU) can be utilized to further accelerate the computations. Additionally, in the future 3D video codecs, an optimized procedure for 3D point projection will be already available. In such a case, inter-view projection of pixel coordinates will be performed anyway for texture synthesis purposes. Consequently, the proposed depth-based inter-view prediction algorithms will simply reuse the results of these computations.

Chapter 7

Application of proposed depth-based inter-view motion information prediction in future 3D video coding

7.1. Description of Poznań University of Technology proposal for 3D video coding

In 2010, MPEG began works on new techniques for multiview video plus depth coding. As a result, the Call for Proposals (CfP) on compression of the 3D video was announced in 2011 [MP11b]. In response to this CfP, over 20 proposals have been submitted in two categories: AVC- and HEVC-compatible. The results of subjective quality assessment of the proposed compression techniques were disclosed at MPEG meeting in Geneva in November 2011. The proposal of Poznań University of Technology, in which methods described in this dissertation have been incorporated, achieved outstanding results. Together with proposals of Fraunhofer Institute – H. Hertz Institute Berlin, it was qualified as the best in HEVC-compatible class.

The 3D video codec proposed by Poznań University of Technology is developed on the top of the MV-HEVC codec, described in Section 2.4.2. As the new HEVC-compatible compression technology is intended for compression of multiview video with corresponding depth or disparity maps, the MV-HEVC has been extended with additional compression tools that utilize available

depth information to encode the side views of multiview video more efficiently [Dom11a, Dom12a, Dom12b]. We will refer to this 3D video codec as HEVC-3D.

In HEVC-3D, one of the views, called the base view, is encoded using the HEVC syntax. As the remaining side views are reconstructed from the base view in the decoder using view synthesis, only the disoccluded regions need to be encoded and transmitted for these views (see Fig. 7.1). The shape of the disoccluded regions is determined based on a view synthesis using reconstructed depth maps. Thus, no additional side information describing this shape needs to be transmitted to decoder.

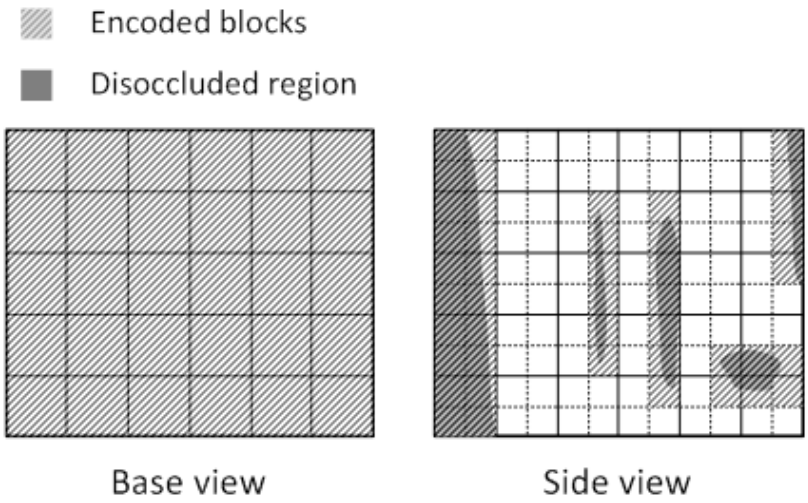


Fig. 7.1. Exemplary image area encoded in HEVC-3D codec for base and side views.

The main idea exploited in HEVC-3D codec is to predict as much information from the base view as possible. Consequently, apart from methods of inter-view prediction introduced in MVC and MV-HEVC, a number of new coding tools utilizing view synthesis approach and prediction with 3D mapping is used to provide higher compression efficiency. These coding tools are:

- Inter-view depth consistency refinement,
- Nonlinear depth representation,
- Depth-dependent adjustment of QP for texture,
- Depth-Gradient-based Loopback Filter (DGLF) and Availability Deblocking Loopback Filter (ADLF) - additional in-loop filters used in side views to reduce the artifacts resulting from coding of disoccluded regions only.
- Depth-Based Motion Prediction (DBMP) – depth-based inter-view prediction of motion information proposed by the author of this thesis.

Apart from the coding tools designed for increasing the compression efficiency of depth maps, the proposed DBMP is the only method used for depth-based inter-view prediction of syntax

elements in HEVC-3D. DBMP is an implementation of the proposed IPDBP motion information prediction algorithm (refer to Section 3.2.2) adopted for HEVC-3D codec. As the syntax of considered 3D video codec is HEVC-compatible, the proposed algorithm is implemented with the same syntax modifications as presented for MV-HEVC codec (see Section 5.3). Based on the experimental results presented in Section 6.4, syntax variant with the proposed depth-based predictor (*DBP*) located at the first position on the merge candidate list has been selected. Also, the unassigned area filling algorithm called *FILLmax* is utilized in DBMP.

As a consequence of outstanding results achieved by HEVC-3D codec, the proposed approach for depth-based inter-view motion information prediction was included in the Test Model under Consideration [ISO11b], which presents the starting point for MPEG's standardization process on HEVC-compatible 3D video coding. Moreover, ideas discussed in this dissertation were also submitted in form of a patent application [Kon10] and are likely to be enclosed in the future standard for 3D video coding.

7.2. Experimental results

In this section, performance of the proposed depth-based inter-view motion information prediction algorithm adopted in HEVC-3D codec in form of DBMP coding tool (refer to Section 7.1) is analyzed. In the experiment, HEVC-3D codec with DBMP enabled (HEVC-3D_DBMPon) [Dom11a] is compared with HEVC-3D codec in which DBMP is turned off (HEVC-3D_DBMPoff). Similarly as in the research on MV-HEVC codec presented in Section 6.4, 2-view and 3-view coding scenarios are investigated. Also, a complete set of multiview test sequences (see Annex A.1) and QP values equal to $\{22,27,32,37\}$ are used.

As in previous experiments, the results are presented for side views only. In 2-view camera setup, encoded side view is denoted as *view* 1. In 3-view camera setup, encoded vies are denoted as *view 1* and *view 2* for central and outermost side views respectively. The arrangement of the cameras in a multiview acquisition system is illustrated in Fig. 2.10b (refer to views c_1 and c_2).

Tab. 7.1 and Tab. 7.2 show results obtained for individual test sequences encoded using 2-view and 3-view case respectively. The improvement in compression performance is measured for HEVC-3D_DBMPon codec and compared with HEVC-3D_DBMPoff using Bjontegaard metrics (see Section 2.2.2). Average changes of bitrate and luminance PSNR (PSNRY) of encoded side

views are presented. Following the approach applied in previous experiments, bitrate required for transmission of depth information is not included in the results.

The results show that, despite the advanced predictions and view synthesis applied in HEVC-3D, the proposed DBMP coding tool can still improve compression performance of the 3D video codec. The bitrate reduction averaged over all considered test sequences in 2-view case is equal to 2.4 [%]. In 3-view case, the average bitrate reductions are equal to 2.6 [%] for *view 1* and 1.1 [%] for *view 2*. However, the coding gains achieved for the bitrate are usually 1.5-3.0 [pp] smaller than the ones observed for MV-HEVC (refer to Tab. 6.7 and Tab. 6.8). This results from the approach used in HEVC-3D, in which only disoccluded regions and their nearest neighborhood are encoded in the side views. As the vast majority of the picture area is synthesized from the reference views, only small number of coding units need to be transmitted to the decoder. Obviously, the amount of information that can be reduced due to utilization of the proposed motion information prediction is decreased in such case.

Despite the smaller coding gains observed for DBMP coding tool when compared to MV-HEVC codec, the analysis of the rate-distortion curves obtained for two analyzed HEVC-3D codecs shows that achieved bitrate savings grow for lower bitrates (see Fig. 7.2). This effect was also noticed for all of the previously investigated multiview video codecs and results from the fact that the cost of encoding motion information is higher for small bitstreams.

To conclude, experimental results presented in this section show that the proposed methods for depth-based inter-view prediction of motion information can be successfully adopted in the future 3D video codecs and contribute to improvement of the 3D video compression.

Tab. 7.1. Performance of HEVC-3D_DBMPon vs. HEVC-3D_DBMPoff codec, calculated using Bjontegaard metrics for QP={22,27,32,37} (2-view case, *view 1*).

Sequence	ΔPSNRY [dB]	ΔBitrate [%]
Poznan Street	0.00	-0.8
Poznan Hall2	0.01	-1.2
Dancer	0.05	-2.7
GT Fly	0.06	-3.2
Kendo	0.04	-5.1
Balloons	0.02	-4.1
Lovebird1	0.00	0.1
Newspaper	0.00	-1.9
Avg.	0.02	-2.4

Tab. 7.2. Performance of HEVC-3D_DBMPon vs. HEVC-3D_DBMPoff codec, calculated using Bjontegaard metrics for QP={22,27,32,37} (3-view case).

Sequence	View 1 (central)		View 2 (outer)	
	ΔPSNRY [dB]	ΔBitrate [%]	ΔPSNRY [dB]	ΔBitrate [%]
Poznan Street	0.00	0.0	0.00	-0.5
Poznan Hall2	0.01	-2.3	0.00	-0.3
Dancer	0.14	-5.8	0.02	-1.0
GT Fly	0.02	-1.1	0.03	-2.0
Kendo	0.01	-4.2	0.00	-2.4
Balloons	0.01	-3.7	0.00	-1.7
Lovebird1	0.00	-2.7	0.00	0.0
Newspaper	0.00	-0.9	0.00	-0.9
Avg.	0.02	-2.6	0.01	-1.1

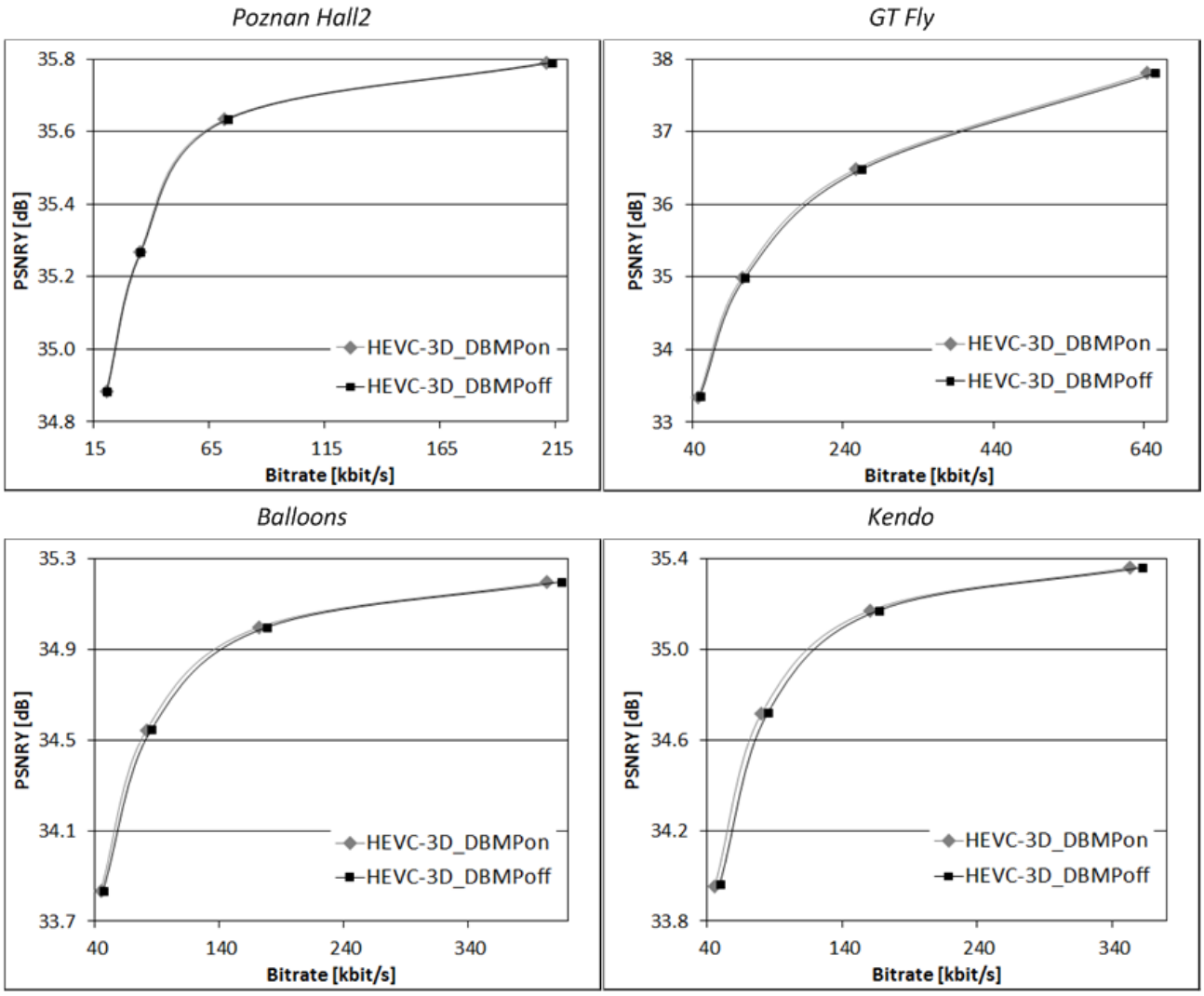


Fig. 7.2. Exemplary rate-distortion curves for HEVC-3D_DBMPon and HEVC-3D_DBMPoff codecs (2-view case, *view 1*).

Chapter 8

Recapitulation and conclusions

8.1. Recapitulation

The author focused his efforts on developing new, efficient methods of representation of motion information for the 3D video coding. The goal was to find algorithms that result in better compression efficiency when compared to the one offered by existing techniques of a motion information representation.

The problem of motion information representation in 3D natural video sequences with additional depth information has been formulated. The problem concerns prediction and coding of motion information in a 3D video codec.

In order to develop efficient solutions for motion information representation in the 3D video coding, the following studies and experiments have been performed:

- Existing multiview video coding techniques have been analyzed, with an emphasis on the most advanced motion information prediction algorithms known from the literature (Chapter 2).
- Weak sides of existing inter-view prediction techniques have been identified (Chapter 3).
- Number of preliminary experiments have been conducted, resulting in development of the depth-based inter-view prediction algorithms proposed by the author: the Forward Projection Depth-Based Prediction (FPDBP) and the Inverse Projection Depth-Based Prediction (IPDBP) (Chapter 3).
- Performance of FPDBP and IPDBP algorithms in case of motion information prediction have been analyzed and compared with the state-of-the-art median predictor (Chapter 4).

The proposed depth-based inter-view prediction algorithms proved to be very accurate for motion information prediction. In particular, the considered prediction methods in many cases perform better than the state-of-the-art median predictor. This encouraged the author to exploit the potential of the proposed algorithms in order to improve the compression efficiency of the state-of-the-art multiview video codecs. Consequently, the following activities have been taken:

- The methods for efficient utilization of the proposed motion information prediction algorithms in MVC and MV-HEVC state-of-the-art multiview video codecs have been developed (Chapter 5).
- On the basis of these methods, FPDBP and IPDBP algorithms have been implemented within the reference anchor software of MVC and MV-HEVC and experimentally tested in order to check their usefulness in the multiview video coding (Chapter 6).

In the design process, a special attention has been given to minimize the complexity of the structure of resultant codec and to reduce the number of required syntax modifications introduced due to implementation of the proposed prediction algorithms.

The FPDBP and IPDBP algorithms developed by the author improve the compression efficiency of the state-of-the-art multiview video codecs. Increase in usage of the low-cost block coding modes causes a compression efficiency improvement for side views of a multiview video sequence. In case of MVC codec, objective distortion measures show average bitrate reduction of 7.7-15.2 [%] for a single side view. On the other hand, the average bitrate reduction measured for MV-HEVC codec is equal to 2.6-5.5 [%]. The coding gains are determined with Bjontegaard metrics [Bjo01] which are commonly used in the Moving Picture Experts Group (MPEG) of the International Organization for Standardization (ISO).

The proposed methods have been experimentally compared with another technique of inter-view motion information prediction, called Motion Skip, which was proposed in MPEG committee during development process of the MVC standard. The compression efficiency achieved by the MVC encoders that used the FPDBP or IPDBP algorithms proposed by the author outperformed the compression efficiency of the Motion Skip algorithm. Average bitrate reductions achieved with the methods proposed by the author were 2.1-7.9 [pp] larger than for the Motion Skip.

In order to evaluate the performance of the proposed prediction algorithms, more than 50 000 hours of experiments have been conducted. The calculations have been performed using one of the most powerful computing servers that were commercially available at the time of experiments.

