Light-field image and video compression for future immersive applications

Antoine Dricot

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017-ENST-0008

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Image »

présentée et soutenue publiquement par

Antoine Dricot

le 1er mars 2017

Light-field image and video compression for future immersive applications

Directeur de thèse : **Marco CAGAZZO**

Jury

M. Fernando PEREIRA, Professor, Instituto Superior Técnico	Président
M. Gauthier LAFRUIT, Professor, Université Libre de Bruxelles	Rapporteur
M. Adrian MUNTEANU, Professor, Vrije Universiteit Brussel	Rapporteur
M. Thomas MAUGEY, Research Scientist, INRIA	Examinateur
M. Marco CAGNAZZO, Associate Professor, Télécom ParisTech	Directeur de thèse
M. Joel JUNG, Senior Research Scientist, Orange Labs	Encadrant de thèse
Mme. Béatrice PESQUET, Professor, Télécom ParisTech	Encadrant de thèse
M. Frédéric DUFAUX, Research Director, CNRS, L2S, CentraleSupelec	Encadrant de thèse

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

Contents

Abstract **5**

Introduction **7**

I Context and state-of-the-art **11**

1 Principle of current video compression standards **13**
1.1 Hybrid video coding scheme 13
1.2 Some improvements of HEVC over H.264/AVC 14
1.3 Multi-view and 3D extensions of HEVC 16
1.3.1 Multi-View plus Depth format (MVD) 16
1.3.2 MV-HEVC - Multi-view extension 16
1.3.3 3D-HEVC - 3D extensions 17
1.4 Performances 18

2 Towards an end-to-end light-field system: current status and limitations **19**
2.1 Introduction 19
2.2 Sampling the light-field: capture and formats 20
2.2.1 Definition of the light-field 20
2.2.2 Super Multi-View: convergent and divergent camera arrays 20
2.2.3 Integral imaging: light-field or plenoptic cameras 22
2.2.4 Other light-field formats: Point Clouds and Meshes 23
2.2.5 Similarities, differences, and tradeoffs between formats 24
2.3 Display systems 24
2.3.1 Main light-field displays: projection-based systems 24
2.3.2 Other light-field displays 25
2.4 Processing tools 26
2.4.1 View extraction from integral images 27
2.4.2 Depth map estimation 28
2.4.3 View Synthesis 29
2.5 Light-field content compression based on current encoders 30
2.5.1 Super Multi-View compression 30
2.5.2 Integral images compression 31
2.6 Conclusion 31

II Integral imaging **33**

3 Integral images compression scheme based on view extraction **35**
3.1 Introduction . . . 35
3.2 State-of-the-art . . . 36
3.3 Proposed scheme . . . 37
3.4 Anchor selection and performance evaluation method . . . 38
3.5 Proposed methods with one extracted view . . . 42
3.5.1 Iterative methods to tune the scheme . . . 42
3.5.2 Impact of the position and size of the extracted patch . . . 49
3.6 Improvement of the filtering step . . . 51
3.6.1 Wiener Filter in integral image reconstruction . . . 52
3.6.2 Proposed Wiener filter based methods . . . 52
3.6.3 Experimental results . . . 52
3.7 Proposed methods with several views . . . 54
3.7.1 Experimental conditions . . . 54
3.7.2 Experimental results . . . 55
3.8 Combination and comparison with state-of-the-art methods . . . 56
3.9 Perspectives . . . 59
3.9.1 CU level competition with intra mode . . . 59
3.9.2 View extraction with dense disparity map . . . 59
3.9.3 Display/format scalable feature . . . 61
3.9.4 Other perspectives . . . 61
3.10 Conclusion . . . 62

III Super Multi-View **63**

4 Subjective evaluation of super multi-view compressed contents on light-field displays **65**
4.1 Introduction . . . 65
4.2 Super Multi-View display system used in our experiments . . . 66
4.2.1 Example of light-field display system . . . 66
4.2.2 Light-field conversion . . . 67
4.3 Preliminary encoding configurations experiments . . . 67
4.3.1 Experimental content . . . 67
4.3.2 Depth estimation . . . 68
4.3.3 View synthesis . . . 68
4.3.4 Group of views (GOV) . . . 69
4.3.5 Inter-view reference pictures structure . . . 71
4.4 Objective experimental results . . . 73
4.5 Subjective evaluation . . . 73
4.5.1 Experimental conditions . . . 73
4.5.2 Subjective results . . . 79
4.5.3 Impact of depth estimation and view synthesis . . . 81
4.5.4 Range of bitrate values for compressed light-field content . . . 82
4.5.5 Comparison between objective and subjective results . . . 83
4.5.6 Impact of the light-field conversion step . . . 86

4.5.7 Comments on motion parallax . . . 87
4.6 Conclusion . . . 87

5 Full parallax super multi-view video coding 91
5.1 Introduction . . . 91
5.2 State-of-the-art . . . 92
5.2.1 Multi-view video coding standards and specific coding tools . . . 92
5.2.2 Improvement for full parallax configuration . . . 92
5.3 Proposed inter-view reference pictures configuration . . . 93
5.3.1 Reference and proposed schemes . . . 93
5.3.2 Experimental results . . . 96
5.4 Adaptation and improvement of inter-view coding tools . . . 97
5.4.1 Merge candidate list improvement . . . 97
5.4.2 Inter-view derivation of the second DV . . . 97
5.4.3 Experimental results . . . 98
5.5 Conclusion . . . 99

6 On the interest of arc specific disparity prediction tools 101
6.1 Motivations . . . 101
6.2 State-of-the-art . . . 101
6.2.1 Anchor results . . . 101
6.2.2 Generalization of 3D-HEVC coding tools . . . 102
6.3 Comparison of coding performances between arc and linear content . . . 102
6.4 Analysis of the content . . . 103
6.4.1 Disparity in arc content . . . 103
6.4.2 Percentage of the total bitrate dedicated to motion/disparity . . . 104
6.5 Proposed methods and preliminary results . . . 107
6.5.1 Modification of NBDV . . . 109
6.5.2 Modification of AMVP . . . 110
6.6 Conclusion . . . 112

7 Compression scheme for free navigation applications 113
7.1 Introduction . . . 113
7.2 State-of-the-art . . . 114
7.3 Performances comparison with existing encoders in different configurations 116
7.3.1 Tested structures . . . 116
7.3.2 Performance evaluation . . . 117
7.3.3 Experimental conditions . . . 118
7.3.4 Experimental results . . . 118
7.3.5 Results analysis . . . 122
7.4 Conclusion and perspectives . . . 125

8 Conclusion 127

Appendix: Proposed compression scheme for free navigation applications 131
8.1 Proposed coding scheme . . . 131
8.1.1 Coding structure . . . 131
8.1.2 Example with the basic method . . . 131
8.1.3 Proposed method . . . 134

8.2 Conclusion and perspectives 137

List of publications **139**

Bibliography **141**

Abstract

Evolutions in video technologies tend to offer increasingly immersive experiences. However, currently available 3D technologies are still very limited and only provide uncomfortable and unnatural viewing situations to the users. The next generation of immersive video technologies appears therefore as a major technical challenge, particularly with the promising light-field (LF) approach.
The light-field represents all the light rays (i.e. in all directions) in a scene. New devices for sampling/capturing the light-field of a scene are emerging fast such as camera arrays or plenoptic cameras based on lenticular arrays. Several kinds of display systems target immersive applications like Head Mounted Display and projection-based light-field display systems, and promising target applications already exist. For several years now this light-field representation has been drawing a lot of interest from many companies and institutions, for example in MPEG and JPEG groups.
Light-field contents have specific structures, and use massive amounts of data, that represent a challenge to set up future services. One of the main goals of this work is first to assess which technologies and formats are realistic or promising. The study is done through the scope of image/video compression, as compression efficiency is a key factor for enabling these services on the consumer markets. Secondly, improvements and new coding schemes are proposed to increase compression performance in order to enable efficient light-field content transmission on future networks.

Introduction

Recent evolutions in video technologies tend to provide increasingly immersive experiences to the viewer. On the one hand, Ultra High Definition (UHD), with 4K and 8K resolutions, High Frame Rates (HFR), High Dynamic Range (HDR) and also Wide Color Gamut (WCG) are progressively bringing 2D video towards the limits of the perception of the Human Visual System (HVS). However, on the other hand, currently available 3D video technologies fail to massively reach the consumer market, and are not accepted by users because they are still very limited and do not provide comfortable enough experiences.

Stereoscopic 3D only uses 2 views (one for each eye) and therefore cannot provide motion parallax, i.e. it is not possible for the viewer to change his point of view (for example by moving in front of the screen to gather more information about the scene). This psychological cue that contributes to the perception of depth is however a key element for immersive applications [1]. Moreover, the use of glasses causes discomfort, and the conflict between the accommodation distance (eyes are focused on the screen) and the convergence distance (eyes converge on the image of the object possibly in front of or behind the screen) provides an unnatural viewing situation and is reported to cause headaches and eyestrain (sometimes referred to as cybersickness). Auto-stereoscopic display systems use more than two views (e.g. from 8 to 30) but are still limited by the lack of smooth motion parallax. The viewing positions that allow the users to watch the scene conveniently (i.e. with a correct perception of depth and without artefact) are restricted to certain areas called sweet spots. These unnatural perception stimuli are severe limitations that alter the quality of the visualization and make the viewing experience unrealistic.

The next generation of immersive video technologies appears therefore as a major technical challenge, particularly with the *light-field* (LF) approach that shows up as one of the most promising candidate solutions. A light-field represents all the light rays in a scene, i.e. rays at every points in space and in every directions, and thus is a function of two angles (ray direction) and three spatial coordinates. This 5-dimensional function is called plenoptic function [2][3]. Conceptually, as 2D video provides a basic sampling of the light-field offering a view of the scene from one angle, light-field acquisition devices provide a wider and denser sampling that offers several views of the scene (i.e. capturing the rays coming from several angles).

For several years now this so-called light-field representation has been drawing a lot of interest from the experts in many companies and institutions. Efforts have been made to assess the potential of the emerging devices and formats, for example by Ad-Hoc Groups in MPEG [4], particularly *Free Viewpoint Television* (FTV) [5] and *Virtual Reality* (VR) groups, in JPEG with *JPEG Pleno* [6], and more recently with a *Joint ad hoc group for digital representations of light/sound fields for immersive media applications* [7]. New devices have reached the market or are emerging fast. Capture devices are now available like camera arrays (e.g. Google Jump/GoPro Odyssey [8][9], Lytro Immerge [10]) or

plenoptic cameras based on lenticular arrays (e.g. Lytro Illum [10], Raytrix [11]). Several kinds of display systems target immersive applications like Head Mounted Display (e.g. Samsung Gear VR [12], Oculus Rift [13]) and projection-based LF display systems (e.g. Holografika's Holovizio [14]). Moreover, appealing and promising target applications already exists (e.g. 360° video, already implemented in Youtube [15] and Facebook [16], that is a first step before 360° virtual reality) or are developed (e.g. binocular stereoscopic 360°, immersive telepresence, free navigation, etc.). Light-field image and video contents required to create these immersive experiences have specific structures and formats, and use a massive amount of data, that represent a challenge for future transmission on our networks and to set up future services.

The main goal of our work is to study the feasibility of implementing new immersive light-field video services. This study is done through the scope of image and video compression, as compression efficiency is a key factor for enabling these services on the consumer and industry markets. We first aim to assess which technologies and formats are realistic and which ones are promising for light-field acquisition, display, and compression considering several target applications. Secondly, we propose improvements of the state-of-the-art compression technologies and new coding schemes in order to increase the compression performance and to enable efficient light-field content transmission on future networks. This manuscript is organized as follows.

- **Part I** is dedicated to the description of the context of our work.
  - In Chapter 1, we describe some basic principles of image and video compression that are implemented in current encoders and that are useful to understand the technical work described in this thesis.
  - Chapter 2 sets up the context of our work by providing an overview of state-of-the-art light-field technologies from capture to display, including several processes like rendering. This chapter particularly emphasizes on Integral Imaging and Super Multi-View video (SMV) technologies, that are based on microlens arrays and camera arrays respectively, and that are the main focus of our technical contributions.
- **Part II** is focused on our contributions on integral images (or plenoptic images) compression. This representation provides a dense sampling of the light-field in a narrow angle of view, with a challenging structure for compression.
  - Chapter 3 proposes an original integral images compression scheme based on view extraction. It takes advantages of the view extraction process to reconstruct a reliable predictor and creates a residual integral image that is encoded. We first propose several iterative methods to select the most efficient configuration, using a rate-distortion optimization (RDO) process to avoid exhaustive search methods. Additional runtime savings are then reported by exploring how the different parameters interact. We assess the impact of the position and size of the patches used for the view extraction on the compression performance. We propose to improve the method with advanced filtering techniques. Methods based on the Wiener filter are used to improve the reconstruction step. The performance of the scheme using several extracted views is studied. Finally, the behavior of this method in competition or in collaboration with state-of-the-art methods is assessed.
- Because integral imaging only captures the light-field under a narrow angle of view, it cannot be used for applications where a large angle of view is required, such as

Free Navigation for example. Therefore in **Part III**, we also study the compression of Super Multi-View content, that provides a sparser sampling of the light-field but with a large baseline.

- In Chapter 4, we present a subjective quality evaluation of compressed SMV video content on a light-field display system. While the in-depth understanding of the interactions between video compression and display is of prime interest, evaluating the quality of light-field content is a challenging issue [7]. The main goal of this study is to assess the impact of compression on perceived quality for light-field video content and displays. To the best of our knowledge, the work presented in this chapter is the first to carry out subjective experiments and to report results of this kind.

- Chapter 5 is focused on the compression of full parallax SMV content, e.g. content captured with a 2D camera array (with cameras arranged in horizontal and vertical dimensions). Multi-view encoder extensions are adequate to encode SMV content with horizontal parallax only. Modifications of these encoders have to be applied to encode content with full parallax. We first propose an efficient inter-view prediction scheme to exploit horizontal and vertical dimensions at the coding structure level. Then we propose improvements of inter-view coding tools to exploit the two dimensional structure also at the coding unit level.

- Chapter 6 reports results from a study that is focused on the impact of the arc camera arrangements (i.e. instead of typical linear camera arrays) on the compression performance. The performances of existing coding technologies on linear and on arc camera arrangements are first compared. Then we propose perspectives to improve specifically the performance in the arc case (without degrading it for the linear case).

- In Chapter 7, we study the compression of SMV content targeting Free Navigation (FN) applications. We focus on applications where all the views are encoded and sent to the decoder, and the user interactively requests to the decoder a point of view to be displayed (e.g. on a state-of-the-art 2D display). We first compare the performances of state-of-the-art coding methods based on current multi-view encoders. Performance evaluation is based on the tradeoff between compression efficiency (i.e. lowest bitrate possible) and degree of freedom (i.e. the ability for the user to change the viewpoint, that mainly depends on the decoder capability and the number of pictures to decode in order to display one). Additionally, we propose in an appendix chapter a Free Navigation coding scheme that performs redundant encodings, thus allowing the users to shift viewpoint without decoding additional views.

- Conclusions and perspectives are finally drawn in Chapter 8, followed by a list of the publications resulting from the work presented in this manuscript.

Part I

Context and state-of-the-art

Chapter 1

Principle of current video compression standards

In this chapter we provide an overview of the principles of video compression and also focus on specific tools. The goal is to provide the reader with the description of some fundamental processes that are helpful to understand and appreciate the technical work of our contributions in the next chapters of this manuscript. We first discuss the basic structure of hybrid video encoders like HEVC [17] (High Efficiency Video Coding) and its predecessor H.264/AVC [18] (Advanced Video Coding) by briefly describing the main encoding steps. Secondly we emphasize on specific tools by describing differences and improvements of HEVC against H.264/AVC. Finally, we provide a concise description of the multi-view and 3D extensions of HEVC [19], respectively MV-HEVC and 3D-HEVC.

1.1 Hybrid video coding scheme

Hybrid video coding has been the basis for all video coding standards since ITU-T H.261 in 1989 [21]. HEVC (like its predecessor H.264/AVC) is also based on this concept, which is illustrated in Figure 1.1 [20]. The pictures of the original video sequence are given as input signal to the encoder. A prediction signal is obtained from information that has already been encoded and reconstructed (i.e. available both at encoder and decoder) and is subtracted from the input signal. Resulting prediction errors are represented in the residual, that is transformed, quantized, and encoded into the bitstream. The prediction parameters required at the decoder side to perform the same prediction are also encoded. Blocks included both at encoder and decoder are represented inside the gray box in Fig. 1.1.