The depth-based inter-view prediction algorithms proposed by the author were also incorporated into Poznań University of Technology proposal for the 3D video coding technology,

developed as a response for the Call for Proposals [MP11b] announced by the MPEG committee (Chapter 7). The proposed 3D video codec achieved outstanding results and, together with proposals of Fraunhofer Institute – H. Hertz Institute Berlin, was qualified as the best in the HEVC-compatible class. As a result, the approach for motion information prediction proposed by the author was included in the Test Model under Consideration [ISO11b], which presents the starting point for MPEG's standardization process on HEVC-compatible 3D video coding. Consequently, the ideas proposed in this dissertation are likely to be enclosed in the emerging standard for the 3D video coding.

It is worth noticing that the proposed FPDBP and IPDBP techniques were the first inter-view algorithms that utilize depth information for accurate point-to-point prediction of motion information. They were also one of the first successful algorithms of inter-view motion information prediction presented for the HEVC-compatible 3D video codec.

The goal of the dissertation was to develop new techniques for compression of 3D video in the multiview video plus depth format. The author has proposed two methods of inter-view prediction in 3D video: FPDBP and IPDBP. It has been shown that the proposed predictors allow to reduce the bitrate compared to the currently known systems by providing the more efficient representation of motion information and utilization of available depth information. The proposed techniques have been researched and experimentally evaluated to accurately assess their actual impact on the coding efficiency of existing and future multiview video codecs.

The theses stated in the dissertation have been proven. It has been proven that the proposed depth-based inter-view prediction algorithms improve the efficiency of motion information representation in coding of 3D video in the multiview video plus depth format. It has also been proven that the proposed techniques of representation of motion information are competitive to the methods described in the literature and developed simultaneously with the author's investigations.

8.2. Original achievements of the dissertation

The **main achievements**:

- Development of original depth-based inter-view motion information prediction algorithms for the 3D video coding: Forward Projection Depth-Based Prediction (FPDBP) and Inverse Projection Depth-Based Prediction (IPDBP) techniques (Section 3.2).

- Proposal of original algorithms of motion information representation in the state-of-the-art multiview video codecs: MVC and MV-HEVC (Chapter 5).
- Experimental evaluation of different signaling strategies for the proposed FPDBP and IPDBP prediction techniques (Section 6.3.1 and Section 6.4).
- Experimental evaluation of the proposed prediction techniques against MVC and MV-HEVC multiview video codecs (Section 6.3.2 and Section 6.4).
- Experimental comparison of the proposed FPDBP and IPDBP techniques against Motion Skip mode of JMVM (Section 6.3.2).
- Development of original algorithm of motion information representation for an emerging 3D video coding standard and its experimental evaluation (Chapter 7).

Other original results of the dissertation:

- Experimental investigations on the participation of individual visual stream components in the state-of-the art multiview video coder MVC (Section 4.1).
- Comparison of the accuracy of the proposed depth-based inter-view motion information predictors with the state-of-the-art median predictor (Section 4.3).
- Proposal of the *o-mask* occlusion detection algorithm (Section 3.3).
- Adaptation of the *z-test* algorithm for fast and simple occlusion detection (Section 3.3).
- Development of simple unassigned area filling methods: *FILLmax*, *FILLmin* and *FILLsim* (Section 3.4).
- Experimental evaluation of the influence of depth quality on performance of the proposed algorithms of motion information representation in the state-of-the-art multiview video codec MVC (Section 6.3.3).
- Experimental investigations on the complexity of proposed algorithms for motion information representation in MVC and MV-HEVC multiview video codecs (Section 6.5).

8.3. General conclusions

The thesis of this dissertation states that exploiting the correlation between motion fields of neighboring views in a multiview video sequence and utilizing the available depth information describing the visual scene should improve the efficiency of a motion information representation in a multiview video coding.

In the modern video coders, motion information, together with control data, constitutes a significant part of the bitstream. Experimental results obtained by the author show that in case of multiview video encoding, the abovementioned syntax elements become a dominant part of the bitstream for the side views of a multiview sequence, especially when the lower bitrates are considered. These observations show clearly that development of more efficient ways of representing motion information is expedient. Improvements in this field should result in increasing the coding efficiency of the multiview video codecs.

As a consequence, dedicated methods for efficient motion information representation in a multiview video bitstream have been developed in recent years. In particular, inter-view correlation between motion field of the neighboring views of a multiview sequence have been exploited. However, the techniques proposed previously suffered from utilization of a global disparity vector or simple block disparity estimation as the determinants of correspondence between different views. Such approach may lead to significant errors, especially if unified value of the global disparity is used to describe disparity between areas of the visual scene that have very different distance from the camera. Moreover, the usage of local disparity assigned to image blocks may also cause inaccuracies, as it does not preserve depth discontinuities on object borders. On the other hand, availability of additional depth information enables utilization of the 3D-space modeling techniques based on the Depth Image Based Rendering (DIBR), known from computer graphics. This way, a mapping can be done independently for each point of encoded image. Consequently, two techniques of point-to-point depth-based inter-view prediction of motion information have been proposed by the author and presented in this dissertation.

The obtained experimental results proved that the proposed improvement of motion information representation leads to a noticeable increase in compression efficiency of the state-of-the-art multiview video codecs. The modified MVC coder with proposed inter-view predictors outperforms the original MVC coder reducing the bitstream by 7.7-15.2 [%] on average. In author's opinion it is a very good result. For comparison, in concurrent technique of inter-view motion information prediction, called the Motion Skip, which was proposed in the MPEG committee during development process of the MVC standard, coding gains are equal to approximately 5.6-7.3 [%]. Significant coding gains are also observed in the new generation multiview video codec, which is based on the emerging HEVC coding technology. The achieved average bitrate reduction is equal to 2.6-5.5 [%].

On the basis of experimental investigations of the proposed depth-based inter-view predictors, the following conclusions were drawn:

- The considered prediction methods in many cases perform better than the state-of-the-art median predictor.
- Utilization of the proposed prediction algorithms results in higher usage of block coding modes that do not require transmitting motion information to the decoder and preserves some of encoded blocks from further partitioning into smaller parts. This leads to improved compression efficiency of the codec.
- Bitrate savings achieved due to usage of the proposed prediction techniques grow for lower bitrates, as the cost of encoding motion information is higher for small bitstreams.
- Decreasing the depth quality has generally a negative impact on performance of the proposed prediction algorithms, reducing the compression efficiency of a codec. However, the impact on achieved coding gains is only slightly.
- The influence of proposed prediction techniques on performance of the state-of-the-art multiview video codecs is very similar.

The experimental evaluation of efficiency of the proposed depth-based inter-view algorithms in representing motion information leads to a conclusion that techniques proposed by the author could be also successfully adopted to prediction of other syntax elements in multiview video coding. This is one of the possible areas for future research.

A new 3D video coding standard is now intensively developed. The techniques of depth-based inter-view prediction of motion information presented in this dissertation were included in the Test Model under Consideration [ISO11b] which is the starting point for MPEG's standardization process on the HEVC-compatible 3D video coding. As a consequence, the approach proposed by the author is likely to be enclosed in the future standard for the 3D video coding.

References

[Adam01] A. Adam, E. Rivlin, I. Shimshoni, "ROR: rejection of outliers by rotations", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. l, pp. 78-84, January 2001.

[ANSI03] ANSI T1.801.03, American National Standard for Telecommunications: "Digital transport of one-way video signals. Parameters for objective performance assessment", American National Standards Institute, 2003.

[Berg00] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, "Computational geometry", 2nd Edition, Springer-Verlag, 2000.

[Bert00] M. Bertalmío, G. Sapiro, V. Caselles, C. Ballester, "Image inpainting", Proceedings of SIGGRAPH 2000, New Orleans, USA, July 2000.

[Bjo01] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", VCEG Contribution VCEG-M33, Austin, USA, April 2001.

[Bos10] F. Bossen, V. Drugeon, E. Francois et al., "Video coding using a simplified block structure and advanced coding techniques", IEEE Trans. Circuits Syst. Video Technol., Vol. 20, No. 12, pp. 1667-1675, December 2010.

[Bos11] F. Bossen, "Common test conditions and software reference configurations", ISO/IEC JTC1/SC29/WG11, JCTVC-F900, Geneva, Switzerland, July 2011.

[Bros11] B. Bross, J. Jung, Y.-W. Huang, Y. H. Tan, I.-K. Kim, T. Sugio, M. Zhou, T. K. Tan, E. F. K. Kazui, W.-J. Chien, S. Sekiguchi, S. Park, W. Wan, "BoG report of CE9: MV coding and skip/merge operations", ITU-T and ISO/IEC JTC1 Doc. JCTVC-E481, March 2011.

[Chen07] Y.-W. Chen , W.-H. Peng , S.-Y. Lee, "Adaptive macroblock and motion skip flags for Multiview Video Coding (MVC)", Joint Video Team (JVT), Doc. JVT-X047, Geneva, Switzerland, June 2007.

[Chen09] Y. Chen, P. Pandit, S. Yea, "WD 4 reference software for MVC", Joint Video Team (JVT), Doc. JVT-AD207, Geneva, Switzerland, February 2009.

[Chen09a] Y. Chen, M.M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj, H. Li, "Coding techniques in Multiview Video Coding and Joint Multiview Video Model", Picture Coding Symposium, 2009. PCS 2009, Chicago, USA, May 2009.

[Cyg02] B. Cyganek, „Komputerowe przetwarzanie obrazów trójwymiarowych", Exit 2002.

[Cyg09] B. Cyganek, J. P. Siebert, „An introduction to 3D computer vision techniques and algorithms", Wiley 2009.

[Davi97] E. R. Davies, "Machine vision. Theory, algorithms, practicalities", 2nd Edition, Academic Press, 1997.

[Deve95] F. Devernay, O. Faugeras, "From projective to euclidean reconstruction", INRIA Technical Report No. 2725, 1995.

[Ding07] L.-F. Ding, P.-K. Tsung, S.-Y. Chien, W.-Y. Chen, L.-G. Chen, "Computation-free motion estimation with inter-view mode decision for Multiview Video Coding", IEEE 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2007, Kos Island, Greece, May 2007.

[Ding08] L.-F. Ding, P.-K. Tsung, W.-Y. Chen, S.-Y. Chien, L.-G. Chen, "Fast motion estimation with inter-view motion vector prediction for stereo and multiview video coding", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2008, pp. 1373-1376, Las Vegas, USA, April 2008.

[Ding08a] L.-F. Ding, P.-K. Tsung, S.-Y. Chien, W.-Y. Chen, L.-G. Chen, “Content-aware prediction algorithm with inter-view mode decision for Multiview Video Coding”, IEEE Transactions on Multimedia, Vol. 10, No. 8, pp. 1553-1564, December 2008.

[Dom98] Domański M., "Zaawansowane techniki kompresji obrazów i sekwencji wizyjnych", Wydawnictwo Politechniki Poznańskiej, Poznań 1998.

[Dom09] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner, „Poznań multiview video test sequences and camera parameters", ISO/IEC JTC1/SC29/WG11 MPEG/M17050, Xian, China, October 2009.

[Dom10] M. Domański, „Obraz cyfrowy”, Wydawnictwo Komunikacji i Łączności, 2010.

[Dom11] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, K. Wegner, “Multiview HEVC – experimental results”, ISO/IEC JTC1/SC29/WG11, JCTVC-G582, Geneva, Switzerland, November 2011.

[Dom11a] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, K. Wegner, “Technical description of Poznań University of Technology proposal for call on 3d video coding technology”, ISO/IEC JTC1/SC29/WG11, MPEG/M22697, Geneva, Switzerland, November 2011.

[Dom12] M. Domański, „HEVC – nowa generacja technik kompresji obrazu”, Przegląd Telekomunikacyjny No. 4/2012, vol. LXXXV, p. 103, 2012.

[Dom12a] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, K. Wegner, „New coding technology for 3D video with depth maps as proposed for standardization within MPEG”, 19th International Conference on Systems, Signals and Image Processing (IWSSIP), 2012, pp. 401-404, Vienna, Austria, April 2012.

[Dom12b] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, K. Wegner, „Coding of multiple video+depth using HEVC Technology and reduced representations of side views and depth maps”, Picture Coding Symposium (PCS), 2012 IEEE, pp. 5-8, Kraków, Poland, May 2012.

[Eri85] S. Ericsson, “Fixed and adaptive predictors for hybrid predictive/transform coding”, IEEE Transactions on Communications, Vol. COM-33, No. 12, pp. 1291-1302, December 1985.

[Faug93] O. Faugeras, “Three-dimensional computer vision: a geometric viewpoint”, MIT Press, London, UK, ISBN 0–262–06158–9, 1993.

[Faug01] O. D. Faugeras, Q.-T. Luong, “The geometry of multiple images”, MIT Press, 2001.

[Feck05] U. Fecker, A. Kaup, “Statistical analysis of multi-reference block matching for dynamic light field coding”, 10th International Fall Workshop - Vision, Modeling, and Visualization (VMV), pp. 445-452, Erlangen, Germany, November 2005.

[Feld08] I. Feldmann, M. Mueller, F. Zilly, R. Tanger, K. Mueller, A. Smolic, P. Kauff, T. Wiegand, “HHI test material for 3d video”, ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15413, Archamps, France, April 2008.

[Flier04] M. Flierl, B. Girod, "Video coding with superimposed motion-compensated signals: applications to H.264 and beyond", Kluwer Academic Publishers, Dordrecht 2004.

[Flier07] M. Flierl, A. Mavlankar, B. Girod, “Motion and disparity compensated coding for multiview video”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 11, pp. 1474-1484, 2007.

[Fol94] J. D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, R.L. Phillips, “Introduction to computer graphics”, Addison-Wesley, 1994.

[Fran06] R. Fransens, C. Strecha, L. Van Gool, "A mean field EM-algorithm for coherent occlusion handling in MAP-estimation problems", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, Vol. 1, pp. 300-307, New York, USA, June 2006.

[Fu11] C.-M. Fu et al., "CE13: sample adaptive offset with LCU-independent decoding", in Tech. Rep. JCTVC- E049, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Geneva, Switzerland, March 2011.

[Guo06] X. Guo, Y. Lu, F. W, W. Gao, "Inter-view direct mode for Multiview Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 12, pp. 1527–1532, December 2006.

[Han10] W.-J. Han, J. Min, I.-K. Kim et al., "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools", IEEE Trans. Circuits Syst. Video Technol., Vol. 20, No. 12, pp. 1709–1720, December 2010.

[Hart00] R. I. Hartley, A. Zisserman, "Multiple view geometry in computer vision", Cambridge University Press, 2000.

[Hech98] E. Hecht, "Optics (3rd edition)", Addison Wesley, ISBN 0-201-30425-2, 1998.

[Heik00] J. Heikkila, "Geometric camera calibration using circular control points", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, pp. 1066-1077, October 2000.

[Ho08] Y.-S. Ho, E.-K. Lee, C. Lee, "Multiview video test sequence and camera parameters", ISO/IEC JTC1/SC29/WG11 MPEG/M15419, Archamps, France, April 2008.

[Huo11] Junyan Huo, Yilin Chang, Yanzhuo Ma, "Efficient prediction structure for key pictures in Multiview Video Coding", Symposium on Photonics and Optoelectronics (SOPO), May 2011.

[Hur07] J. H. Hur, S. Cho, Y. L. Lee, "Adaptive local illumination change compensation method for H.264/AVC-based Multiview Video Coding", IEEE Trans. Circuits Syst. Video Technol., Vol. 17, No. 11, pp. 1496–1505, November 2007.

[ISO10] ISO/IEC JTC1/SC29/WG11, "Joint call for proposals on video compression technology", Doc. N11113, Kyoto, Japan, January 2010.

[ISO11] ITU-T and ISO/IEC JTC1, "Advanced video coding for generic audiovisual services", ITU-T Rec. H.264 – ISO/IEC 14496-10 AVC, 2011.

[ISO11a] ISO/IEC MPEG, "Description of exploration experiments in 3d video coding", MPEG N12037, March 2011.

[ISO11b] ISO/IEC MPEG, "Technology under consideration for HEVC based 3D video coding", MPEG N12350, December 2011.

[ISO93] ISO/IEC 11172-2, "Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, part 2: video" (MPEG-1), November 1993.

[ISO94] ISO/IEC 13818-2/ITU-T Rec. H.262, "Generic coding of moving pictures and associated audio, part 2: video" (MPEG-2), November 1994.

[ISO10] ISO/IEC JTC1/SC29/WG11, "Report on experimental framework for 3d video coding", Doc. N11631, Guangzhou, China, October 2010.