Input pictures are partitioned into blocks that can be predicted using either intra or inter modes. In intra mode, pixels values in a block are predicted using spatially neighboring pixels (i.e. within the current picture). In inter mode, a block is predicted by a reference block in a temporal reference picture, i.e. with a different Picture Order Count (POC), as illustrated in Figure 1.2. Inter prediction is referred to as motion compensated prediction. The displacement between the predictor block in the reference picture and the current block is interpreted as the motion of this area between the two pictures, and is represented by a Motion Vector (MV). At the encoder, the selection of the best prediction mode is driven by a lagrangian Rate-Distortion Optimization process (RDO) that takes into account the degradation of the reconstructed picture compared to the original one, and the cost required to encode the residual signal and all the prediction information (i.e.

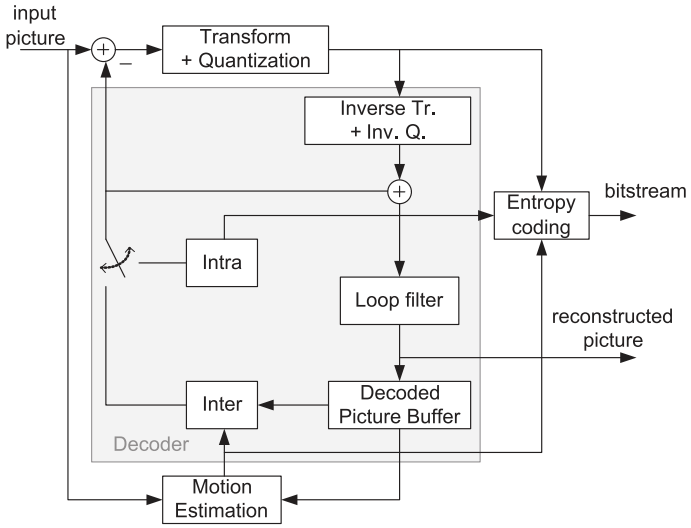


Figure 1.1: Hybrid video coding block diagram [20]

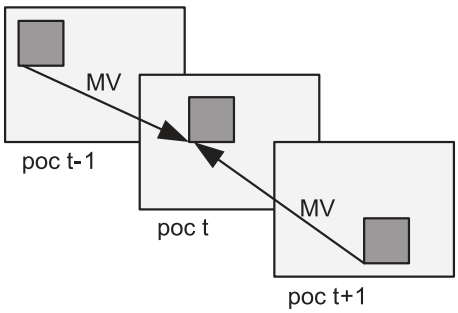


Figure 1.2: Motion compensated prediction

directions for intra mode, motion vectors and reference indexes for inter mode).

1.2 Some improvements of HEVC over H.264/AVC

A first version of the HEVC standard (also known as H.265 and MPEG-H Part 2) has been finalized in January 2013 and published in June 2013, 10 years after its widely used predecessor H.264/AVC. Section 1.1) describes the general scheme of hybrid video encoders. In this section we emphasize on some improvements and particularities of HEVC over H.264/AVC, as a way to describe specific aspects of the encoder with more details.

Performances of HEVC offer a gain of 50% compared to H.264/AVC, i.e. for a given image quality level, the required bitrate is two times smaller on average. This performance comes at the cost of an increase in complexity. In [17], it is mentioned that HEVC's decoder implementation complexity (using modern processing technology) is not a major burden when compared to H.264/AVC, and that encoder complexity is also manageable. Details on implementation complexity are given in [22]. The improvement is not due to a modification of the encoding structure (see Sec. 1.1), but rather to several changes distributed upon the whole set of coding tools. For example here we can cite first the partitioning of the pictures. As opposed to the traditional macroblock in H.264/AVC

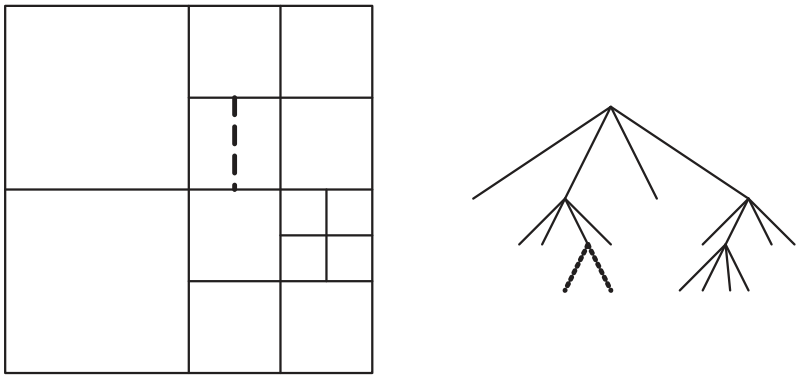


Figure 1.3: Quad-tree partitioning [17]

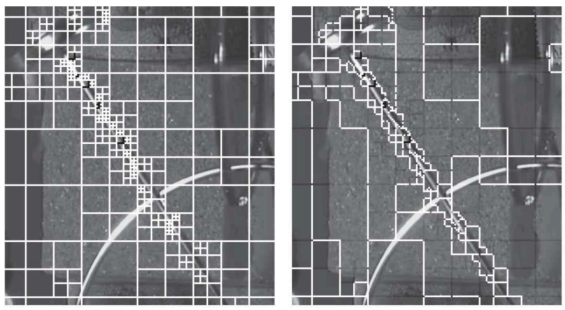


Figure 1.4: *left)* Quad-tree partitioning, *right)* CUs with same motion parameters [25]

using a fixed array size, the Coding Unit (CU) in HEVC supports variable sizes and is therefore adapted to the content of the picture. Partitioning is also more flexible with CUs, Transform Units (TUs) and Prediction Units (PUs) that are organized in quad-trees (see Fig. 1.3 and Fig 1.4). Some other improvements do not change anything conceptually and are just an increase in precision and complexity that is made possible with the evolution of hardware capacity, i.e. faster processors and larger storage. As in H.264/AVC, the precision of motion estimation goes up to a quarter-sample position (i.e. one fourth pixel) for inter mode, and the filtering for sample interpolation is improved with a eight-tap filter and a seven-tap filter, respectively for the half-sample positions for the quarter-sample positions. The number of prediction directions in intra mode is increased from 8 to 35 sub-modes. Additionally, advanced prediction tools have been added. Advanced Motion Vector Prediction (AMVP) is a tool, derived from the work of Laroche [23][24], that predicts the current MVs from the MVs used to encode neighboring CUs. Similarly, Merge [25] mode allows the current CU to copy prediction parameters (MVs and reference indexes) from temporal or spatial neighbors in a candidate list. The same candidate list is built at the decoder side, and only the index of the candidate selected by the encoder is transmitted to the decoder. This mode is very efficient in zones with homogeneous motion, as illustrated in Fig. 1.4. Other improvements are brought to parallel processing (Tiles, Wavefront, Slices) and also to the structure of the bitstream (in Network Abstraction Layers, NAL) for examples. A complete overview of HEVC is given in [17].

Figure 1.5: Texture image and associated depth map (*Poznan Blocks*)

1.3 Multi-view and 3D extensions of HEVC

1.3.1 Multi-View plus Depth format (MVD)

Multi-view video content consists of several video sequences representing the same scene, but captured from different points/angles of view by two or more cameras. This kind of content usually targets 3D stereoscopic (2 views) and autostereocopic (around 10 views) display systems that provide a visualization of the scene in relief (although with a lot of limitations as mentioned in our Introduction Chapter). These views present strong correlations that can be exploited by dedicated encoders [26][19] (see Sec. 1.3.2 and Sec. 1.3.2). The multi-view format can be extended to Multi-View plus Depth format (MVD), where the content also includes the depth maps associated to the views. In MVD format, the view is also referred to as texture. Depth maps are gray level images that represent the distance of the objects from the camera, as illustrated in Figure 1.5. They can be estimated from textures or captured by dedicated sensors. From textures and depth maps it is possible to synthesize additional intermediate views [27]. It is therefore possible to reduce the total bitrate required to encode multi-view content by encoding only a subset of the views with their associated depth maps. At the decoder side, the views that were skipped (i.e. not encoded) can be synthesized from the decoded textures and decoded depth maps. We provide further detailed descriptions about depth maps and view synthesis in Chapter 2.