[ITUR97] ITU-R Rec. BT. 1128, "Subjective assessment of conventional television systems", 1997.

[ITUR98] ITU-R Rec. BT. 1129, "Subjective assessment of digital television systems at or near the quality of conventional systems", 1998.

[ITUR98a] ITU-R Rec. BT. 710-4, "Subjective assessment methods for image quality in high-definition television", 1998.

[ITUR03] ITU-R Rec. BT. 500-11, "Methodology for the subjective assessment of the quality of television pictures", 2003.

[ITUR04] ITU-R Rec. BT. 1683, “Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference”, 2004.
[ITUT04] ITU-T Rec. J. 144, “Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference”, 2004.
[ITUT05] ITU-T Rec. H.263, "Video coding for low bit rate communication", August 2005.
[Iyer10] K. N. Iyer, K. Maiti, B. Navathe, H. Kannan, A.Sharma, "Multiview video coding using depth based 3D warping", IEEE International Conference on Multimedia and Expo (ICME), 2010, pp.1108-1113, July 2010.
[Jai81] J. R. Jain, A. K. Jain, "Displacement measurement and its application in inter-frame image coding", IEEE Transactions on Communications, Vol. 29, No. 12, pp. 1799-1808, December 1981.
[Jav06] B. Javidi, F. Okano (guest editors), “Special issue on 3-D technologies for imaging and display”, Proc. of the IEEE, Vol. 94, pp. 487-663, 2006.
[Jeon08] Yong-Joon Jeon, Byeong-Moon Jeon, “High level syntax for motion skip mode”, ITU-T and ISO/IEC JTC1, Doc. JVT-AA031, Geneva, Switzerland, April 2008.
[Karc10] M. Karczewicz, P. Chen, R. Joshi et al., “A hybrid video coder based on extended macroblock sizes, improved interpolation, and flexible motion representation”, IEEE Trans. Circuits Syst. Video Technol., Vol. 20, No. 12, pp. 1698–1708, December 2010.
[Kauf07] P. Kauff, N. Atzpadin, C. Fehn, M. Mueller, O. Schreer, A. Smolic, R. Tanger, “Depth Map Creation and Image Based Rendering for Advanced 3DTV Services Providing Interoperability and Scalability”, Signal Processing: Image Communication, Vol. 22, No. 2, Special Issue on 3D Video and TV, pp. 217-234, February 2007.
[Kaup06] A. Kaup, U. Fecker, “Analysis of multi-reference block matching for Multiview Video Coding”, in Proc. 7th Workshop Digital Broadcasting, pp. 33-37, Erlangen, Germany, September 2006.
[Kit06] M. Kitahara, H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, K. Yamamoto, T. Yendo, T. Fujii, M. Tanimoto, “Multi-view video coding using view interpolation and reference picture selection”, in Proc. IEEE Int. Conf. Multimedia Expo, Toronto, Canada, July 2006.
[Klet98] R. Klette, K. Schliins, A. Koschan, “Computer vision. Three-dimensional data from images”, Springer-Verlag 1998.
[Kog81] T. Koga, K. Iinuma, A. Hirano, A. Iijima, T. Ishiguro, “Motion-compensated inter-frame coding for video conferencing”, Proc. of National Telecommunications Conference, Vol. 4, 1981.
[Kol95] C. Kolb, D. Mitchell, P. Hanrahan, “A realistic camera model for computer graphics”, Technical Report, Computer Science Department, Princeton University, 1995.
[Kon10] J. Konieczny, M. Domański, “Sposób kodowania informacji o ruchu w sekwencjach wielowidokowych z dostępną informacją o rozbieżności lub głębi stereoskopowej”, patent application P.390327, 2010.
[Kon10a] J. Konieczny, M. Domański, “Depth-based inter-view prediction of motion vectors for improved Multiview Video Coding”, IEEE 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010, Tampere, Finland, June 2010.
[Kon10b] J. Konieczny, M. Domański, “Inter-view direct mode for Multiview Video Coding”, ISO/IEC JTC1/SC29/WG11, MPEG2010/M17800, Geneva, Switzerland, July 2010.
[Kon11] J. Konieczny, M. Domański, “Extended inter-view direct mode for Multiview Video Coding”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 845-848, Prague, Czech Republic, May 2011.
[Kon11a] J. Konieczny, M. Domański, “Depth-enhanced compression for 3D video”, Video Coding and Image Processing (VCIP), 2011 IEEE, Tainan, Taiwan, November 2011.

[Kon12] J. Konieczny, M. Domański, "Depth-based inter-view motion data prediction for HEVC-based multiview video coding", Picture Coding Symposium (PCS), 2012 IEEE, pp. 33-36, Kraków, Poland, May 2012.

[Koo06] H. S. Koo, Y. J. Jeon, B. M. Jeon, "Motion skip mode for MVC", Joint Video Team (JVT), Doc. JVT-U091, Hangzhou, China, October 2006.

[Koo07] H. S. Koo, Y. J. Jeon, B. M. Jeon, "MVC motion skip mode", Joint Video Team (JVT), Doc. JVT-W081, San Jose, USA, April 2007.

[Koo08] H. S. Koo, Y. J. Jeon, B. M. Jeon, "Motion information inferring scheme for multi-view video coding", IEICE Trans. Commun., pp. 1247–1250, 2008.

[Kri97] R. Krishnamurthy, "Compactly-encoded optical flow fields for motion compensated video compression and processing", doctoral thesis, Rensselaer Polytechnic Institute Troy, New York, 1997.

[Lai07] P. Lai, A. Ortega, P. Pandit, P. Yin, C. Gomila, "Adaptive reference filtering for MVC", Joint Video Team (JVT) Doc. JVT-W065, San Jose, USA, April 2007.

[Lai08] P. Lai, A. Ortega, P. Pandit, P. Yin, C. Gomila, "Focus mismatches in multiview systems and efficient adaptive reference filtering for multiview video coding", in Proc. SPIE Conf. Vis. Commun. Image Process., Vol. 6822, San Jose, USA, January 2008.

[Lang04] Rafał Lange, Łukasz Błaszak, Marek Domański, "Simple AVC-Based Codecs with Spatial Scalability", International Conference on Image Processing (ICIP), 2004 IEEE, Singapore, October 2004.

[Lang06] R. Lange, "Multiresolution representation of motion vectors in video compression", doctoral thesis, Poznań, 2006.

[Lav94] S. Laveau and O. Faugeras, "3-D scene representation as a collection of images", International Conference on Pattern Recognition, Vol. 1, pp. 689–691, Jerusalem, Israel, October 1994.

[Lee06] Y. L. Lee, J. H. Hur, Y. K. Lee, K. H. Han, S. H. Cho, N. H. Hur, J. W. Kim, J. H. Kim, P. Lai, A. Ortega, Y. Su, P. Yin, C. Gomila, "CE11: illumination compensation", Joint Video Team (JVT) Doc. JVT-U052, Hangzhou, China, 2006.

[Lee10] E.K. Lee, Y.K. Jung, and Y.S. Ho, "3-D video generation using foreground separation and disocclusion detection", IEEE 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010, Tampere, Finland, May 2010.

[Lin07] S. Lin, S. Gao, H. Yang, L. Xiong, "RDV based MVC motion skip mode", Joint Video Team (JVT), Doc. JVT-Y036, Shenzhen, China, October 2007.

[List03] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, "Adaptive deblocking filter", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 614 –619, July 2003.

[Marp06] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications", IEEE Commun. Mag., Vol. 44, No. 8, pp. 134–144, August 2006.

[Marp10] D. Marpe , H. Schwarz, S. Bosse et al., "Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding", IEEE Trans. Circuits Syst. Video Technol., Vol. 20, No. 12, pp. 1676–1687, December 2010.

[Mart06] E. Martinian, A. Behrens, J. Xin, A. Vetro, "View synthesis for multiview video compression", in Proc. Picture Coding Symposium PCS, Beijing, China, 2006.

[Mat10] R. Mathew, D. S. Taubman, "Quad-tree motion modeling with leaf merging", IEEE Trans. Ciruits and Syst. Video Technol., Vol. 20, No. 10, pp. 1331–1345, October 2010.

[McC11] K. McCann et al., “HM4: HEVC test model 4 encoder description”, Tech. Rep. JCTVC-F802, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Torino, Italy, July 2011.

[McMi97] L. McMillan, “An image-based approach to three-dimensional computer graphics”, Doctoral thesis, University of North Carolina, Chapel Hill, USA, April 1997.

[Mend09] B. Mendiburu, “3D movie making – stereoscopic digital cinema from script to screen”, Focal Press, Burlington MA 2009.

[Merk07] P. Merkle, A. Smolic, K. Muller, T. Wiegand, "Efficient prediction structures for Multiview Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 11, pp.1461-1473, Nov. 2007.

[Merk08] P. Merkle, Y. Morvan, A. Smolic, K. Mueller, P. H. de With, T. Wiegand, “The effect of depth compression on multi-view rendering quality”, IEEE 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2008, pp. 245-248, Istanbul, Turkey, May 2008.

[Moh96] R. Mohr, B. Triggs, “Projective geometry for image analysis”, tutorial given at ISPRS, Vienna, July 1996.

[MP08] “Joint multiview video model (JMVM) 8.0”, JVT-AA207, Geneva, Switzerland, April 2008.

[MP11] “WD3: working draft 3 of High-Efficiency Video Coding”, ISO/IEC JTC1/SC29/WG11, JCTVC-E603, Geneva, Switzerland, March 2011.

[MP11a] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-3.0rc2/, HEVC reference software, HM 3.0.

[MP11b] “Call for proposals on 3d video coding technology”, ISO/IEC JTC1/SC29/WG11, MPEG2011/N12036, Geneva, Switzerland, March 2011.

[Mun92] J. L. Mundy, A. Zisserman, “Geometrie invariance in computer vision”, MIT Press, Cambridge, Massachusetts, 1992.

[Nin82] Y. Ninomiya, Y. Ohtsuka, “A motion-compensated inter-frame coding scheme for television pictures”, IEEE Transactions on Communications, Vol. COM-30, No. 1, pp. 201-211, January 1982.

[Ohm04] J.–R. Ohm, “Multimedia communication technology. representation, transmission and identification of multimedia signals”, Springer, 2004.

[Oja03] V. Ojansivu, O. Silven, R. Huotari, "A technique for digital video quality evaluation", Proc. IEEE International Conference on Image Processing ICIP'03, Vol. 3, pp. 181-184, Barcelona, Spain, 2003.

[Oliv00] M. M. Oliveira, “Relief texture mapping”, Doctoral thesis, University of North Carolina, Chapel Hill, USA, March 2000.

[Ort98] A. Ortega, K. Ramchandran, “Rate-distortion methods for image and video compression: an overview”, IEEE Signal Processing Magazine, Vol. 15, Issue 6, pp. 23-50, November 1998.

[Oud11] S. Oudin, P. Helle, J. Stegemann, C. Bartnik, B. Bross, D. Marpe, H. Schwarz, T. Wiegand, "Block merging for quadtree-based video coding", Multimedia and Expo (ICME), 2011 IEEE International Conference on, July 2011.

[Pan08] P. Pandit, A. Vetro, Y. Chen, ”JMVM 8 software”, ITU-T and ISO/IEC JTC1, JVT-AA208, Geneva, Switzerland, April 2008.

[Pas06] R. Pastrana-Vidal, J.-Ch. Gicquel, “Automatic quality assessment of video fluidity impairments using a no-reference metric”, Proc. of International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-06, Scottsdale 2006.

[Pat07] S. Pateux, “An excel add-in for computing Bjontegaard metric and its evolution”, VCEG Contribution VCEG-AE07, Marrakech, January 2007.

[Pedr98] L. S. Pedrotti, F.L. Pedrotti, “Optics and vision”, Prentice-Hall, 1998.

[Pfis04] H. Pfister, A. Vetro, H. Sun, “3D TV system”, ISO/IEC JTC1/SC29/WG11 MPEG04/M10624, Munich, Germany, March 2004.

[Pre07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, “Numerical Recipes. The Art of Scientific Computing”, 3rd Edition, New York: Cambridge University Press, 2007.

[Ric02] I. Richardson, “Video codec design”, John Wiley & Sons Ltd., Chichester 2002.

[Ric10] I. Richardson, “The H.264 Advanced Video Compression standard second edition”, John Wiley & Sons, Chichester 2010.

[Rusa11] D. Rusanovskyy, P. Aflaki, M. M. Hannuksela, "Undo Dancer 3DV sequence for purposes of 3DV standardization", ISO/IEC JTC1/SC29/WG11 MPEG/M20028, Geneva, Switzerland, March 2011.

[Ryu11] S. Ryu, J. Seo, J. Y. Lee, K. Sohn, “Advanced motion vector coding framework for multiview video sequences”, Multimedia Tools and Applications, Springer Netherlands, 2011.

[Sad02] A. H. Sadka, “Compressed video communications”, John Wiley & Sons Ltd., Chichester, 2002.

[Sem98] J. G. Semple, G. T. Kneebone, “Algebraic projective geometry (Oxford classic texts in the physical sciences)”, Oxford University Press, 3rd edition, 1998.

[Sha98] J. Shade, S. Gortler, L. wei He, R. Szeliski, “Layered depth images”, International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), pp. 231–242. ACM Press, 1998.

[Shan97] R. R. Shannon, “The art and science of optical design”, Cambridge University Press, 1997.

[Shi00] Y. Q. Shi, H. Sun, "Image and video compression for multimedia engineering", CRC Press, Boca Raton 2000.

[Shim07] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, Y. Yashima, “View scalable multiview video coding using 3D warping with depth map”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17 (11), pp. 1485–1495, 2007.

[Ska93] W. Skarbek, “Metody reprezentacji obrazów cyfrowych”, Warszawa, 1993.

[Ska98] W. Skarbek (ed), "Multimedia. algorytmy i standardy kompresji", Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998.

[Smo06] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, T. Wiegand, “3D Video and Free Viewpoint Video – Technologies, Applications and MPEG Standards,” Proc. of ICME 2006, pp. 2161-2164, Toronto, Canada, July 2006.

[Smo07] A. Smolic, K. Mueller, P. Merkle, N. Atzpadin, C. Fehn, M. Mueller, O. Schreer, R. Tanger, P. Kauff, T. Wiegand, “Multi-view video plus depth (MVD) format for advanced 3D video systems”,JVT-W100, San Jose, USA, April 2007.

[Song07] H.-S. Song, W.-S. Shim, Y.-H. Moon, J.-B. Choi, “Macroblock information skip for MVC”, Doc. JVT-V052, Marrakech, Morocco, January 2007.

[Sta12] J. Stankowski, M. Domański, O. Stankiewicz, J. Konieczny, J. Siast, K. Wegner, “Extensions of the HEVC technology for efficient Multiview Video Coding”, IEEE International Conference on Image Processing (ICIP), 2012, Orlando, USA, October 2012, in print.

[Str96] D. Strachan, “Video compression”, The Society of Motion Picture and Television Engineers Journal, Vol. 105, pp. 68-73, February 1996.

[Su06] Y. Su, A. Vetro, A. Smolic, “Common test conditions for multiview video coding”, JVT-T207, Klagenfurt, Austria, 2006.

[Sue10] K. Suehring, J.-R. Ohm, “Results of breakout on decoder speed”, JCT-VC-A201, Dresden, Germany, April 2010.

[Sul98] G. J. Sullivan, T. Wiegand, "Rate-distortion optimization for video compression", IEEE Signal Processing Magazine, Vol. 15, Issue 6, pp. 74-90, November 1998.

[Sul05] G. J. Sullivan and T. Wiegand, “Video compression - from concepts to the H.264/AVC standard”, Proc. IEEE, Vol. 93, No. 1, pp. 18–31, January 2005.

[Sul10] G. J. Sullivan, J.-R. Ohm, “Meeting report of the first meeting of the Joint Collaborative Team on Video Coding”, JCT-VC-A200, Dresden, Germany, April 2010.

[Sun05] J. Sun, Y. Li, S. B. Kang, H.-Y. Shum, "Symmetric stereo matching for occlusion handling", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 399-406, San Diego, USA, June 2005.

[Tan08] TK Tan, G. Sullivan, T. Wedi, “Recommended simulation conditions for coding efficiency experiments”, VCEG Contribution VCEG-AJ10, San Diego, July 2008.

[Tani08] M. Tanimoto, T. Fujii, N. Fukushima, “1D parallel test sequences for MPEG-FTV”, ISO/IEC JTC1/SC29/WG11 MPEG2008/M15378, Archamps, France, April 2008.

[Tani10] M. Tanimoto, T. Fujii, M. P. Tehrani, M. Wildeboer, "3DV/FTV EE1 report on Kendo and Balloons sequences", ISO/IEC JTC1/SC29/WG11 MPEG/M17500, Dresden, Germany, April 2010.

[Thor05] T. Thormahlen, H. Broszio, “Automatic line-based estimation of radial lens distortion”, Integrated Computer-Aided Engineering, Vol. 12 (2), pp. 177–190, 2005.

[Tok12] M. Tok, A. Glantz, A. Krutz, T. Sikora, "Parametric motion vector prediction for hybrid video coding", Picture Coding Symposium (PCS), 2012, pp. 381-384, Kraków, Poland, May 2012.

[Tou05] A. M. Tourapis, F. Wu, and S. Li, “Direct mode coding for bi-predictive slices in the H.264 standard”, IEEE Trans. Circuits and Syst. Video Technol., Vol. 15, No. 1, pp. 119–126, January 2005.

[Truc98] E. Trucco, A. Verri, “Introductory techniques for 3-D computer vision”, Prentice-Hall, 1998.

[Um08] G.-M. Um, G. Bang, N. Hur, J. Kim, Y.-S. Ho, “3D video test material of outdoor scene”, ISO/IEC JTC1/SC29/WG11 MPEG/M15371, Archamps, France, April 2008.

[Vet07] A. Vetro, S. Yea, W. Matusik, H. Pfister, M. Zwicker, “Antialiasing for 3D displays”, ITU-T and ISO/IEC JTC1, Doc. JVT-W060, San Jose, USA, April 2007.

[Vet07a] A. Vetro, P. Pandit, H. Kimata, A. Smolic, “Joint Multiview Video Model (JMVM) 5.0”, ITU-T and ISO/IEC JTC1, Doc. JVT-X207, Geneva, Switzerland, July 2007.

[Vet11] A. Vetro, T. Wiegand, G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard", Proceedings of the IEEE , Vol.99, No.4, pp.626-642, April 2011.

[Wie03] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264/AVC video coding standard”, IEEE TCSVT, Vol. 13, No. 7, pp. 560-576, July 2003.

[Wie10] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, K. Ugur, “Special section on the joint call for proposals on High Efficiency Video Coding (HEVC) standardization”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 20, No. 12, pp. 1661-1666, December 2010.

[Wie11] T. Wiegand, B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, “WD3: working draft 3 of High-Efficiency Video Coding”, ISO/ITU JCT-VC, Doc. JCTVC-E603, Geneva, Switzerland, March 2011.

[Win10] M. Winken , S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, D. Marpe, S. Oudin, M. Preiss, H. Schwarz, M. Siekmann, K. Suehring, T. Wiegand, “Description of video coding technology proposal by Fraunhofer HHI”, Doc. JCT-VC-A116, Dresden, Germany, April 2010.

[Wink05] S. Winkle, "Digital video quality. vision models and metrics", John Wiley & Sons Ltd., Chichester, 2005.

[Wink10] S. Winkle, F. Dufaux, D. Barba, V. Barocini (guest ed.), "Special issue on image and video quality assessment", Signal Processing: Image Communications, Vol. 25, Issue 7, pp. 467-558, August 2010.

[Wol90] G. Wolberg, “Digital image warping”, IEEE Computer Society Press, July 1990.

[Yan05] X. K. Yang, W. Lin, Z. K. Lu, E. P. Ong, S. S. Yao, “Just noticeable distortion model and its applications in video coding”, Signal Processing: Image Communication, Vol. 20, Issue 7, pp. 662-680, 2005.

[Yang07] H. Yang, J. Huo, Y. Chang, S. Lin, S. Gao, L. Xiong, “Inter-view motion skipped multi-view video coding with fine motion matching”, Doc. JVT-Y037, Shenzhen, China, October 2007.

[Yang08] H. Yang, Y. Chang, J. Huo, S. Lin, S. Gao, L. Xiong, “CE1: Fine motion matching for motion skip mode in MVC”, ITU-T and ISO/IEC JTC1, Doc. JVT-Z021, Antalya, Turkey, January 2008.

[Yang09] H. Yang, Y. Chang, J. Huo, “Fine-granular motion matching for inter-view motion skip mode in Multiview Video Coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19 (6), pp. 887–892, 2009.

[Yea07] S. Yea, A. Vetro, M. Zwicker, “MVC showcase for SEI on scene and acquisition info”, ITU-T and ISO/IEC JTC1, JVT-X074, Geneva, Switzerland, June 2007.

[Yea09] S. Yea, A. Vetro, “View synthesis prediction for multiview video coding”, Image Commun., Vol. 24, No. 1–2, pp. 89–100, January 2009.

[Youn94] T. Y. Young, “Handbook of pattern recognition and image processing: computer vision”, Vol. 2, Academic Press 1994.

[Zha11] J. Zhang, R. Li, H. Li, D. Rusanovskyy, M. M. Hannuksela, "Ghost Town Fly 3DV sequence for purposes of 3DV standardization", ISO/IEC JTC1/SC29/WG11 MPEG/M20027, Geneva, Switzerland, March 2011.

[Zit04] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, “High-quality video view interpolation using a layered representation”, ACM Trans. on Graphics, pp. 600-608, August 2004.

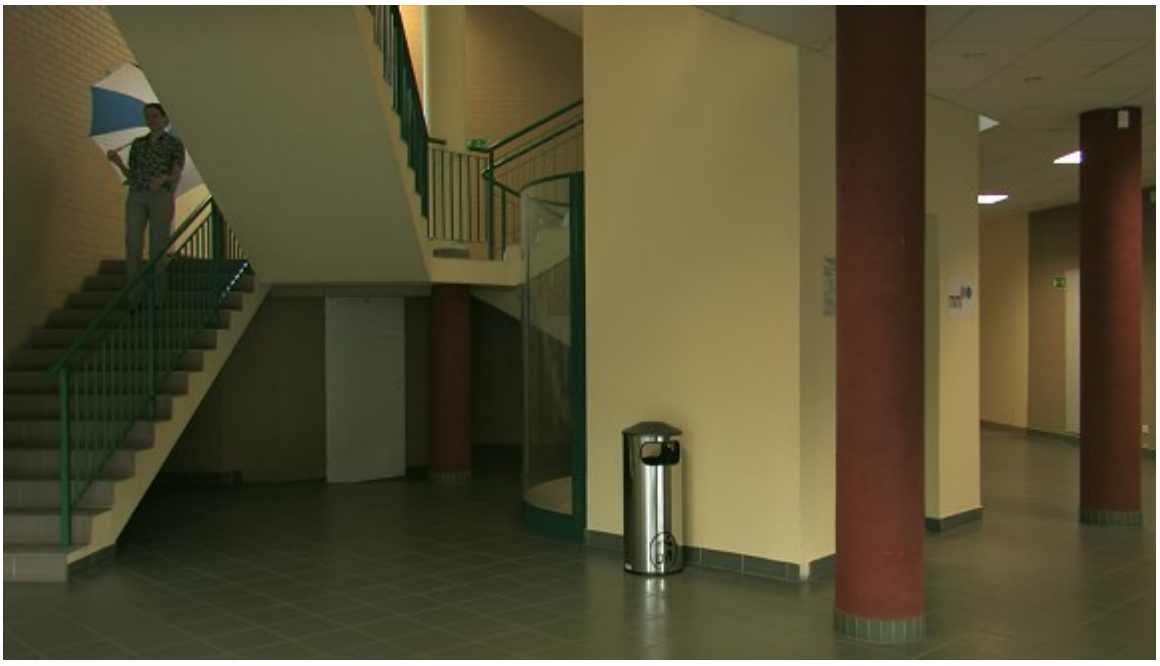
Anex A Video sequences used in experiments

A.1. Parameters of video sequences

#	Sequence	Sequence name	Resolution	Frame-rate [FPS]	Number of frames used	Coded views		Source
						2-view case	3-view case	
1.	Poznan Hall2	Poznan Hall2	1920×1088	25	200	6-7	5-7-6	[Dom09]
2.	Poznan Street	Poznan Street	1920×1088	25	250	3-4	3-5-4	[Dom09]
3.	Dancer	Undo Dancer	1920×1088	25	250	5-2	1-9-5	[Rusa11]
4.	GT Fly	Ghost Town Fly	1920×1088	25	250	2-5	1-9-5	[Zha11]
5.	Kendo	Kendo	1024×768	30	300	5-3	1-5-3	[Tani10]
6.	Balloons	Balloons	1024×768	30	300	5-3	1-5-3	[Tani10]
7.	Lovebird1	Lovebird1	1024×768	30	240	8-6	4-8-6	[Um08]
8.	Newspaper	Newspaper	1024×768	30	300	6-4	2-6-4	[Ho08]

A.2. Exemplary frames from video test sequences

Poznan Hall2



Poznan Street

Dancer

GT Fly

Kendo

Balloons

Loveird1

Newspaper

Anex B Codec configurations

B.1. Configuration of MVC codec

Parameter	Value
Group of Pictures (GOP) size	12, hierarchical B frames
Intra period	12
Entropy coder	CABAC
Number of reference frames	2
Maximum number of forward inter-view reference frames	1
Maximum number of backward inter-view reference frames	1
Anchor reference frames	enabled
Non-anchor reference frames	enabled
QP values	{22,27,32,37} unless other specified
View coding order	2 view case: 0-1
	3 view case: 0-2-1

B.2. Configuration of MV-HEVC codec

Parameter	Value
Group of Pictures (GOP) size	12, hierarchical B frames
Intra period	12
Entropy coder	CABAC
Number of reference frames	2
Maximum number of forward inter-view reference frames	1
Maximum number of backward inter-view reference frames	1
Anchor reference frames	enabled
Non-anchor reference frames	enabled
SAO	enabled
ALF	enabled
QP values	{22,27,32,37} unless other specified
View coding order	2 view case: 0-1
	3 view case: 0-2-1

Anex C Detailed experimental results

C.1. Bitstream structure for MVC

Tables show bitstream structure obtained for individual test sequences using MVC with configuration described in Annex B.1 for 2-view codec setup. Number of bits representing individual syntax elements is determined based on CAVLC (see Section 4.1).

Tab. C.1. Bitstream structure for MVC, values for individual test sequences.

Control data [%]								
Sequence	Base view				Side view			
	QP							
	22	27	32	37	22	27	32	37
Poznan Street	8.0	12.6	20.3	29.7	9.3	20.8	38.9	56.8
Poznan Hall2	16.5	28.5	39.5	48.2	19.7	36.8	50.4	60.2
Dancer	6.2	10.0	16.2	24.4	11.1	19.5	31.3	42.9
GT Fly	10.5	15.2	21.2	28.9	15.1	24.6	35.9	45.9
Kendo	12.4	17.8	24.4	31.6	13.7	20.7	29.4	39.0
Balloons	11.9	16.1	21.2	27.6	14.2	20.7	29.0	38.5
Lovebird1	5.9	10.1	16.0	24.2	6.6	12.7	24.0	41.4
Newspaper	8.1	11.9	17.7	24.9	11.0	18.9	30.7	44.1
Avg.	9.9	15.3	22.1	29.9	12.6	21.8	33.7	46.1

Transform coefficients [%]								
Sequence	Base view				Side view			
	QP							
	22	27	32	37	22	27	32	37
Poznan Street	83.6	78.8	70.6	61.9	77.2	55.3	31.8	16.2
Poznan Hall2	65.5	54.7	45.7	39.8	49.7	28.0	15.8	10.4
Dancer	85.8	78.9	69.9	60.4	72.3	53.9	33.2	19.0
GT Fly	68.9	60.7	52.4	44.8	58.2	38.9	21.8	13.0
Kendo	69.7	60.9	53.0	46.7	60.0	45.2	31.1	20.7
Balloons	68.5	62.5	57.4	53.3	56.2	42.1	29.6	21.0
Lovebird1	89.1	83.4	77.2	69.4	84.5	72.3	54.6	34.2
Newspaper	82.9	77.8	71.4	64.9	72.3	56.7	38.7	24.0
Avg.	76.8	69.7	62.2	55.1	66.3	49.0	32.1	19.8

Motion information [%]								
Sequence	Base view				Side view			
	QP							
	22	27	32	37	22	27	32	37
Poznan Street	8.3	8.7	9.1	8.4	13.5	23.9	29.3	27.0
Poznan Hall2	18.0	16.8	14.8	12.1	30.6	35.2	33.8	29.4
Dancer	8.0	11.0	13.9	15.2	16.6	26.6	35.5	38.1
GT Fly	20.6	24.1	26.3	26.3	26.7	36.5	42.3	41.1
Kendo	17.9	21.3	22.7	21.8	26.2	34.1	39.5	40.2
Balloons	19.6	21.5	21.4	19.1	29.6	37.2	41.3	40.4
Lovebird1	5.0	6.5	6.8	6.3	8.9	15.0	21.4	24.4
Newspaper	9.0	10.3	10.9	10.3	16.7	24.4	30.6	31.9
Avg.	13.3	15.0	15.7	14.9	21.1	29.1	34.2	34.1

C.2. Accuracy measures for inter-view motion information prediction

PCC and VSIM accuracy measures () for motion information prediction using median, IPDBP and FPDBP predictors, calculated for QP={22,27,32,37}. Tests performed for MVC reference codec with different motion information predictors implemented using configuration described in Annex B.1 and 2-view codec setup (refer to Section 4.3).

Tab. C.2. Accuracy measures for different motion information predictors, QP=22.

	Predictor	median					IPDBP					FPDBP				
Reference list 0	Sequence	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM
	Poznan Street	73	0.95	0.87	0.91	0.91	80	0.98	0.92	0.95	0.92	80	0.98	0.92	0.95	0.93
	Poznan Hall2	76	0.94	0.70	0.82	0.94	78	0.90	0.64	0.77	0.90	78	0.91	0.64	0.77	0.90
	GT Fly	23	0.97	0.95	0.96	0.90	62	0.98	0.98	0.98	0.96	62	0.98	0.98	0.98	0.96
	Dancer	61	0.88	0.66	0.77	0.89	71	0.98	0.75	0.86	0.95	71	0.98	0.75	0.87	0.95
	Kendo	71	0.88	0.69	0.78	0.89	80	0.93	0.85	0.89	0.92	80	0.94	0.86	0.90	0.93
	Balloons	69	0.86	0.89	0.87	0.87	83	0.94	0.95	0.94	0.92	83	0.94	0.95	0.95	0.92
	Lovebird1	76	0.79	0.75	0.77	0.97	93	0.85	0.81	0.83	0.98	93	0.86	0.80	0.83	0.98
	Newspaper	79	0.85	0.76	0.81	0.92	88	0.97	0.82	0.90	0.95	88	0.97	0.85	0.91	0.95
	Average	61	0.92	0.77	0.85	0.91	74	0.95	0.83	0.89	0.93	74	0.96	0.83	0.89	0.93
Reference list 1	Poznan Street	77	0.95	0.87	0.91	0.92	84	0.97	0.90	0.93	0.92	84	0.96	0.91	0.93	0.92
	Poznan Hall2	80	0.95	0.69	0.82	0.93	81	0.89	0.57	0.73	0.88	81	0.90	0.57	0.73	0.89
	GT Fly	28	0.95	0.93	0.94	0.92	76	0.98	0.98	0.98	0.95	76	0.98	0.98	0.98	0.95
	Dancer	57	0.95	0.61	0.78	0.93	78	0.97	0.73	0.85	0.94	77	0.97	0.72	0.84	0.93
	Kendo	77	0.87	0.64	0.75	0.87	86	0.91	0.77	0.84	0.89	86	0.92	0.78	0.85	0.89
	Balloons	76	0.90	0.88	0.89	0.87	86	0.93	0.93	0.93	0.91	87	0.94	0.93	0.93	0.91
	Lovebird1	83	0.83	0.72	0.77	0.96	95	0.81	0.76	0.78	0.97	95	0.80	0.75	0.78	0.97
	Newspaper	83	0.94	0.70	0.82	0.93	90	0.95	0.76	0.85	0.94	90	0.97	0.81	0.89	0.95
	Average	64	0.93	0.75	0.84	0.91	81	0.95	0.79	0.87	0.92	81	0.95	0.79	0.87	0.92

Tab. C.3. Accuracy measures for different motion information predictors, QP=27.