1.3.2 MV-HEVC - Multi-view extension

MV-HEVC is the multi-view extension of HEVC dedicated to the encoding of multi-view video content. It is the equivalent of MVC for H.264/AVC. MV-HEVC does not provide specific additional prediction tools, but rather some syntax elements that enable inter-view prediction. Inter-view prediction or disparity compensated prediction, is based on the same algorithm as motion compensated prediction. In practice the main difference

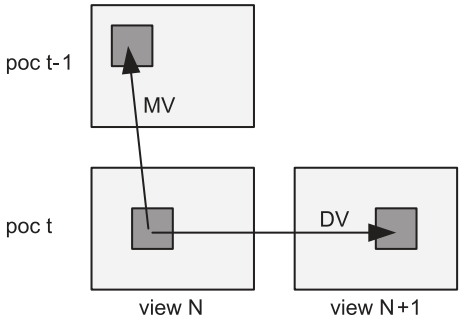


Figure 1.6: Motion and disparity compensated predictions

is the use of a reference picture taken at the same time, i.e. that has the same Picture Order Count (POC) index, but from a different view (instead of a picture from the same view with a different POC for temporal prediction). Vectors are called Disparity Vectors (DV) in that case instead of Motion Vectors (MV). Indeed, in temporal prediction the vector represents the motion of a block, i.e. the displacement of an object between two given time instants (or two pictures). In inter-view prediction, the vector represents the disparity between two blocks, i.e. the change of position of an object in the frame due to the different points of view. This is illustrated in Figure 1.6.

1.3.3 3D-HEVC - 3D extensions

3D-HEVC is dedicated to the encoding of multi-view and 3D content. Its design is mainly oriented towards the MVD format. It provides additional specific coding tools. Some are advanced inter-view prediction tools related to textures, e.g. Neighboring Block Disparity Vector (NBDV) and Inter-View Motion Prediction (IVMP), that are both described in Chapter 5, and Advanced Residual Prediction (ARP), or Illumination Compensation (IC). Inter-component prediction tools can use information from the texture encoding parameters to encode the depth maps, e.g. Quad-Tree Limitation and Predictive Coding (QTL/PC) [28]. Also related to the MVD format, specific depth map coding tools are added, e.g. Intra Wedgelet Mode [19]. And finally, synthesis based coding tools, e.g. View Synthesis Prediction (VSP) [19], takes advantages of the synthesis process from textures and depth maps. 3D-HEVC is optimized for MVD formats but is also more efficient than MV-HEVC for textures only.

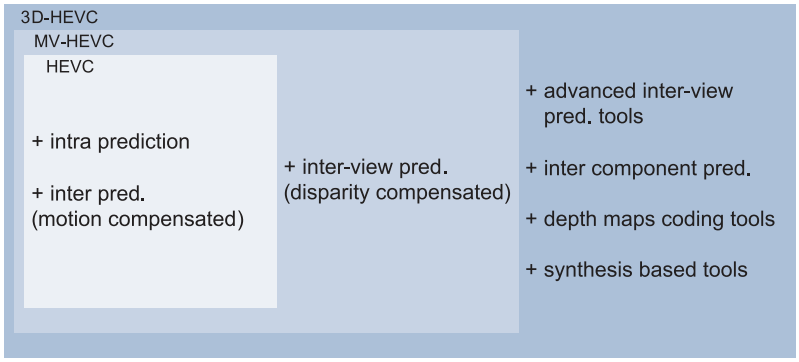


Figure 1.7: HEVC, MV-HEVC and 3D-HEVC

1.4 Performances

As illustrated in Figure 1.7, the extensions described in this chapter are built on top of each other. 3D-HEVC includes MV-HEVC syntax elements, and MV-HEVC includes HEVC coding tools. These encoders are currently providing the best performance in terms of compression efficiency compared to other state-of-the-art compression techniques in general. For 2D video, HEVC provides 50% gains over its predecessor H.264/AVC. Although the successor of HEVC is already in preparation, with many new coding tools implemented in the Joint Exploration Test Model [29] (JEM) and providing large gains over HEVC (26% at the time of writing this manuscript), this is still an exploration phase with a large increase in complexity, and the standardization process has not started yet. Anchor results for multi-view and 3D extensions of HEVC are reported in [19]. The results depend on the number of coded views. When two texture views are encoded, MV-HEVC provides around 30% average gains over the simulcast case (i.e. each view is encoded independently with HEVC). This gain is brought up to approximately 70% when taking into account only the enhancement view, i.e. the view that benefits the inter-view prediction, and not the base view that has to be coded with HEVC. This second results is of particular interest in our case, because light-field content can include a large number of views, therefore the expected gains are even larger. Finally 3D-HEVC provides additional 19% gains over MV-HEVC in the cases where three textures and associated depth maps are encoded, with six synthesized views. HEVC based compression is de facto the anchor for light-field content encoding and is used for comparison with the methods that we further propose in this manuscript.

Chapter 2

Towards an end-to-end light-field system: current status and limitations

2.1 Introduction

In this section, we provide a wide overview of existing light-field technologies. We address the elements that would compose an end-to-end light-field system from capture to display, and discuss some of the bottlenecks and key factors for its development. The question of the requirements for light-field representations has recently been extensively studied, for example in [30] and [31], and a similar study has been done recently in [7] (although with a larger scope, e.g. including audio), where all the elements numbered in Figure 2.1 are discussed in order to find commonalities between the different available formats and technologies. In Figure 2.1, the blocks *Sensor* and *Sensed data converted* correspond to the capture of the light-field from a scene and to its representation in a given format. The *Encoder* and *Decoder* blocks are dedicated to the compression of this data, hence to the main focus of the work presented in this manuscript. Finally the *Renderer* and *Presentation system* blocks represent the conversion of the decompressed data to a target format that depends on the application, and the corresponding display system.

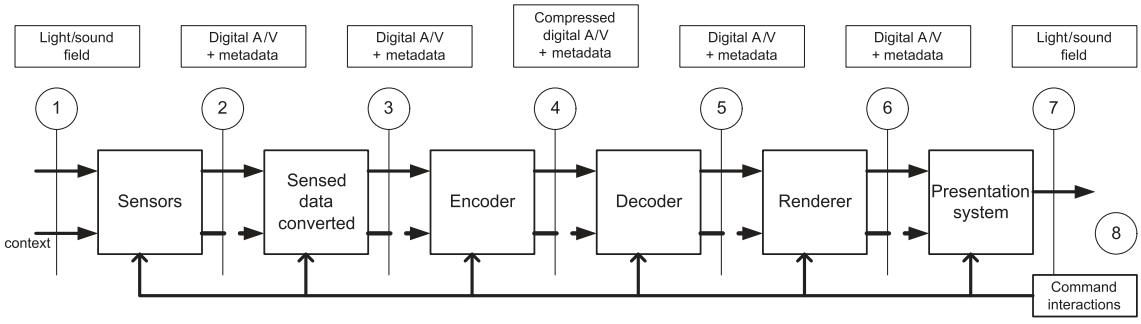


Figure 2.1: Generic end-to-end light-field workflow [7] (*source*: L. Chiariglione, MPEG).

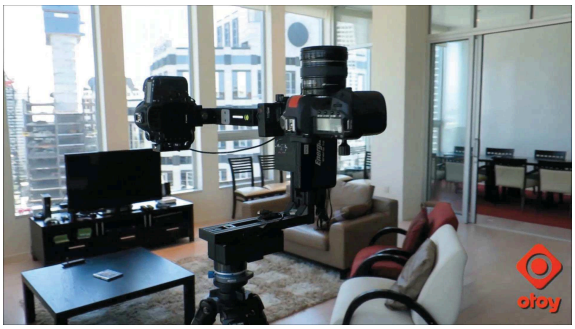


Figure 2.2: Still light-field capture system (Otoy) based on a single moving camera [32].

2.2 Sampling the light-field: capture and formats

2.2.1 Definition of the light-field

A light-field represents all the light rays in a scene, and thus is a function of two angles (ray direction) and three spatial coordinates. This 5-dimensional function is called plenoptic function [2][3]. Depending on the context, this function can also be used with seven dimensions when the time and wavelength (i.e. color) are taken into account. A light-field representation of a scene can be obtained by sampling and capturing a subset of the rays from different points/angles of view. Several techniques are currently used for this purpose, mostly based on a camera array or a single camera coupled to a microlens array, as described in the following sections. The elements presented in the following of this section relate to the *Sensor* and *Sensed data converted* blocks in Fig. 2.1.