	Predictor	median					IPDBP					FPDBP				
Reference list 0	Sequence	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM
	Poznan Street	84	0.96	0.89	0.93	0.97	87	0.97	0.93	0.95	0.97	87	0.97	0.93	0.95	0.97
	Poznan Hall2	84	0.97	0.76	0.86	0.98	90	0.93	0.82	0.87	0.96	90	0.93	0.81	0.87	0.96
	GT Fly	38	0.97	0.96	0.97	0.95	76	0.97	0.96	0.96	0.96	76	0.97	0.96	0.97	0.96
	Dancer	68	0.92	0.78	0.85	0.93	81	0.98	0.80	0.89	0.95	81	0.98	0.80	0.89	0.95
	Kendo	75	0.92	0.75	0.83	0.93	85	0.95	0.87	0.91	0.95	86	0.95	0.89	0.92	0.95
	Balloons	72	0.90	0.91	0.91	0.92	89	0.96	0.96	0.96	0.95	89	0.96	0.96	0.96	0.95
	Lovebird1	84	0.82	0.79	0.80	0.98	91	0.86	0.83	0.84	0.98	91	0.87	0.83	0.85	0.98
	Newspaper	85	0.89	0.76	0.83	0.95	91	0.97	0.87	0.92	0.96	91	0.98	0.88	0.93	0.97
	Average	70	0.95	0.83	0.89	0.95	84	0.96	0.88	0.92	0.96	84	0.96	0.88	0.92	0.96
Reference list 1	Poznan Street	87	0.96	0.91	0.93	0.97	88	0.96	0.92	0.94	0.97	88	0.96	0.92	0.94	0.97
	Poznan Hall2	89	0.98	0.76	0.87	0.97	93	0.91	0.71	0.81	0.94	92	0.91	0.73	0.82	0.94
	GT Fly	44	0.96	0.96	0.96	0.95	85	0.97	0.96	0.97	0.95	85	0.97	0.96	0.97	0.95
	Dancer	68	0.96	0.79	0.87	0.95	86	0.97	0.78	0.88	0.94	86	0.97	0.78	0.88	0.94
	Kendo	82	0.91	0.71	0.81	0.91	91	0.94	0.83	0.88	0.93	91	0.94	0.83	0.88	0.93
	Balloons	81	0.92	0.91	0.92	0.91	92	0.95	0.95	0.95	0.93	92	0.95	0.95	0.95	0.94
	Lovebird1	88	0.85	0.77	0.81	0.97	92	0.84	0.80	0.82	0.97	92	0.84	0.80	0.82	0.97
	Newspaper	88	0.95	0.72	0.83	0.95	93	0.96	0.79	0.88	0.96	93	0.97	0.85	0.91	0.96
	Average	74	0.95	0.83	0.89	0.95	89	0.95	0.84	0.89	0.95	89	0.95	0.85	0.90	0.95

Tab. C.4. Accuracy measures for different motion information predictors, QP=32.

	Predictor	median					IPDBP					FPDBP				
Reference list 0	Sequence	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM
	Poznan Street	90	0.97	0.93	0.95	0.99	94	0.95	0.91	0.93	0.98	94	0.95	0.91	0.93	0.98
	Poznan Hall2	87	0.98	0.83	0.90	0.98	95	0.95	0.93	0.94	0.98	95	0.95	0.93	0.94	0.98
	GT Fly	51	0.98	0.97	0.97	0.96	86	0.97	0.96	0.96	0.97	86	0.97	0.96	0.96	0.98
	Dancer	72	0.94	0.85	0.90	0.96	88	0.98	0.87	0.92	0.96	88	0.98	0.87	0.92	0.96
	Kendo	80	0.95	0.82	0.88	0.95	89	0.95	0.90	0.93	0.96	90	0.95	0.91	0.93	0.97
	Balloons	76	0.92	0.93	0.92	0.95	92	0.96	0.96	0.96	0.96	92	0.96	0.96	0.96	0.96
	Lovebird1	86	0.84	0.83	0.84	0.99	93	0.87	0.85	0.86	0.98	93	0.88	0.86	0.87	0.99
	Newspaper	89	0.92	0.79	0.86	0.97	94	0.97	0.89	0.93	0.97	94	0.98	0.90	0.94	0.97
	Average	76	0.96	0.88	0.92	0.97	90	0.96	0.91	0.94	0.97	90	0.96	0.91	0.94	0.97
Reference list 1	Poznan Street	92	0.97	0.94	0.96	0.99	95	0.95	0.94	0.95	0.98	95	0.95	0.94	0.95	0.98
	Poznan Hall2	92	0.98	0.79	0.88	0.98	96	0.92	0.87	0.90	0.97	96	0.92	0.87	0.90	0.97
	GT Fly	56	0.97	0.96	0.96	0.96	92	0.97	0.96	0.96	0.97	92	0.97	0.96	0.96	0.97
	Dancer	77	0.96	0.86	0.91	0.96	92	0.97	0.86	0.91	0.95	92	0.97	0.86	0.91	0.95
	Kendo	85	0.94	0.78	0.86	0.94	93	0.95	0.86	0.90	0.95	93	0.95	0.87	0.91	0.95
	Balloons	84	0.94	0.93	0.94	0.94	95	0.96	0.95	0.96	0.95	95	0.96	0.95	0.96	0.95
	Lovebird1	92	0.85	0.81	0.83	0.98	94	0.85	0.84	0.84	0.98	94	0.85	0.83	0.84	0.98
	Newspaper	91	0.95	0.75	0.85	0.97	95	0.96	0.84	0.90	0.97	95	0.97	0.89	0.93	0.97
	Average	80	0.96	0.87	0.92	0.97	94	0.95	0.90	0.92	0.96	94	0.95	0.90	0.93	0.96

Tab. C.5. Accuracy measures for different motion information predictors, QP=37.

	Predictor	median					IPDBP					FPDBP				
Reference list 0	Sequence	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM	% of points	PCCx	PCCy	PCCavg	VSIM
	Poznan Street	92	0.97	0.93	0.95	0.99	97	0.96	0.95	0.95	0.99	97	0.96	0.95	0.95	0.99
	Poznan Hall2	89	0.97	0.90	0.94	0.99	97	0.97	0.97	0.97	0.99	97	0.97	0.97	0.97	0.99
	GT Fly	59	0.98	0.97	0.97	0.97	92	0.98	0.98	0.98	0.98	92	0.98	0.98	0.98	0.98
	Dancer	77	0.94	0.90	0.92	0.97	93	0.98	0.92	0.95	0.97	92	0.98	0.92	0.95	0.97
	Kendo	83	0.96	0.81	0.88	0.97	92	0.95	0.90	0.93	0.98	93	0.96	0.91	0.93	0.98
	Balloons	82	0.94	0.94	0.94	0.97	95	0.96	0.96	0.96	0.97	95	0.96	0.97	0.96	0.97
	Lovebird1	93	0.87	0.85	0.86	0.99	96	0.89	0.88	0.88	0.99	96	0.89	0.89	0.89	0.99
	Newspaper	91	0.96	0.84	0.90	0.98	96	0.97	0.92	0.95	0.98	96	0.97	0.92	0.95	0.98
	Average	80	0.96	0.90	0.93	0.98	94	0.97	0.94	0.96	0.98	94	0.97	0.94	0.96	0.98
Reference list 1	Poznan Street	94	0.98	0.95	0.96	0.99	98	0.95	0.94	0.95	0.99	98	0.95	0.94	0.94	0.99
	Poznan Hall2	94	0.97	0.87	0.92	0.98	98	0.96	0.94	0.95	0.98	98	0.96	0.94	0.95	0.98
	GT Fly	64	0.97	0.96	0.97	0.97	95	0.98	0.97	0.98	0.98	95	0.98	0.97	0.98	0.98
	Dancer	83	0.96	0.91	0.93	0.97	95	0.97	0.92	0.94	0.96	95	0.97	0.92	0.94	0.96
	Kendo	88	0.95	0.83	0.89	0.96	95	0.95	0.89	0.92	0.97	95	0.95	0.89	0.92	0.97
	Balloons	88	0.95	0.93	0.94	0.96	96	0.96	0.96	0.96	0.96	96	0.96	0.96	0.96	0.96
	Lovebird1	95	0.89	0.88	0.88	0.99	96	0.85	0.82	0.84	0.98	96	0.86	0.82	0.84	0.98
	Newspaper	93	0.97	0.82	0.90	0.98	97	0.96	0.84	0.90	0.97	97	0.97	0.91	0.94	0.98
	Average	85	0.97	0.90	0.93	0.97	96	0.96	0.93	0.95	0.98	96	0.96	0.93	0.95	0.98

C.3. Performance of different occlusion detection algorithms

Table presents percentages of unassigned pixels determined by different occlusion detection algorithms for individual test sequences and QD={0,20,30,40,50}, using 2-view setup (see Section 6.1).

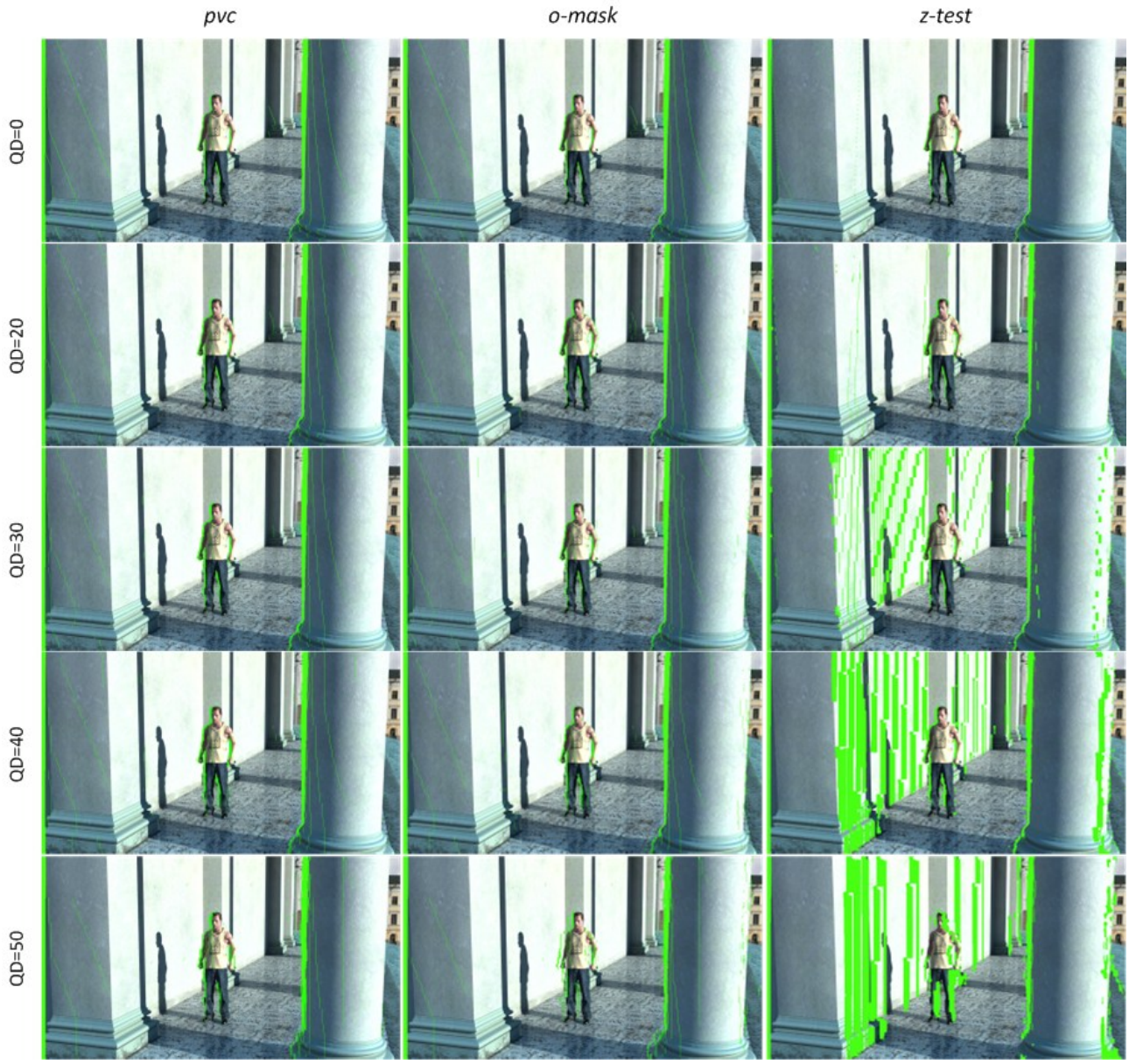
Tab. C.6. Unassigned pixels for different occlusion detection algorithms and QD values.

Unassigned pixels [% of picture area]							
Sequence	Algorithm	QD					
		0	20	30	40	50	Avg.
Poznan Street	pvc	4.350	3.722	3.010	2.617	2.220	3.186
	z-test	29.746	27.096	20.839	11.617	9.151	19.690
	o-mask	3.875	3.446	3.034	2.649	2.298	3.060
Poznan Hall2	pvc	3.445	3.583	3.710	3.573	3.377	3.538
	z-test	15.703	15.671	14.466	13.988	15.263	15.018
	o-mask	3.437	3.565	3.794	3.928	3.659	3.677
Dancer	pvc	3.746	3.836	3.865	3.887	3.914	3.850
	z-test	3.375	3.819	7.038	22.855	21.384	11.694
	o-mask	3.746	3.848	3.893	3.953	4.091	3.906
GT Fly	pvc	2.144	2.237	2.156	2.021	1.824	2.076
	z-test	2.128	5.105	10.633	15.017	11.943	8.965
	o-mask	2.144	2.243	2.150	2.020	1.843	2.080
Kendo	pvc	5.859	5.953	6.018	6.225	6.043	6.020
	z-test	44.713	45.518	46.561	46.781	40.961	44.907
	o-mask	4.193	4.247	4.324	4.758	5.236	4.552
Balloons	pvc	6.009	6.094	6.076	5.995	5.588	5.953
	z-test	39.724	41.000	42.492	39.946	30.637	38.760
	o-mask	4.956	5.004	5.027	5.426	5.883	5.259
Lovebird1	pvc	3.496	3.338	3.084	2.769	2.431	3.024
	z-test	7.954	7.492	6.943	6.286	3.512	6.437
	o-mask	3.689	3.467	3.186	2.897	2.669	3.182
Newspaper	pvc	11.022	11.134	10.940	10.472	9.909	10.696
	z-test	28.467	29.378	29.303	28.336	27.668	28.630
	o-mask	12.382	12.581	12.255	11.770	11.306	12.059

C.4. Exemplary pictures with unassigned pixels for different occlusion detection algorithms

Pictures present unassigned pixels determined by different occlusion detection algorithms for QD={0,20,30,40,50} and selected test sequences (see Section 6.1). Unassigned pixels are marked in green.

Dancer **(synthetic video)**



***Kendo* (natural video)**

C.5. Bitrate change for different unassigned area filling algorithms

Tables show bitrate change measured for MVC codec using different combinations of prediction, occlusion detection and unassigned area filling algorithms, against original MVC codec. Changes in bitrate are calculated using Bjontegaard metrics for QP={22,27,32,37}. Results are obtained using MVC configuration described in Annex B.1 for 2-view codec setup and depth maps compressed with QD={0,20,40} (refer to Section 6.2).

Tab. C.7. Bitrate change (ΔBitrate [%]) for different unassigned area filling algorithms, *Poznan Street* test sequence (side view).

Prediction & occlusion detection algorithm	Filling algorithm	QD			
		0	20	40	Avg.
IPDBP	FILLno	-2.74	-2.99	-3.45	-3.06
	FILLmax	-6.05	-6.05	-6.01	-6.04
	FILLmin	-6.04	-6.03	-5.97	-6.01
FPDBP *z-test*	FILLno	-1.17	-1.63	-2.22	-1.67
	FILLmax	-4.52	-4.53	-4.57	-4.54
	FILLmin	-4.75	-4.80	-5.06	-4.87
	FILLsim	-4.82	-4.80	-4.84	-4.82
FPDBP *o-mask*	FILLno	-3.20	-3.26	-3.41	-3.29
	FILLmax	-6.22	-6.23	-6.02	-6.15
	FILLmin	-6.23	-6.24	-6.01	-6.16
	FILLsim	-6.23	-6.24	-6.01	-6.16

Tab. C.8. Bitrate change (ΔBitrate [%]) for different unassigned area filling algorithms, *GT Fly* test sequence (side view).