2.2.2 Super Multi-View: convergent and divergent camera arrays

A static light-field (i.e. still image of a scene from several angles) can be captured using one single camera moving along and/or around one or several axis, e.g. as demonstrated by Otoy [32] with the system illustrated in Figure 2.2. This method is however limited to still scenes only and is also time consuming. In order to capture a moving scene, it is required that the several angles of view are captured simultaneously.

A first intuitive approach to instantly sample the light-field consists in capturing at a fixed point the light rays coming from all around. This way of acquisition is tightly linked to 360° viewing applications where the user can change the orientation of the visualization but not the position. It can basically be done by using a camera with one or two large angle lenses (e.g. fish-eyes). Several acquisition devices are already available like Ricoh Theta [33] or Kodak SP360 [34], shown in Figure 2.3. All the information is captured onto one sensor with this kind of devices, hence the resolution is limited.

It is also possible to increase the resolution of the acquired light-field with several sensors using camera arrays. Many arrangements are possible for camera arrays. Existing setups include 1D (horizontal only) or 2D (horizontal and vertical) linear arrays, convergent or divergent circular arrangements, as well as unstructured arrays. Divergent (or omnidirectional) camera arrays can also provide views from all around (i.e. 360°), as illustrated in Figure 2.4. For example GoPro Odissey [9] (based on Google Jump [8]) is an horizontal structure based on 16 GoPro cameras. Lytro Immerge [10] has five layers of similar divergent camera arrays (i.e. an horizontal and vertical structure) and is provided with its own stack of servers for storage and synchronization. Several solutions

Kodak SP360 [34]

Ricoh Theta [33]

Figure 2.3: Single or double lens based 360° acquisition devices

GoPro Odissey (based on Google Jump) [8][9]

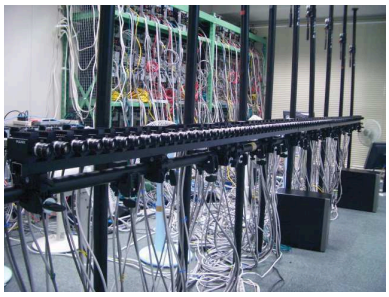
Lytro Immerge [10]

Figure 2.4: Divergent (omnidirectional) camera arrays

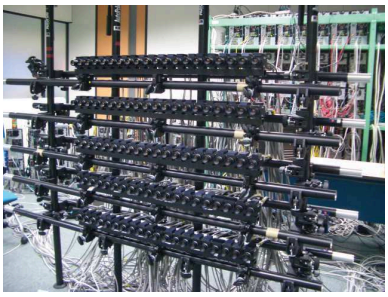
based on the same principle exist such as 360Rize (formerly 360 Heroes) [35], that offers different camera structures with for example coupled cameras for stereoscopy, Samsung Beyond [36], based on 16(+1) cameras, and also JauntVR [37] or Omnicam (HHI) [38]. After the capture, the technical challenge consists in mapping the different views into one image that can be fed to existing virtual reality (VR) viewers and display systems. This operation is called stitching and the goal is to process the views in order to match borders and overlapping regions in a way that makes transitions as smooth and as imperceptible as possible, as illustrated in Figure 2.5. Efficient stitching solutions currently exist such as Video Stitch [39].

A counterpart of this approach is the convergent camera array, where cameras are set around (or in front of) the scene. Fujii Laboratory at Nagoya University has implemented

Figure 2.5: Example of 360° content



1D linear



2D linear

Figure 2.6: Fujii Laboratory camera arrays at Nagoya University [40]

several types of camera arrays [40], as shown in Figure 2.6, that have been used to provide SMV content to the community for research purpose [41]. Other companies or institutions have presented examples of camera rigs, for example like Technicolor with 4×4 cameras or HHI Fraunhofer. Although they provide a large number of views (e.g. from 80 to 100 for Nagoya University's arrays) with good resolution (i.e. one camera/sensor per view), SMV convergent camera arrays present obvious technical drawbacks. They are costly and bulky, controlled by stack of servers, and therefore complicated to set-up and to move. Moreover, the camera synchronization, color correction, and storage are operations that increase in complexity when the number of cameras gets larger. Non-professional camera array demonstrations are now spread over the Internet [42] with limited numbers of affordable cameras (e.g. based on 15 GoPro cameras), that provide SMV advantages while possibly limiting the aforementioned drawbacks.

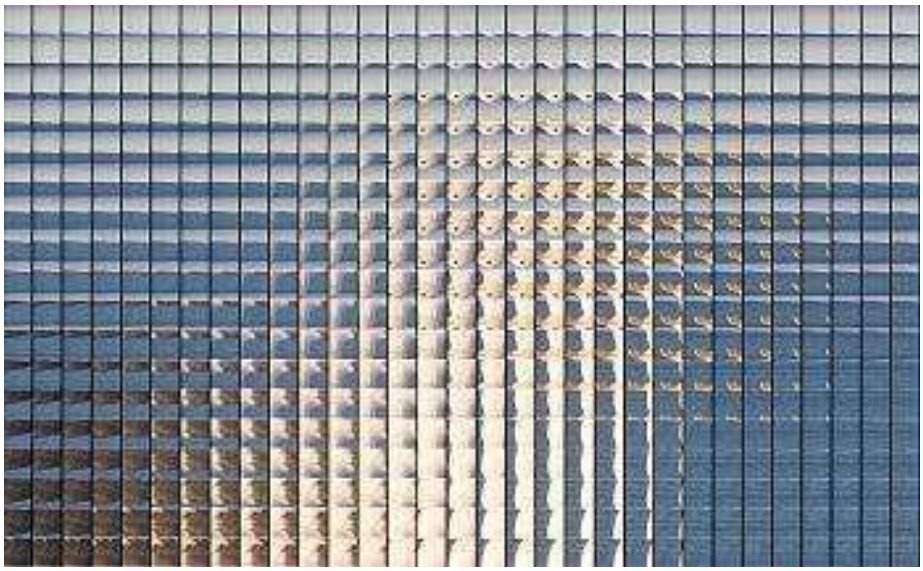
2.2.3 Integral imaging: light-field or plenoptic cameras

Integral imaging, also called plenoptic or holoscopic imaging, is another way of sampling the light-field. This technology is based on plenoptic photography [43]. Integral imaging acquisition uses a lenticular array set in front of a single camera device. This lenticular array is composed of a large number of micro-lenses, that can have a round, hexagonal or square shape, and can be aligned in rectangular grid or in quincunx. The resulting integral image consists of an array of Micro-Images (MIs, sometimes referred to as elemental images) as illustrated in Figure 2.7.

Each micro-lens produces one MI, and each MI contains the light information coming from several angles of view. Integral images can be either orthoscopic or pseudoscopic. In the case of pseudoscopic content, the MIs are flipped horizontally and vertically (as shown in Fig. 2.7, where the darker blue of the sea is above the lighter blue of the sky in the MIs). This characteristic has an impact on the processing of the picture and on the display system requirements (see Sec. 2.3).

In [44], Georgiev and Lumstaine describe the focused plenoptic camera. Traditional plenoptic cameras focus the main lens on the micro-lenses and focus the micro-lenses at infinity. In the focused plenoptic camera the main lens is focused well in front of the lenticular array, which is in turn focused on the image formed inside the camera, so that each micro-lens acts as a relay system of the main lens. This configuration allows a trade-off between spatial and angular information inside each MI (see Sec. 2.4.1).

Examples of plenoptic hand-held cameras exist on the consumer market provided by companies like Lytro [10] and Raytrix [11], as shown in Figure 2.8. Test sets composed



(a) Close-up on Micro-Images,

(b) Rendering of the original scene

Figure 2.7: An integral image - *Seagull* [45]

of integral images have also been made available, e.g. by Todor Georgiev on his website [45] or by the ICME 2016 Grand Challenge on Light-Field image compression [46] that provides images taken with the Lytro Illum camera.

2.2.4 Other light-field formats: Point Clouds and Meshes

As opposed to image-based representations like Integral Imaging and Super Multi-View, the light-field can also be represented with object-based (or geometry-based) methods [7] such as 3D meshes and Point Clouds. 3D meshes [47] are a typical data structure used for Computer Graphics and Computer Generated content. Object structures are represented for example by triangular meshes on which 2D textures are applied.

Point Clouds is another object based representation, that can be used for CG content as well, but can also be captured with dedicated depth cameras (e.g. time-of-flight scanners). A point cloud is a collection of points in 3D space, each holding depth and color information in all directions around the central acquisition position. Each point has 3 coordinates

Lytro Illum [10]

Raytrix [11]

Figure 2.8: Examples of plenoptic cameras

(x, y, z) and can be thought of as a 2D pixel (x, y) of a photograph that is pushed into the depth (z). A technique called splatting is used to fill the holes in the rendering of Point Clouds. Each point is extended with a rectangular or circular shape, either frontal to the viewing direction or oriented with the surface normal, so that overlaps between adjacent splats hide the holes.

2.2.5 Similarities, differences, and tradeoffs between formats

Image-based (Integral imaging, Super Multi-View) and object-based (Point Clouds, 3D meshes) formats all provide light-field representations because they sample a subset of the light-field of a scene. Conversions from one representation to another are technically possible, e.g. from Point Clouds to depth maps or from an integral image to views of the scene. Because the points emit light rays all around, and because the splatting creates a union of all these rays overall, a correspondence between Point Clouds and light-field is clearly suggested [7]. However, the sampling is done in different ways, implying the following tradeoffs. Point Clouds can theoretically provide the wider sampling as a large number of rays is captured, however holes in between points have to be filled to obtain a view of the scene. Therefore a very large amount of data has to be captured to provide a dense sampling. Integral imaging and Super Multi-View capture images of the scene from several angles of view in a more straightforward way. Using a camera rig allows to obtain a wider baseline (e.g. several meters) than using an holoscopic camera for which the baseline, hence the angle, are limited by the size of the micro-lens array. With holoscopic cameras, the resolution of the viewpoint images is limited because the same sensor is shared between all the captured views, while with a camera rig the full resolution of each camera is used for each view. Finally, holoscopic cameras allow a denser sampling of the light-field, because with a camera rig the distance between each view is limited by the size of the cameras.

2.3 Display systems

2.3.1 Main light-field displays: projection-based systems

In this section we focus on the *Renderer* and *Presentation system* blocks in Figure 2.1. Similarly to the variety of formats and representations for light-field content, several kinds

of display systems exist based on various technologies. We first cite here the projection-based Super Multi-View displays, often just referred to as light-field displays, as they are the most advanced and spread in the domain. The main examples are the Holovizio displays provided by Holografika [14]. These systems are based on a large number of projection units (e.g. up to 80) and screens with anisotropic properties (i.e. that reflects the light according to the angle of projection). SMV content is taken as input (e.g. 80 views), and converted to another form of representation called light-field slices by a dedicated internal process. The conversion depends on the characteristics of the display system such as the resolution, size, arrangement/positions of the projection units, angle of view, and field of depth. Each projection unit takes a light-field slice as input and projects it onto the screen, that separately reflects parts of the picture in the target direction. As a result, the user perceives the part of the content that represents one point of view of the scene depending on his position. The scene appears therefore displayed in 3 dimensions, with depth going from behind to the forefront of the screen, and the viewer is able to move around the object to take benefits of the motion parallax without wearing glasses or being tracked.

Holovizio systems range from tiny screens dedicated to vehicles (e.g. inside cars) to large cinema-like systems with screens sizing up to 3×5 meters, offering a range of various target applications with intermediary sizes in between that are closer to a common TV set configuration. Additionally to the projection of Super Multi-View content, these displays can take as input Computer Generated content in the form of Point Clouds for example, like the output of an OpenGL video games or data from Google Maps, opening another field of interactive applications.

These systems are currently considered as the most advanced of their kind, and although they are already available for sale, there are still limitations, as for example visual artefacts that prevent to provide a truly immersive experience, especially for natural content (see Chapter 4). Their usage requires large storage and costly computing processes that are performed by servers (provided as part of the display system when purchased). This aspect makes the whole system bulky and not yet affordable for common users. Therefore they are currently used mostly in universities and laboratories, and not yet ready to reach the living rooms of consumers before several years.

As storage, transmission, and lack of content are the main obstacles to a larger development, the improvement of compression efficiency and representation for light-field content is a key element to trigger the spread of this technology. The processes related to the conversion and projection steps are described in greater technical details in Chapter 4 as the subjective evaluation experiments described in that chapter were performed in collaboration with Holografika on one of the Holovizio systems.

2.3.2 Other light-field displays

With Super Multi-View, Integral Imaging is the other format that offers interesting perspectives for light-field representation dedicated to immersive applications. Current applications concern mostly plenoptic photography, with the rendering of one 2D picture of the scene, that can be refocused, and rendered with different angles of view and depths of field. However, other use cases are foreseen involving light-field display systems based on a lenticular array, similarly to the capture device. Several systems have been proposed or mentioned in the literature, as for example by NHK [48]. However, most of these displays are experimental or prototypes. Limitations still have to be overcome in order to

Samsung Gear VR [12]

Oculus Rift [13]

Figure 2.9: Examples of Head Mounted Displays

make these systems realistic and ready for the users. As integral images can be either orthoscopic or pseudoscopic (i.e. the MIs are flipped horizontally and vertically), display systems should be adapted. Moreover, very large resolutions are required for integral images. Many improvements are proposed in the literature to solve the current limitations of holoscopic display systems like the limited depth of field, limited range of viewing angles, or the conversion from pseudoscopic to orthoscopic images [49][50]. Among all the display systems cited in [48], only integral imaging systems are able to display content with full parallax.

One of the main target display systems for VR applications are Head Mounted Displays (HMD). Examples of HMD are already available on the consumer market, such as Samsung Gear VR [12] or Oculus Rift [13], illustrated in Figure 2.9. Other HMD systems are proposed by major companies (e.g. Google Cardboard, Zeiss, Razer, Intel, Canon, Sony). A connection can be discussed with systems dedicated to Augmented Reality (e.g. Microsoft HoloLens) as these systems can display virtual 3D content that is mixed with the reality perceived through the glasses. The main limitations of HMD systems are: the resolution, as the input content generally consists in extremely large frames as illustrated in Fig. 2.5 (Sec. 2.2); the frame rate, that should target 120 frames per second; the degree of freedom (DoF), with 6 degrees of freedom being the optimal case where the user can change the position and the angle of views across all the axis; and the field of view (FoV), that depends on the target application. Systems like Google Cardboard or Samsung Gear are dedicated to mobile devices and can make use of a smartphone as a screen. Therefore improvements are to come with increased resolutions (e.g. 8K or more) for mobile devices.

Another kind of system is presented in [48] and referred to as *all-around*. These display systems are presented in a table-top configuration, hence they offer a viewing angle of 360 degrees to the user who can walk/turn around the device. Some systems use a rotating mirror and/or a rotating holographic diffuser (e.g. the light-field 3D display developed by USC [51], or Holo Table developed by Holy Mine [52]) while others are based on parallax barrier (SeeLinder developed by Nagoya University [53]).

2.4 Processing tools

In this section, we describe some of the main processing tools that are used for light-field content. First in Sec. 2.4.1, we describe state-of-the-art methods used to extract viewpoint images from integral images. Secondly in Sec. 2.4.2 and Sec. 2.4.3, the depth estimation

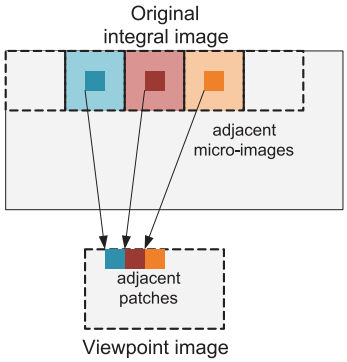


Figure 2.10: View extraction process.

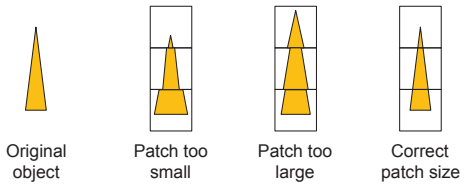


Figure 2.11: Patch size in view extraction

and view synthesis processes related to the MVD (Multi-View plus Depth) format are discussed.