Prediction & occlusion detection algorithm	Filling algorithm	QD			
		0	20	40	Avg.
IPDBP	FILLno	-15.85	-15.15	-16.02	-15.67
	FILLmax	-23.18	-23.01	-22.51	-22.90
	FILLmin	-23.03	-22.87	-22.44	-22.78
FPDBP *z-test*	FILLno	-15.83	-12.09	-9.70	-12.54
	FILLmax	-22.84	-21.75	-17.81	-20.80
	FILLmin	-22.66	-21.96	-18.64	-21.09
	FILLsim	-22.84	-22.11	-18.32	-21.09
FPDBP *o-mask*	FILLno	-15.82	-14.97	-16.06	-15.62
	FILLmax	-22.79	-22.70	-22.20	-22.57
	FILLmin	-22.64	-22.58	-22.17	-22.46
	FILLsim	-22.66	-22.60	-22.19	-22.48

Tab. C.9. Bitrate change (ΔBitrate [%]) for different unassigned area filling algorithms, *Balloons* test sequence (side view).

Prediction & occlusion detection algorithm	Filling algorithm	QD			
		0	20	40	Avg.
IPDBP	FILLno	-13.63	-13.79	-15.36	-14.26
	FILLmax	-23.68	-23.65	-23.53	-23.62
	FILLmin	-23.65	-23.63	-23.49	-23.59
FPDBP *z-test*	FILLno	-5.21	-4.57	-5.01	-4.93
	FILLmax	-12.12	-11.66	-12.00	-11.93
	FILLmin	-13.31	-12.76	-13.29	-13.12
	FILLsim	-13.54	-13.06	-13.11	-13.24
FPDBP *o-mask*	FILLno	-16.58	-16.50	-15.37	-16.15
	FILLmax	-22.82	-22.80	-22.55	-22.72
	FILLmin	-22.78	-22.79	-22.51	-22.70
	FILLsim	-22.79	-22.79	-22.51	-22.70

Tab. C.10. Bitrate change (ΔBitrate [%]) for different unassigned area filling algorithms, *Newspaper* test sequence (side view).

Prediction & occlusion detection algorithm	Filling algorithm	QD			
		0	20	40	Avg.
IPDBP	FILLno	-6.01	-5.96	-6.26	-6.08
	FILLmax	-11.02	-10.96	-10.85	-10.94
	FILLmin	-10.96	-10.91	-10.75	-10.87
FPDBP *z-test*	FILLno	-5.46	-5.26	-6.20	-5.64
	FILLmax	-9.51	-9.29	-9.24	-9.35
	FILLmin	-9.86	-9.67	-9.81	-9.78
	FILLsim	-9.76	-9.57	-9.64	-9.66
FPDBP *o-mask*	FILLno	-4.86	-4.67	-5.33	-4.95
	FILLmax	-11.28	-11.29	-11.20	-11.26
	FILLmin	-11.34	-11.34	-11.24	-11.31
	FILLsim	-11.34	-11.34	-11.24	-11.31

C.6. Correlation of coordinates of corresponding pixels assigned by mapping with original and compressed depth information

Table shows correlation of horizontal coordinates of the corresponding pixels calculated using original (QD=0) and compressed depth information (QD={20,30,40,50}), for different unassigned area filling algorithms (refer to Section 6.2). The values are averaged over all test sequences (see Annex A.1).

Tab. C.11. PCC coefficients for horizontal coordinates of assigned corresponding pixels calculated for QD={20,30,40,50} vs. QD=0.

Prediction & occlusion detection algorithm	Filling algorithm	QD				
		20	30	40	50	Avg.
IPDBP	FILLno	0.99999888	0.99999800	0.99999650	0.99999200	0.99999708
	FILLmax	0.99999900	0.99999775	0.99999650	0.99999225	0.99999710
	FILLmin	0.99999900	0.99999788	0.99999650	0.99999225	0.99999713
FPDBP *z-test*	FILLno	0.99999988	0.99999963	0.99999813	0.99999138	0.99999780
	FILLmax	0.99864063	0.99700913	0.99261038	0.98924038	0.99550010
	FILLmin	0.99870463	0.99724388	0.99283288	0.98958513	0.99567330
	FILLsim	0.99866125	0.99712288	0.99295463	0.98965650	0.99567905
FPDBP *o-mask*	FILLno	1.00000000	0.99999963	0.99999800	0.99999025	0.99999758
	FILLmax	0.99999900	0.99999788	0.99999600	0.99998863	0.99999630
	FILLmin	0.99999925	0.99999850	0.99999675	0.99998938	0.99999678
	FILLsim	0.99999925	0.99999850	0.99999675	0.99998925	0.99999675

C.7. Usage of low-cost modes for different codecs

Tables present usage of low-cost modes in JMVC, JMVC+FPDBP, JMVC+IPDBP and JMVM+MS codecs, averaged over complete set of test sequences (see Annex A.1). The results are obtained using MVC configuration described in Annex B.1 for 2-view and 3-view codec setup (refer to Section 6.3.2).

Tab. C.12. Average usage [%] of low-cost modes for different codecs (2-view case, side view).

Algorithm	MB mode	QP				
		22	27	32	37	Avg.
JMVC	Skip	56.4	68.2	75.5	79.8	70.0
	Direct	3.8	1.4	0.7	0.4	1.6
	IVS	0.0	0.0	0.0	0.0	0.0
	IVD	0.0	0.0	0.0	0.0	0.0
	Sum	60.2	69.6	76.2	80.3	71.6
JMVC +FPDBP	Skip	12.3	10.6	7.3	4.8	8.7
	Direct	2.1	0.7	0.4	0.2	0.9
	IVS	55.6	68.4	76.9	82.2	70.8
	IVD	3.4	1.5	0.7	0.4	1.5
	Sum	73.5	81.2	85.3	87.6	81.9
JMVC +IPDBP	Skip	12.2	10.5	7.2	4.7	8.6
	Direct	2.1	0.7	0.4	0.2	0.9
	IVS	55.9	68.6	77.1	82.3	71.0
	IVD	3.5	1.5	0.7	0.4	1.5
	Sum	73.7	81.3	85.3	87.6	82.0
JMVM +MS	Skip	52.7	65.8	73.4	78.2	67.5
	Direct	2.3	0.8	0.4	0.2	0.9
	IVS	0.0	0.0	0.0	0.0	0.0
	IVD	0.0	0.0	0.0	0.0	0.0
	Sum	55.0	66.6	73.8	78.5	68.5

Tab. C.13. Average usage [%] of low-cost modes for different codecs (3-view case).

Algorithm	MB mode	View 1 (central)					View 2 (outer)				
		QP									
		22	27	32	37	Avg.	22	27	32	37	Avg.
JMVC	Skip	58.9	72.8	81.2	86.6	74.8	54.7	67.3	74.6	79.2	68.9
	Direct	5.8	2.7	1.5	1.1	2.8	4.4	1.5	0.8	0.5	1.8
	IVS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	IVD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Sum	64.6	75.4	82.7	87.7	77.6	59.1	68.8	75.4	79.7	70.7
JMVC +FPDBP	Skip	13.9	11.7	9.7	8.0	10.8	14.5	13.0	9.5	6.6	10.9
	Direct	4.0	2.0	1.1	0.8	2.0	2.7	0.9	0.5	0.3	1.1
	IVS	55.8	71.0	79.8	85.6	73.1	49.6	63.1	72.4	78.8	66.0
	IVD	2.7	1.1	0.5	0.3	1.1	2.7	1.1	0.5	0.3	1.2
	Sum	76.5	85.8	91.2	94.6	87.0	69.5	78.1	82.9	86.0	79.1
JMVC +IPDBP	Skip	13.5	11.4	9.4	7.6	10.5	13.6	12.1	8.6	5.8	10.0
	Direct	4.0	1.9	1.1	0.8	2.0	2.6	0.9	0.4	0.3	1.0
	IVS	56.4	71.6	80.4	86.1	73.6	51.4	64.9	74.1	80.1	67.6
	IVD	2.8	1.1	0.5	0.3	1.2	3.3	1.4	0.7	0.4	1.4
	Sum	76.8	86.0	91.4	94.8	87.2	70.9	79.3	83.8	86.6	80.1
JMVM +MS	Skip	55.6	70.3	79.2	85.0	72.5	52.1	65.8	73.5	78.4	67.5
	Direct	4.0	2.1	1.3	0.9	2.1	3.0	1.0	0.5	0.3	1.2
	IVS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	IVD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Sum	59.6	72.4	80.4	85.9	74.6	55.2	66.8	74.1	78.7	68.7

C.8. Bitstream structure for different codecs

Tables present bitstream structure for JMVC, JMVC+FPDBP, JMVC+IPDBP and JMVM+MS codecs, averaged over complete set of test sequences (see Annex A.1). The results are obtained using MVC configuration described in Annex B.1 for 2-view and 3-view codec setup (refer to Section 6.3.2).

Tab. C.14. Bitstream structure [%] for different codecs (2-view case, side view).

Algorithm	Syntax element	QP				
		22	27	32	37	Avg.
JMVC	Control data	13	22	34	46	29
	Transform coefficients	66	49	32	20	42
	Motion information	21	29	34	34	30
JMVC +FPDBP	Control data	12	22	37	53	31
	Transform coefficients	71	56	38	24	47
	Motion information	17	22	25	23	22
JMVC +IPDBP	Control data	12	22	37	53	31
	Transform coefficients	71	56	38	24	47
	Motion information	17	22	25	23	22
JMVM +MS	Control data	10	19	33	48	27
	Transform coefficients	76	62	46	32	54
	Motion information	14	19	22	20	19

Tab. C.15. Bitstream structure [%] for different codecs (3-view case).

Algorithm	Syntax element	View 1 (central)					View 2 (outer)				
		QP									
		22	27	32	37	Avg.	22	27	32	37	Avg.
JMVC	Control data	14	25	37	49	31	12	19	29	40	25
	Transform coefficients	62	44	27	17	38	69	55	41	29	48
	Motion information	23	31	35	33	31	19	26	31	32	27
JMVC +FPDBP	Control data	14	26	41	57	35	11	19	30	44	26
	Transform coefficients	66	49	32	21	42	72	59	45	33	52
	Motion information	19	25	27	22	23	16	21	24	24	22
JMVC +IPDBP	Control data	14	26	42	57	35	11	19	30	44	26
	Transform coefficients	67	49	32	21	42	73	60	46	33	53
	Motion information	19	25	26	22	23	16	20	23	23	21
JMVM +MS	Control data	11	22	37	53	31	9	17	28	41	24
	Transform coefficients	74	58	41	29	51	77	65	51	38	58
	Motion information	15	20	21	18	19	14	18	21	21	18

C.9. Coding efficiency measures for JMVC+FPDBP and JMVC+IPDBP calculated for different depth quality

Tables show Bjontegaard metrics for JMVC+FPDBP and JMVC+IPDBP vs. JMVC codec, calculated for QP={22,27,32,37} and different QD values. The results present coding efficiency measures for side views only and are obtained using MVC configuration described in Annex B.1 for 2-view and 3-view codec setup (refer to Section 6.3.3).

Tab. C.16. Bjontegaard metrics for JMVC+FPDBP and JMVC+IPDBP vs. JMVC codec, calculated for side view with QP={22,27,32,37} (2-view case, *view 1*).

ΔPSNRY [dB]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	0.12	0.12	0.12	0.12	0.11	0.12	0.12	0.12	0.12	0.12
Poznan Hall2	0.34	0.33	0.33	0.33	0.33	0.34	0.34	0.34	0.33	0.33
Dancer	0.39	0.39	0.38	0.37	0.36	0.41	0.40	0.39	0.38	0.38
GT Fly	0.67	0.67	0.66	0.65	0.63	0.68	0.67	0.67	0.66	0.65
Kendo	0.55	0.55	0.55	0.54	0.52	0.55	0.55	0.54	0.54	0.53
Balloons	0.89	0.89	0.89	0.88	0.86	0.92	0.92	0.92	0.92	0.90
Lovebird1	0.10	0.10	0.10	0.10	0.09	0.10	0.10	0.10	0.09	0.09
Newspaper	0.40	0.40	0.40	0.40	0.38	0.39	0.39	0.39	0.39	0.38
Avg.	0.43	0.43	0.43	0.42	0.41	0.44	0.44	0.43	0.43	0.42
ΔBitrate [%]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	-6.2	-6.2	-6.2	-6.0	-5.7	-6.1	-6.0	-6.0	-6.0	-5.8
Poznan Hall2	-22.1	-22.0	-21.9	-21.8	-21.4	-22.4	-22.3	-22.1	-21.8	-21.5
Dancer	-12.7	-12.6	-12.3	-11.9	-11.6	-13.3	-12.9	-12.6	-12.4	-12.2
GT Fly	-22.8	-22.7	-22.5	-22.2	-21.6	-23.2	-23.0	-22.8	-22.5	-22.1
Kendo	-15.4	-15.4	-15.3	-15.1	-14.5	-15.2	-15.1	-15.1	-15.1	-14.7
Balloons	-22.8	-22.8	-22.8	-22.5	-21.9	-23.7	-23.7	-23.6	-23.5	-23.1
Lovebird1	-2.8	-2.8	-2.8	-2.7	-2.6	-2.7	-2.7	-2.7	-2.6	-2.5
Newspaper	-11.3	-11.3	-11.2	-11.2	-10.7	-11.0	-11.0	-10.9	-10.8	-10.5
Avg.	-14.5	-14.5	-14.4	-14.2	-13.8	-14.7	-14.6	-14.5	-14.3	-14.1

Tab. C.17. Bjontegaard metrics for JMVC+FPDBP and JMVC+IPDBP vs. JMVC codec, calculated for side view with QP={22,27,32,37} (3-view case, *view 1*).

ΔPSNRY [dB]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	0.18	0.18	0.18	0.18	0.17	0.17	0.18	0.18	0.18	0.17
Poznan Hall2	0.40	0.40	0.40	0.39	0.38	0.40	0.40	0.40	0.40	0.39
Dancer	0.43	0.40	0.39	0.37	0.35	0.44	0.42	0.41	0.39	0.38
GT Fly	0.55	0.55	0.54	0.53	0.52	0.56	0.54	0.53	0.53	0.52
Kendo	0.55	0.54	0.53	0.52	0.51	0.60	0.60	0.59	0.59	0.58
Balloons	0.93	0.91	0.91	0.89	0.89	1.05	1.05	1.05	1.04	1.03
Lovebird1	0.11	0.11	0.11	0.11	0.10	0.14	0.14	0.14	0.13	0.12
Newspaper	0.42	0.41	0.41	0.41	0.39	0.44	0.43	0.43	0.42	0.41
Avg.	0.45	0.44	0.43	0.42	0.41	0.48	0.47	0.47	0.46	0.45
ΔBitrate [%]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	-8.2	-8.2	-8.2	-8.1	-7.9	-8.1	-8.1	-8.1	-8.1	-7.9
Poznan Hall2	-22.2	-22.2	-21.9	-21.6	-21.0	-22.3	-22.1	-22.0	-21.7	-21.1
Dancer	-13.1	-12.2	-11.9	-11.3	-10.8	-13.8	-13.1	-12.6	-12.1	-11.7
GT Fly	-18.4	-18.2	-17.9	-17.6	-17.2	-18.5	-18.1	-17.8	-17.6	-17.2
Kendo	-15.5	-15.2	-15.2	-14.8	-14.3	-17.0	-16.9	-16.8	-16.7	-16.3
Balloons	-23.2	-22.8	-22.9	-22.5	-22.3	-26.7	-26.6	-26.7	-26.5	-26.1
Lovebird1	-3.0	-3.0	-2.9	-2.9	-2.7	-3.6	-3.6	-3.6	-3.4	-3.1
Newspaper	-11.1	-10.9	-10.9	-10.8	-10.5	-11.6	-11.5	-11.4	-11.2	-10.8
Avg.	-14.3	-14.1	-14.0	-13.7	-13.3	-15.2	-15.0	-14.9	-14.7	-14.3

Tab. C.18. Bjontegaard metrics for JMVC+FPDBP and JMVC+IPDBP vs. JMVC codec, calculated for side view with QP={22,27,32,37} (3-view case, *view 2*).

ΔPSNRY [dB]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	0.09	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Poznan Hall2	0.32	0.32	0.31	0.31	0.30	0.32	0.32	0.32	0.31	0.30
Dancer	0.19	0.16	0.16	0.15	0.14	0.25	0.23	0.23	0.22	0.21
GT Fly	0.51	0.51	0.50	0.49	0.47	0.52	0.51	0.50	0.50	0.48
Kendo	0.21	0.18	0.18	0.18	0.18	0.37	0.37	0.36	0.36	0.35
Balloons	0.40	0.36	0.36	0.35	0.36	0.68	0.68	0.67	0.66	0.64
Lovebird1	0.02	0.02	0.02	0.01	0.01	0.05	0.05	0.05	0.05	0.05
Newspaper	0.09	0.07	0.07	0.08	0.07	0.18	0.18	0.18	0.19	0.19
Avg.	0.23	0.21	0.21	0.21	0.20	0.31	0.30	0.30	0.30	0.29
ΔBitrate [%]										
Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QD									
	0	20	30	40	50	0	20	30	40	50
Poznan Street	-4.0	-4.0	-3.9	-3.8	-3.4	-3.8	-3.8	-3.8	-3.7	-3.6
Poznan Hall2	-17.2	-17.2	-17.0	-16.8	-15.8	-17.4	-17.4	-17.2	-16.9	-16.3
Dancer	-5.9	-5.1	-5.0	-4.7	-4.2	-7.7	-7.2	-7.1	-6.9	-6.7
GT Fly	-16.8	-16.6	-16.4	-16.1	-15.4	-17.1	-16.7	-16.5	-16.3	-15.8
Kendo	-5.4	-4.9	-4.8	-4.7	-4.8	-9.8	-9.7	-9.6	-9.6	-9.2
Balloons	-9.7	-8.8	-8.8	-8.6	-8.8	-16.4	-16.3	-16.3	-16.0	-15.3
Lovebird1	-0.5	-0.5	-0.5	-0.4	-0.2	-1.6	-1.6	-1.6	-1.5	-1.4
Newspaper	-2.2	-1.7	-1.8	-2.0	-1.8	-4.4	-4.3	-4.4	-4.5	-4.5
Avg.	-7.7	-7.3	-7.3	-7.1	-6.8	-9.8	-9.6	-9.5	-9.4	-9.1

C.10. Usage of low-cost modes for JMVC+FPDBP and JMVC+IPDBP measured for different depth quality

Tables present usage of low-cost modes in JMVC+FPDBP and JMVC+IPDBP codecs for different QD values. The values are averaged over QP={22,27,32,37} and complete set of test sequences (see Annex A.1). The results are obtained using MVC configuration described in Annex B.1 for 2-view and 3-view codec setup (refer to Section 6.3.3).

Tab. C.19. Average usage [%] of low-cost modes for JMVC+FPDBP and JMVC+IPDBP codecs (2-view case, side view).

Algorithm	MB mode	QD				
		0	20	30	40	50
JMVC +FPDBP	Skip	8.75	8.75	8.76	8.78	8.84
	Direct	0.86	0.86	0.86	0.86	0.87
	IVS	70.80	70.79	70.76	70.71	70.58
	IVD	1.51	1.50	1.49	1.48	1.45
	Sum	81.91	81.90	81.88	81.84	81.73
JMVC +IPDBP	Skip	8.64	8.65	8.66	8.68	8.71
	Direct	0.86	0.86	0.86	0.86	0.86
	IVS	70.98	70.96	70.93	70.90	70.81
	IVD	1.51	1.49	1.49	1.49	1.47
	Sum	81.99	81.96	81.95	81.92	81.85

Tab. C.20. Average usage [%] of low-cost modes for JMVC+FPDBP and JMVC+IPDBP codecs (3-view case).

Algorithm	MB mode	View 1 (central)					View 2 (outer)				
		QD									
		0	20	30	40	50	0	20	30	40	50
JMVC +FPDBP	Skip	10.84	10.92	10.94	11.05	11.13	10.89	11.04	11.05	11.12	11.23
	Direct	1.98	1.98	1.98	1.99	1.99	1.12	1.13	1.13	1.13	1.14
	IVS	73.05	72.94	72.89	72.71	72.55	65.97	65.69	65.67	65.54	65.33
	IVD	1.14	1.11	1.10	1.08	1.07	1.16	1.08	1.07	1.06	1.05
	Sum	87.01	86.95	86.92	86.83	86.74	79.13	78.94	78.92	78.85	78.75
JMVC +IPDBP	Skip	10.48	10.50	10.52	10.56	10.66	10.03	10.06	10.06	10.07	10.14
	Direct	1.96	1.96	1.96	1.97	1.98	1.05	1.05	1.05	1.05	1.06
	IVS	73.61	73.56	73.52	73.43	73.24	67.62	67.56	67.53	67.50	67.34
	IVD	1.18	1.16	1.15	1.14	1.12	1.45	1.42	1.41	1.41	1.39
	Sum	87.24	87.18	87.15	87.09	87.00	80.14	80.08	80.06	80.03	79.92

C.11. Usage of merge candidate predictors in MV-HEVC+IPDBP and MV-HEVC codecs

Tables show usage of merge candidate predictors in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec: MV-HEVC+IPDBP1, MV-HEVC+IPDBP2 and MV-HEVC+IPDBP3, for QP={22,27,32,37}. The values are averaged over all test sequences (see Annex A.1). The results are obtained using MV-HEVC configuration described in Annex B.2 for 2-view and 3-view codec setup (refer to Section 6.4).

Tab. C.21. Usage of merge candidate predictors [% of picture area] in MV-HEVC+IPDBP and MV-HEVC codecs (2-view case, side view).

Codec variant	Merge candidate	QP				
		22	27	32	37	Avg.
MV-HEVC	left	46.6	54.0	58.1	61.5	55.0
	top	16.8	18.4	19.4	19.9	18.6
	co-located	7.3	6.3	6.1	6.1	6.4
	RT corner	4.3	3.5	3.0	2.6	3.4
	BL corner	0.9	0.5	0.3	0.2	0.5
	DBP	0.0	0.0	0.0	0.0	0.0
	Sum	75.8	82.7	87.0	90.3	84.0
MV-HEVC +IPDBP1	left	28.7	23.7	22.7	23.6	24.7
	top	10.7	8.5	6.8	5.8	7.9
	co-located	5.1	3.4	2.3	1.8	3.2
	RT corner	2.6	1.6	1.2	1.1	1.6
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	33.3	50.4	58.3	61.4	50.8
	Sum	81.0	88.0	91.6	93.8	88.6
MV-HEVC +IPDBP2	left	28.0	22.2	18.9	11.5	20.2
	top	10.0	8.1	6.4	5.8	7.6
	co-located	5.2	3.5	2.4	1.9	3.3
	RT corner	2.7	1.6	1.3	1.1	1.7
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	34.3	52.2	62.3	73.4	55.5
	Sum	80.9	87.9	91.6	93.8	88.5
MV-HEVC +IPDBP3	left	31.5	29.3	27.9	28.8	29.4
	top	11.8	10.4	8.8	7.8	9.7
	co-located	5.2	3.6	2.8	2.3	3.5
	RT corner	2.6	1.6	1.3	1.1	1.7
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	29.3	42.8	50.7	53.7	44.1
	Sum	81.0	88.0	91.6	93.8	88.6

Tab. C.22. Usage of merge candidate predictors [% of picture area] in MV-HEVC+IPDBP and MV-HEVC codecs (3-view case, *view 1*).

Codec variant	Merge candidate	QP				
		22	27	32	37	Avg.
MV-HEVC	left	47.0	54.8	58.5	61.6	55.5
	top	17.2	19.0	19.9	20.5	19.2
	co-located	8.0	7.1	7.0	7.0	7.3
	RT corner	4.5	3.5	3.0	2.5	3.4
	BL corner	0.9	0.4	0.3	0.1	0.4
	DBP	0.0	0.0	0.0	0.0	0.0
	Sum	77.7	84.9	88.7	91.8	85.8
MV-HEVC +IPDBP1	left	29.4	24.7	23.4	22.8	25.1
	top	11.5	9.3	7.6	6.3	8.7
	co-located	5.8	4.3	3.3	2.7	4.0
	RT corner	2.9	1.8	1.3	1.1	1.8
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	32.0	49.0	56.8	61.6	49.8
	Sum	82.2	89.5	92.6	94.7	89.7
MV-HEVC +IPDBP2	left	28.2	22.3	18.9	10.0	19.8
	top	10.8	8.9	7.4	6.5	8.4
	co-located	6.0	4.4	3.4	2.8	4.1
	RT corner	3.0	1.8	1.3	1.1	1.8
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	33.5	51.7	61.4	74.2	55.2
	Sum	82.1	89.4	92.6	94.7	89.7
MV-HEVC +IPDBP3	left	32.4	30.2	28.6	27.6	29.7
	top	12.6	11.2	9.8	8.3	10.5
	co-located	5.9	4.4	3.7	3.2	4.3
	RT corner	3.0	1.8	1.4	1.1	1.8
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	27.8	41.5	49.1	54.4	43.2
	Sum	82.3	89.4	92.6	94.6	89.7

Tab. C.23. Usage of merge candidate predictors [% of picture area] in MV-HEVC+IPDBP and MV-HEVC codecs (3-view case, *view 2*).

Codec variant	Merge candidate	QP				
		22	27	32	37	Avg.
MV-HEVC	left	45.3	52.5	56.5	59.7	53.5
	top	16.8	18.0	18.9	19.6	18.3
	co-located	7.1	6.2	6.2	6.3	6.4
	RT corner	4.2	3.3	2.9	2.5	3.3
	BL corner	0.8	0.5	0.3	0.2	0.4
	DBP	0.0	0.0	0.0	0.0	0.0
	Sum	74.2	80.4	84.8	88.3	81.9
MV-HEVC +IPDBP1	left	31.8	28.6	27.8	27.7	29.0
	top	12.2	10.7	9.2	7.9	10.0
	co-located	5.4	4.0	3.2	2.7	3.8
	RT corner	2.7	1.7	1.4	1.2	1.7
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	25.7	39.2	46.7	51.5	40.8
	Sum	78.3	84.5	88.4	91.2	85.6
MV-HEVC +IPDBP2	left	31.1	26.6	22.6	13.9	23.6
	top	11.0	9.8	8.5	7.8	9.3
	co-located	5.5	4.1	3.3	2.8	3.9
	RT corner	2.8	1.8	1.4	1.2	1.8
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	27.2	41.8	52.3	65.2	46.6
	Sum	78.2	84.4	88.3	91.1	85.5
MV-HEVC +IPDBP3	left	34.5	34.4	33.5	33.0	33.9
	top	13.1	12.2	11.1	10.0	11.6
	co-located	5.2	4.0	3.3	2.9	3.9
	RT corner	2.8	1.8	1.4	1.2	1.8
	BL corner	0.6	0.3	0.2	0.1	0.3
	DBP	22.1	31.9	38.8	43.8	34.2
	Sum	78.4	84.5	88.4	91.1	85.6

C.12. Usage of coding units with different modes in MV-HEVC+IPDBP and MV-HEVC codecs

Tables show usage of coding units with different modes in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec: MV-HEVC+IPDBP1, MV-HEVC+IPDBP2 and MV-HEVC+IPDBP3, for QP={22,27,32,37}. The values are averaged over all test sequences (see Annex A.1). The results are obtained using MV-HEVC configuration described in Annex B.2 for 2-view and 3-view codec setup (refer to Section 6.4).

Tab. C.24. Usage of coding units (CUs) with different modes [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (2-view case, side view).

Codec variant	CU mode	QP				
		22	27	32	37	Avg.
MV-HEVC	MERGE	52.29	68.43	76.64	82.87	70.06
	INTER-MERGE	23.56	14.26	10.31	7.48	13.90
	INTER	20.47	15.56	11.96	8.89	14.22
	INTRA	3.69	1.75	1.09	0.76	1.82
MV-HEVC +IPDBP1	MERGE	57.53	74.86	82.74	87.71	75.71
	INTER-MERGE	23.44	13.12	8.84	6.12	12.88
	INTER	15.35	10.28	7.34	5.42	9.60
	INTRA	3.68	1.75	1.09	0.76	1.82
MV-HEVC +IPDBP2	MERGE	57.64	75.04	82.94	87.97	75.90
	INTER-MERGE	23.23	12.90	8.61	5.86	12.65
	INTER	15.46	10.30	7.36	5.40	9.63
	INTRA	3.68	1.76	1.09	0.77	1.83
MV-HEVC +IPDBP3	MERGE	57.42	74.80	82.79	87.78	75.70
	INTER-MERGE	23.58	13.20	8.80	6.05	12.91
	INTER	15.33	10.26	7.33	5.42	9.58
	INTRA	3.67	1.75	1.09	0.76	1.82

Tab. C.25. Usage of coding units (CUs) with different modes [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (3-view case, *view 1*).

Codec variant	CU mode	QP				
		22	27	32	37	Avg.
MV-HEVC	MERGE	53.29	69.83	78.22	84.75	71.52
	INTER-MERGE	24.39	15.04	10.48	7.00	14.23
	INTER	20.00	14.36	10.88	8.02	13.32
	INTRA	2.33	0.77	0.41	0.23	0.93
MV-HEVC +IPDBP1	MERGE	58.68	75.93	83.68	89.04	76.83
	INTER-MERGE	23.49	13.54	8.91	5.62	12.89
	INTER	15.51	9.77	7.00	5.11	9.35
	INTRA	2.33	0.77	0.42	0.23	0.94
MV-HEVC +IPDBP2	MERGE	58.76	76.07	83.99	89.31	77.03
	INTER-MERGE	23.37	13.29	8.60	5.38	12.66
	INTER	15.56	9.86	7.00	5.08	9.38
	INTRA	2.32	0.78	0.42	0.23	0.94
MV-HEVC +IPDBP3	MERGE	58.46	75.88	83.69	89.04	76.77
	INTER-MERGE	23.80	13.56	8.88	5.58	12.95
	INTER	15.41	9.80	7.01	5.15	9.34
	INTRA	2.34	0.77	0.42	0.23	0.94

Tab. C.26. Usage of coding units (CUs) with different modes [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (3-view case, *view 2*).

Codec variant	CU mode	QP				
		22	27	32	37	Avg.
MV-HEVC	MERGE	51.88	67.45	75.29	81.43	69.01
	INTER-MERGE	22.37	12.98	9.48	6.92	12.94
	INTER	20.08	16.02	12.59	9.52	14.55
	INTRA	5.67	3.55	2.64	2.14	3.50
MV-HEVC +IPDBP1	MERGE	56.04	72.24	79.86	85.09	73.31
	INTER-MERGE	22.22	12.25	8.56	6.06	12.27
	INTER	16.08	11.97	8.93	6.71	10.92
	INTRA	5.66	3.55	2.65	2.13	3.50
MV-HEVC +IPDBP2	MERGE	55.92	72.30	79.98	85.28	73.37
	INTER-MERGE	22.33	12.10	8.36	5.86	12.16
	INTER	16.10	12.06	9.01	6.73	10.98
	INTRA	5.66	3.54	2.65	2.13	3.50
MV-HEVC +IPDBP3	MERGE	55.67	72.17	79.84	85.07	73.19
	INTER-MERGE	22.69	12.33	8.55	6.07	12.41
	INTER	15.97	11.96	8.97	6.72	10.91
	INTRA	5.67	3.55	2.64	2.14	3.50

C.13. Usage of coding units with different size in MV-HEVC+IPDBP and MV-HEVC codecs

Tables show usage of coding units with different size in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec: MV-HEVC+IPDBP1, MV-HEVC+IPDBP2 and MV-HEVC+IPDBP3, for QP={22,27,32,37}. The values are averaged over all test sequences (see Annex A.1). The results are obtained using MV-HEVC configuration described in Annex B.2 for 2-view and 3-view codec setup (refer to Section 6.4).

Tab. C.27. Usage of coding units (CUs) with different size [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (2-view case, side view).