2.4.1 View extraction from integral images

Several methods to extract viewpoint images (or views) from an integral image are described in [54]. Most of the methods extract one patch (a square zone of pixels) from each MI, as illustrated in Figure 2.10. This process is based on the characteristics of the focused plenoptic camera [44] for which there are both angular and spatial information within one MI. The angle of view depends on the relative position of the patch within the MI. A basic version of the method consists in using a patch of size 1×1, i.e. one pixel per MI. The size of the patch defines the depth plane in the scene on which the extracted view will be focused: the larger the patch, the closer the focus plane. The objects that are further or closer will present the following artifacts, illustrated in Fig. 2.11. If the patch is too large (i.e. the object is too far), then redundant parts of the object will be represented in several adjacent patches. If the patch is not large enough (i.e. the object is too close), then parts of the object will not be represented (pixelation).

A more advanced method allows reducing block artifacts by smoothing the transitions between adjacent patches. Pixels outside the borders of the patches are blended by a weighted averaging (pixels that are further from the center have a smaller weight, as illustrated in Fig. 2.12). It also blurs the parts of the picture that are out of the depth plane that is in focus (closer or further), which eliminates the above-mentioned mismatch artifacts and provides the same effect as in a common 2D photograph with limited depth of field.

A disparity estimation method is proposed in [44] in order to obtain the relative depth of the objects inside each MI. It is based on a block matching algorithm (illustrated in Fig. 2.13) with the following steps. A square patch P is first selected in the center of the

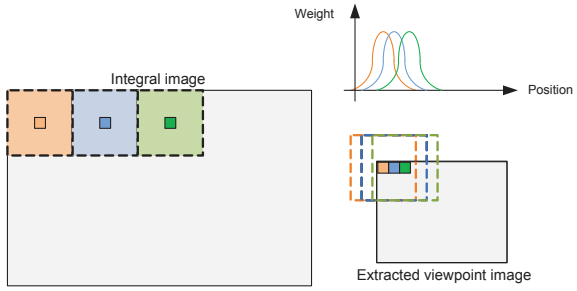


Figure 2.12: Blending in view extraction

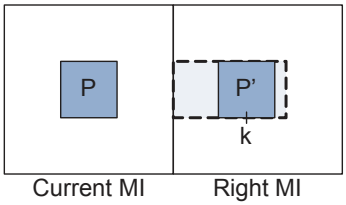


Figure 2.13: Block matching algorithm to estimate disparity between MIs

current MI with coordinates (Px, Py), and a second patch P' is selected in a neighbor MI (e.g. right and/or bottom) with coordinates $P'x = Px + S + k$, with S the size of the MI and k the disparity between the two MIs. The similitude between P and P' is computed (e.g. using normalized cross correlation) for values of k from 1 up to the maximum disparity value possible. The value of k providing the maximum similarities between P and P' corresponds to the disparity value of the current MI. This value in number of pixels corresponds to the adequate patch size to be used for the view extraction. Viewpoint images resulting from a disparity-assisted patch blending extraction (DAPBe [54]) are full-focused, as each patch size is adapted to the depth of the objects.

2.4.2 Depth map estimation

As mentioned in Chapter 1, the MVD (Multi-View plus Depth) format consists of texture videos, which are actual views of the scene from a given angle, and depth maps, which are gray level images providing information about the depth in the scene (i.e. pixels are darker or brighter depending on the distance from the camera). In case of Computer Generated content, depth maps can be generated automatically from the so-called z information of the 3D structure data. These are therefore ground truth depth maps containing reliable information for each pixel.

For natural scenes, depth maps can be captured using infrared cameras or time-of-flight cameras. A well known example of depth acquisition device is the Kinect [55] camera from Microsoft. Additionally to the gaming application that is its main commercial purpose, it has been widely used in the scientific literature when 3D information is needed in imaging applications for example. The resulting depth images present several limitations. The resolution is generally low and there are many artefacts due to the lack of precision of the cameras, especially on edges. Finally, the range of depth that can be acquired is limited to short distances.

Finally, depth maps can also be estimated from the views. The process to obtain depth values for each pixel in camera coordinates relies on disparity estimation techniques [1].

The principle is to estimate the spatial displacement (in pixels) between two (or more [56]) images that is due to the acquisition from different angles of view.

To resolve this problem, also referred to as stereo-matching problem, several techniques are proposed in the literature, mostly based on two approaches: block-based or pixel-based. Pixel-based algorithms are however preferred in most applications (e.g. encoding and synthesis) because the results are more accurate, around objects discontinuities in the pictures for example. In the pixel-based approach, local methods (i.e. the search is done only in a local window) perform well in textured areas and are convenient for real-time implementations [57][58]. However global methods have also been developed in order to avoid artefacts like noisy disparities in untextured areas and to overcome the issue of occluded areas in the pictures. Disparity estimation techniques are based on epipolar geometry [59]. A mapping between the 3D object points and its 2D projection onto the image plane is done with perspective projection. After an epipolar rectification, that is performed to simplify the problem by assuming a parallel camera arrangements, the search for corresponding points is performed in the resulting epipolar planes.

The Depth Estimation Reference Software [60] (DERS) is the most commonly used tool for compression and coding related applications, when depth maps are required in order to synthesize views that are not encoded. Specific improvements of this software have been recently proposed, for example by Poznan University, in studies of light-field technologies like Super Multi-View in the AHG FTV in MPEG in order to tackle the issues caused by non-linear camera arrays, especially for arc camera arrays [61].

2.4.3 View Synthesis

As mentioned in Chapter 1, it is possible to synthesize intermediate views from textures and depth maps. This technique is particularly useful in coding schemes. Indeed, it is possible to encode only a subset of the views of MVD content with associated depth maps, and to synthesize the views that were skipped at decoder side.

In this process of view synthesis called Depth Image Based Rendering (DIBR), the visual information taken from the available textures is projected or warped into the new position corresponding to the target view to synthesize. The new position (e.g. warping or projection distance) is provided per pixel by the depth information, and hole filling techniques (e.g. like inpainting [62]) are used for disoccluded areas (i.e. areas that are not available in the original texture views). The synthesis can also be improved by taking advantage of the temporal correlations in the intermediate views [63]. In coding applications, the View Synthesis Reference Software [64] (VSRS) is the main tool used for this purpose. Existing tools are efficient for linear camera arrays with horizontal disparity only (although in some cases, artefacts can limit the quality of the synthesized views, as described in Chapter 4). Like for DERS, specific improvements have been recently proposed in order to tackle the issues caused by non-linear camera arrays, especially for arc camera arrays [61].

2.5 Light-field content compression based on current encoders

2.5.1 Super Multi-View compression

Regarding Fig. 2.1, this section details the *Encoder* and *Decoder* blocks. Current encoders and their extensions (described in Chapter 1) can be used to encode light-field content [65]. For SMV content, MV-HEVC and 3D-HEVC require only slight syntax modifications [61], e.g. in the number of bits required to encode the view indexes that can be increased. In the first coding configuration considered, all the views are encoded. An example with an MV-HEVC based encoding is illustrated in Figure 2.14. A second configuration is considered where only a subset of the views is encoded as well as the associated depth maps, as illustrated in Figure 2.15. After decoding, the views that were skipped (not encoded) are synthesized (Figure 2.16). Although these coding configurations are technically manageable with current technologies, the results are expected to be sub-optimal as current standards do not take into account specific aspects of the light-field content such as the very large number of views (see Chap. 4), the two dimensional camera arrangements in the case of Full Parallax SMV (see Chap. 5), the non-linear camera arrangements like arcs (see Chap. 6), or application requirements like freedom of navigation between points of view (see Chap. 7). Improvements are therefore required. Several methods have been proposed in the scientific literature concerning the aforementioned aspects, that are further presented in the state-of-the-art sections of the dedicated chapters of this manuscript.