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC	64x64	38.31	58.76	70.51	79.23	61.70
	64x32	2.82	2.25	1.75	1.38	2.05
	32x64	2.48	2.47	2.37	2.01	2.33
	32x32	25.70	19.25	15.29	11.61	17.96
	32x16	3.02	1.53	1.10	0.66	1.57
	16x32	2.83	1.83	1.35	0.90	1.73
	16x16	13.92	8.56	5.24	3.13	7.71
	16x8	1.32	0.72	0.36	0.16	0.64
	8x16	1.41	0.83	0.44	0.24	0.73
	8x8	5.99	3.00	1.31	0.59	2.72
	8x4	0.77	0.27	0.09	0.03	0.29
	4x8	0.78	0.32	0.12	0.05	0.32
	4x4	0.65	0.22	0.08	0.03	0.25
MV-HEVC +IPDBP1	64x64	41.40	64.76	77.38	85.21	67.19
	64x32	2.19	1.44	1.01	0.79	1.35
	32x64	2.22	1.92	1.63	1.26	1.76
	32x32	25.41	16.92	11.93	8.31	15.64
	32x16	2.35	1.08	0.75	0.46	1.16
	16x32	2.38	1.44	0.99	0.64	1.36
	16x16	13.74	7.69	4.28	2.44	7.04
	16x8	1.07	0.57	0.27	0.13	0.51
	8x16	1.19	0.67	0.34	0.18	0.60
	8x8	6.04	2.79	1.14	0.50	2.62
	8x4	0.69	0.23	0.08	0.02	0.26
	4x8	0.70	0.28	0.10	0.04	0.28
	4x4	0.64	0.22	0.08	0.02	0.24

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC +IPDBP2	64x64	41.74	64.98	77.51	85.21	67.36
	64x32	2.14	1.35	0.96	0.74	1.30
	32x64	2.17	1.79	1.49	1.11	1.64
	32x32	25.37	17.01	12.06	8.50	15.74
	32x16	2.28	1.04	0.73	0.45	1.12
	16x32	2.34	1.37	0.92	0.59	1.30
	16x16	13.77	7.75	4.32	2.50	7.08
	16x8	1.04	0.55	0.27	0.13	0.50
	8x16	1.16	0.63	0.32	0.17	0.57
	8x8	5.99	2.81	1.16	0.50	2.62
	8x4	0.68	0.23	0.08	0.02	0.25
	4x8	0.69	0.27	0.10	0.04	0.27
	4x4	0.63	0.22	0.08	0.02	0.24
MV-HEVC +IPDBP3	64x64	41.53	64.81	77.38	85.07	67.20
	64x32	2.23	1.45	1.00	0.78	1.37
	32x64	2.18	1.88	1.59	1.23	1.72
	32x32	25.27	16.94	12.02	8.48	15.68
	32x16	2.38	1.07	0.76	0.46	1.17
	16x32	2.38	1.40	0.94	0.63	1.34
	16x16	13.71	7.70	4.29	2.46	7.04
	16x8	1.09	0.57	0.27	0.13	0.52
	8x16	1.19	0.65	0.33	0.18	0.59
	8x8	5.99	2.80	1.15	0.50	2.61
	8x4	0.70	0.23	0.08	0.02	0.26
	4x8	0.70	0.28	0.10	0.04	0.28
	4x4	0.64	0.22	0.08	0.02	0.24

Tab. C.28. Usage of coding units (CUs) with different size [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (3-view case, *view 1*).

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC	64x64	39.08	60.95	72.91	81.33	63.57
	64x32	3.34	2.27	1.75	1.37	2.18
	32x64	2.78	2.67	2.42	2.05	2.48
	32x32	26.56	18.81	14.39	10.58	17.58
	32x16	2.88	1.53	1.00	0.55	1.49
	16x32	2.82	1.78	1.24	0.79	1.66
	16x16	13.17	7.66	4.40	2.49	6.93
	16x8	1.19	0.59	0.26	0.11	0.54
	8x16	1.32	0.75	0.38	0.20	0.66
	8x8	5.24	2.38	1.04	0.46	2.28
	8x4	0.53	0.19	0.06	0.02	0.20
	4x8	0.67	0.28	0.11	0.04	0.28
	4x4	0.43	0.13	0.04	0.01	0.15
MV-HEVC +IPDBP1	64x64	42.43	66.75	78.95	86.62	68.69
	64x32	2.50	1.42	1.01	0.78	1.43
	32x64	2.41	2.04	1.79	1.39	1.91
	32x32	26.13	16.45	11.33	7.57	15.37
	32x16	2.25	1.13	0.69	0.38	1.11
	16x32	2.39	1.45	0.97	0.60	1.35
	16x16	13.02	6.89	3.64	1.96	6.38
	16x8	0.99	0.47	0.20	0.09	0.44
	8x16	1.15	0.65	0.32	0.17	0.57
	8x8	5.23	2.21	0.90	0.38	2.18
	8x4	0.48	0.17	0.05	0.02	0.18
	4x8	0.61	0.25	0.10	0.04	0.25
	4x4	0.42	0.13	0.04	0.01	0.15
MV-HEVC +IPDBP2	64x64	42.62	66.90	79.04	86.58	68.78
	64x32	2.42	1.38	0.99	0.75	1.39
	32x64	2.42	1.92	1.60	1.22	1.79
	32x32	26.21	16.55	11.51	7.81	15.52
	32x16	2.20	1.09	0.67	0.37	1.08
	16x32	2.37	1.39	0.89	0.55	1.30
	16x16	12.97	6.95	3.70	2.01	6.41
	16x8	0.95	0.45	0.20	0.09	0.42
	8x16	1.13	0.61	0.30	0.16	0.55
	8x8	5.22	2.22	0.92	0.39	2.19
	8x4	0.47	0.16	0.05	0.02	0.18
	4x8	0.60	0.25	0.09	0.03	0.24
	4x4	0.41	0.13	0.04	0.01	0.15

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC +IPDBP3	64x64	42.45	66.72	78.89	86.45	68.63
	64x32	2.59	1.45	1.05	0.79	1.47
	32x64	2.47	1.99	1.69	1.33	1.87
	32x32	25.96	16.51	11.46	7.77	15.42
	32x16	2.28	1.13	0.69	0.39	1.12
	16x32	2.41	1.41	0.94	0.58	1.33
	16x16	12.96	6.91	3.67	1.99	6.38
	16x8	1.01	0.48	0.20	0.09	0.44
	8x16	1.16	0.63	0.32	0.16	0.57
	8x8	5.21	2.22	0.91	0.39	2.18
	8x4	0.49	0.17	0.05	0.02	0.18
	4x8	0.62	0.25	0.10	0.04	0.25
	4x4	0.41	0.13	0.04	0.01	0.15

Tab. C.29. Usage of coding units (CUs) with different size [% of picture area] in MV-HEVC and all analyzed syntax variants of MV-HEVC+IPDBP codec (3-view case, *view 2*).

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC	64x64	38.38	57.55	69.22	77.83	60.74
	64x32	2.18	2.29	1.80	1.41	1.92
	32x64	2.36	2.24	2.21	1.87	2.17
	32x32	25.09	19.67	15.76	12.30	18.20
	32x16	2.68	1.53	1.07	0.69	1.49
	16x32	2.65	1.77	1.31	0.88	1.65
	16x16	14.93	8.99	5.77	3.63	8.33
	16x8	1.27	0.72	0.40	0.21	0.65
	8x16	1.37	0.80	0.45	0.24	0.71
	8x8	6.28	3.27	1.56	0.77	2.97
	8x4	1.02	0.41	0.14	0.04	0.40
	4x8	0.79	0.33	0.13	0.05	0.33
	4x4	1.01	0.42	0.18	0.07	0.42

Codec variant	CU size	QP				
		22	27	32	37	Avg.
MV-HEVC +IPDBP1	64x64	40.93	61.77	73.91	82.10	64.68
	64x32	1.68	1.57	1.17	0.90	1.33
	32x64	2.10	1.91	1.82	1.48	1.83
	32x32	24.67	18.08	13.49	9.87	16.53
	32x16	2.18	1.18	0.81	0.51	1.17
	16x32	2.29	1.53	1.13	0.74	1.42
	16x16	14.89	8.46	5.10	3.15	7.90
	16x8	1.07	0.59	0.34	0.18	0.55
	8x16	1.21	0.70	0.39	0.21	0.63
	8x8	6.31	3.11	1.42	0.70	2.89
	8x4	0.94	0.37	0.13	0.04	0.37
	4x8	0.73	0.30	0.11	0.04	0.30
	4x4	0.99	0.41	0.18	0.07	0.41
MV-HEVC +IPDBP2	64x64	41.19	61.93	73.99	82.14	64.81
	64x32	1.67	1.51	1.15	0.87	1.30
	32x64	2.12	1.90	1.75	1.36	1.78
	32x32	24.64	18.14	13.57	10.05	16.60
	32x16	2.13	1.14	0.79	0.50	1.14
	16x32	2.33	1.51	1.07	0.68	1.40
	16x16	14.80	8.44	5.13	3.18	7.89
	16x8	1.04	0.57	0.33	0.18	0.53
	8x16	1.22	0.69	0.37	0.20	0.62
	8x8	6.20	3.09	1.43	0.70	2.86
	8x4	0.94	0.37	0.13	0.04	0.37
	4x8	0.72	0.30	0.11	0.04	0.29
	4x4	0.99	0.41	0.18	0.07	0.41
MV-HEVC +IPDBP3	64x64	40.97	61.83	73.86	82.04	64.67
	64x32	1.76	1.61	1.18	0.91	1.37
	32x64	2.18	1.91	1.80	1.47	1.84
	32x32	24.46	17.99	13.54	9.93	16.48
	32x16	2.24	1.19	0.82	0.51	1.19
	16x32	2.35	1.51	1.10	0.72	1.42
	16x16	14.75	8.45	5.12	3.17	7.87
	16x8	1.10	0.60	0.34	0.18	0.56
	8x16	1.24	0.70	0.38	0.21	0.63
	8x8	6.26	3.11	1.44	0.71	2.88
	8x4	0.95	0.38	0.13	0.04	0.38
	4x8	0.74	0.31	0.12	0.04	0.30
	4x4	0.99	0.41	0.18	0.07	0.41

C.14. Results of deviation analysis of coding and decoding time

Table presents values of parameters describing population samples of coding and decoding time obtained during deviation analysis (refer to Section 6.5). Quantity of each population sample is equal 30 (*N=30*).

Parameter describing coding or decoding time	Sample (*N*=30)					
	MV-HEVC		JMVC			
	coder	decoder	coder			decoder
			view0	*view1*	*view0* +*view1*	
Average value $\bar{x}$ [s]	1666.949	3.006	273.181	430.395	703.576	0.890
Standard deviation s [s]	2.121	0.021	0.341	0.396	0.554	0.009
99% confidence interval $c_{99\%}$ [s]	5.510	0.054	0.886	1.029	1.440	0.024
$c_{99\%}/\bar{x}$ [%]	0.331	1.800	0.324	0.239	0.205	2.675

C.15. Impact on coding/decoding time due to proposed algorithms in MVC

Tables present increase in coding and decoding time due to proposed depth-based inter-view prediction algorithms measured for MVC codec (refer to Section 6.5).

Tab. C.30. Inc_{DBP} [%] for MVC coder, related to coding time of a single view.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	1.4	3.8	2.4	2.5	2.5	1.3	3.7	2.4	2.4	2.4
GT Fly	3.9	4.1	7.3	6.8	5.5	3.8	4.0	7.0	6.6	5.3
Balloons	3.2	3.6	3.3	3.8	3.5	3.3	3.2	3.1	3.8	3.4
Newspaper	2.4	3.7	2.0	1.3	2.3	2.1	3.6	2.0	1.1	2.2
Avg.	2.7	3.8	3.7	3.6	3.5	2.6	3.6	3.6	3.5	3.3

Tab. C.31. Inc_{DBP} [%] for MVC coder, related to coding time of all views.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	0.8	2.3	1.4	1.4	1.5	0.8	2.2	1.4	1.4	1.5
GT Fly	2.1	2.2	3.9	3.7	3.0	2.0	2.2	3.8	3.5	2.9
Balloons	2.0	2.2	2.0	2.4	2.1	2.1	2.0	1.9	2.3	2.1
Newspaper	1.6	2.4	1.3	0.8	1.5	1.4	2.4	1.3	0.7	1.4
Avg.	1.6	2.3	2.2	2.1	2.0	1.6	2.2	2.1	2.0	2.0

Tab. C.32. Inc_{MCP} [%] for MVC coder, related to coding time of a single view.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	14.1	13.6	17.1	19.0	15.9	13.6	13.1	16.4	18.4	15.4
GT Fly	22.2	23.7	22.5	25.2	23.4	21.9	23.6	22.6	24.8	23.2
Balloons	11.6	12.5	13.3	14.5	13.0	11.5	12.4	13.3	14.3	12.9
Newspaper	11.8	11.2	14.0	15.6	13.1	12.2	11.3	14.1	16.1	13.4
Avg.	14.9	15.2	16.7	18.6	16.4	14.8	15.1	16.6	18.4	16.2

Tab. C.33. Inc_{MCP} [%] for MVC coder, related to coding time of all views.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	8.6	8.2	10.1	11.0	9.5	8.3	7.9	9.7	10.7	9.2
GT Fly	12.0	12.8	12.2	13.5	12.6	11.9	12.8	12.2	13.3	12.6
Balloons	7.3	7.8	8.3	8.9	8.1	7.2	7.8	8.2	8.7	8.0
Newspaper	7.8	7.3	9.0	10.0	8.5	8.1	7.4	9.2	10.3	8.7
Avg.	8.9	9.0	9.9	10.9	9.7	8.9	8.9	9.8	10.8	9.6

Tab. C.34. Inc_{DBP} [%] for MVC decoder, related to coding time of all views.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	1954	2271	2402	2517	2286	1880	2182	2317	2411	2198
GT Fly	1983	2123	2253	2285	2161	1902	2074	2149	2225	2088
Balloons	2116	2271	2396	2512	2324	2041	2187	2311	2426	2241
Newspaper	2234	2378	2504	2566	2420	2149	2303	2401	2472	2331
Avg.	2072	2261	2389	2470	2298	1993	2187	2294	2383	2214

Tab. C.35. Inc_{MCP} [%] for MVC decoder, related to coding time of all views.

Sequence	JMVC+FPDBP					JMVC+IPDBP				
	QP									
	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	3986	6025	7019	7742	6193	3767	5646	6636	7335	5846
GT Fly	4941	6134	6912	7339	6332	4689	5788	6536	6947	5990
Balloons	5239	6444	7183	7867	6683	5010	6145	6863	7455	6368
Newspaper	6001	6850	7486	7931	7067	5842	6612	7237	7683	6843
Avg.	5042	6363	7150	7720	6569	4827	6048	6818	7355	6262

C.16. Impact on coding/decoding time due to proposed algorithm in MV-HEVC

Tables present increase in coding and decoding time due to proposed depth-based inter-view prediction algorithm measured for MV-HEVC codec (refer to Section 6.5).

Tab. C.36. Inc_{DBP} [%] for MV-HEVC coder, related to coding time of all views.

Sequence	MV-HEVC+IPDBP				
	QP				
	22	27	32	37	Avg.
Poznan Street	1.1	1.0	1.1	0.8	1.0
GT Fly	0.9	0.7	0.6	0.5	0.7
Balloons	0.5	0.6	0.8	0.8	0.6
Newspaper	1.0	0.7	0.5	1.1	0.8
Avg.	0.9	0.8	0.8	0.8	0.8

Tab. C.37. Inc_{MCP} [%] for MV-HEVC coder, related to coding time of all views.

Sequence	MV-HEVC+IPDBP1					MV-HEVC+IPDBP2					MV-HEVC+IPDBP2				
	QP														
	22	27	32	37	Avg.	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	8.9	10.9	12.1	14.8	11.7	9.0	10.7	11.9	14.8	11.6	8.7	10.0	10.9	12.1	10.4
GT Fly	10.7	11.9	12.2	11.8	11.6	10.7	12.1	12.7	12.0	11.9	10.2	12.1	12.4	12.3	11.8
Balloons	9.5	10.6	13.1	14.3	11.9	9.7	11.1	13.9	15.0	12.4	9.3	10.4	11.7	11.3	10.7
Newspaper	9.2	10.6	10.6	10.4	10.2	9.3	10.7	11.2	11.1	10.6	8.9	9.8	10.4	10.2	9.8
Avg.	9.6	11.0	12.0	12.8	11.3	9.7	11.1	12.4	13.2	11.6	9.3	10.6	11.4	11.5	10.7

Tab. C.38. Inc_{DBP} [%] for MV-HEVC decoder, related to coding time of all views.

MV-HEVC+IPDBP					
Sequence	QP				
	22	27	32	37	Avg.
Poznan Street	54	62	76	83	69
GT Fly	53	57	61	76	62
Balloons	57	68	85	83	73
Newspaper	76	82	94	103	89
Avg.	60	67	79	86	73

Tab. C.39. Inc_{MCP} [%] for MV-HEVC decoder, related to coding time of all views.

Sequence	MV-HEVC+IPDBP1					MV-HEVC+IPDBP2					MV-HEVC+IPDBP2				
	QP														
	22	27	32	37	Avg.	22	27	32	37	Avg.	22	27	32	37	Avg.
Poznan Street	30	75	110	170	97	34	82	140	242	125	24	48	68	99	60
GT Fly	149	190	233	278	213	144	189	237	312	221	140	180	210	247	194
Balloons	95	155	200	208	165	91	153	209	236	172	89	139	171	184	146
Newspaper	70	97	94	69	82	84	104	127	247	141	58	79	66	50	63
Avg.	86	129	159	181	139	88	132	178	259	164	77	112	129	145	116