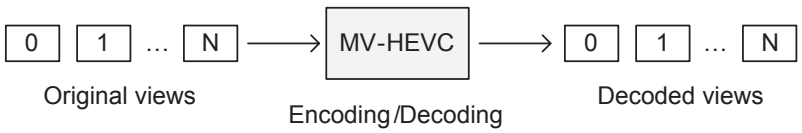


Figure 2.14: MV-HEVC encoding scheme (N views)

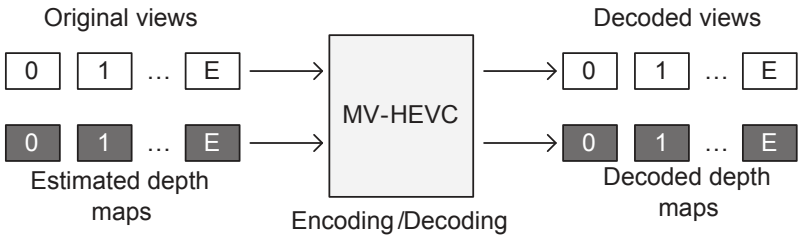


Figure 2.15: MV-HEVC encoding scheme (E views + E depth maps)

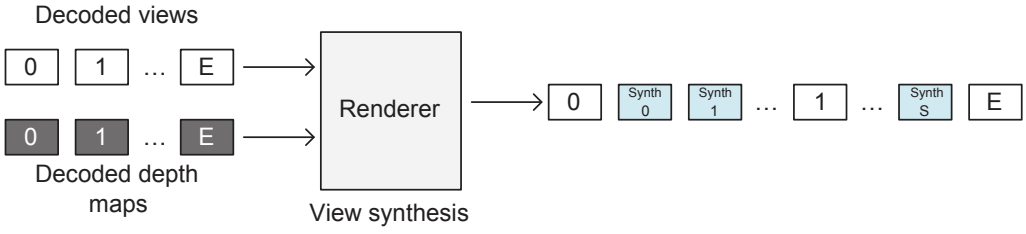


Figure 2.16: Rendering of S synthesized views from E views and associated depth maps

2.5.2 Integral images compression

Integral imaging content is represented as a 2D picture and can therefore be encoded using current 2D image and video compression techniques like JPEG or HEVC. Although JPEG has the advantage of being the most used and spread format for still images, coding efficiency is much higher with HEVC, as discussed in Chapter 3. Similarly to SMV, encoding integral images with existing technologies is therefore technically manageable, however the large resolutions and the Micro-Images are challenging for common encoders that are not adapted. Several methods for integral imaging compression have been presented in the scientific literature. A detailed review of these methods is provided in the state-of-art section in Chapter 3.

2.6 Conclusion

Target use cases and applications are tightly linked to the significant variety of capture and display devices and to the representation formats. As mentioned in the introduction Chapter, efficient representation of the light-field and efficient compression are key factors to trigger and enable the large development of light-field technologies for future immersive applications. In the following chapters we focus on the compression of Integral Imaging content and Super Multi-View content. We assess the performance with current state-of-the-art technologies and we propose improvements and new innovative schemes adapted to the specific characteristics of these contents.

Part II

Integral imaging

Chapter 3

Integral images compression scheme based on view extraction

3.1 Introduction

Integral (or plenoptic imaging) provides a dense sampling of the light-field, by capturing a large number of views within a narrow angle (as mentioned in Chapter 2). Plenoptic cameras use micro-lenses to capture light rays coming from several directions. Each micro-lens provides a micro-image (MI) in the resulting integral image. Integral images have a large resolution in order to provide a large number of viewpoint images with a sufficient resolution. Current resolutions are actually not even sufficient, and therefore larger sizes should be expected in the future for this kind of content. Moreover, the micro-images (MIs) based structure (grid-like) involves a large number of edges, which is challenging to encode. New efficient coding technologies are required to handle these characteristics. We propose an original compression scheme to encode integral images. It takes advantages of a view extraction process to reconstruct a reliable predictor and creates a residual integral image that is encoded. Although it offers some kind of display scalability, as a bitstream containing the extracted views is transmitted to the decoder, the first goal of the proposed scheme is compression efficiency.

In section 3.2, state-of-the-art methods to encode integral imaging content are presented. The proposed compression scheme is described in Section 3.3. The anchor and evaluation methods for compression performances are discussed in Section 3.4. In Section 3.5, we study the performance of the scheme with a single extracted view. As this scheme is highly parameterizable, we first propose several iterative methods to select the most efficient configuration, using a rate-distortion optimization (RDO) process to avoid exhaustive search methods. Additional runtime savings are then reported by exploring how the different parameters interact. In a second time, we assess the impact of the position and size of the patches used for the view extraction on the compression performance. In Section 3.6, we propose to improve the method with advanced filtering techniques. Wiener filter based methods are used to filter the view before the reconstruction step. The performance of the scheme using several extracted views is studied in Section 3.7. Finally in Section 3.8, the proposed scheme is combined and compared to relevant state-of-the-art methods. Perspectives and conclusions are drawn in Section 3.9 and Section 3.10 respectively.

3.2 State-of-the-art

In Chapter 2, we mention that encoding integral images with existing technologies is technically manageable, but that common encoders are not adapted to the specific aspect of this content, and expected to be inefficient. Improvements and new coding schemes have been proposed in the scientific literature.

A natural approach consists in applying the Discrete Cosine Transform (DCT) to the micro-images, followed by quantization and lossless coding. A differential coding between MIs can be used [66]. The differential coding can also be used for video sequences in order to remove the temporal correlations [67][68]. Inter-MIs correlation can be removed using the 3D-DCT on stacked MIs. Several scanning orders are tested in order to create the MIs 3D structure. An optimization of the quantization step (for 3D-DCT based compression algorithms) is proposed in [69]. This optimization is done by generating a matrix of quantization coefficients which depends on the content of the image. In [70], an hybrid 4-dimensional transform based on DWT and DCT is described (4D hybrid DWT-DCT coding scheme). The 2D DWT is applied to the MIs, followed by a 2D DCT applied to the resulting blocks of coefficients. In [71], the integral image is decomposed in viewpoint images. A 2D transform is performed by using 1D transforms on the lines and rows of the viewpoint images, resulting in 4 frequency sub-bands. The lower band is a coarse approximation of the original viewpoint image. The 2D transform is applied recursively to increase the level of decomposition at a coarser scale. The sub-bands are then grouped in $8 \times 8 \times 8$ elements volumes and processed by a 3D-DCT. As in the previous methods, the coefficient are then quantized and arithmetically coded. In [72], the transform is combined with a Principal Component Analysis (PCA, also called Karhunen-Loeve Transform or Hotelling Transform). DWT is applied to viewpoint images, and then PCA is applied to the resulting coefficients. Several kinds of DWT filters are proposed (e.g. Dauchechies wavelets). In [73], the SPIHT (Set Partitioning In Hierarchical Trees) method allows to display/transmit progressively the integral image as a quality scalable bitstream. Two algorithms (2D and 3D) are proposed. The first one is a 2D-DWT applied to the integral image and followed by the 2D-SPIHT. The second is based on the creation of a volume of viewpoint images on which a 3D-DWT is applied and followed by 3D-SPIHT.

Another approach consists in encoding the viewpoint images or the MIs of a still integral image as if they were a video sequence (called Pseudo Video Sequence or PVS) and then exploiting the temporal prediction tools of traditional video coders [74][75]. The method proposed in [76] exploits the inter-MIs redundancies (using the optical characteristic that MIs have overlapping zones). In [77], a KLT is applied to the viewpoint images. The viewpoint images can be encoded as a multi-view sequence (using inter-view prediction). In [78] and [79], the viewpoint images are encoded using MVC encoder [19]. The exploitation of temporal correlation and inter-view correlations induces an increase in complexity. The Evolutionary Strategy (ES) proposed in [80] is based on the evolution theory and allows an optimization of the coding scheme. In [81], ES is also applied and combined to a half-pixel precision for the motion/disparity estimation and compensation.

The Self-Similarity (SS) method exploits the non-local spatial correlation between MIs. The algorithm is mainly the same as for the inter prediction modes (of H.264/AVC [18] and HEVC [17]) but within one frame. A block matching algorithm is used to find a block similar to the current block in the causal zone in the current frame (which corresponds to the blocks that have already been coded and reconstructed). This similar block is then used as a predictor in the same manner as for a temporal prediction. In [82] and [83],

the implementation in H.264/AVC of the INTRA_SS (for INTRA Self Similarity) modes is described. These publications show the BD-rate gain brought by the SS mode and also the interest of a precise partitioning in macro-blocks. In [84], the SS mode is implemented in HEVC and the interest of the CU partitioning is shown. [85] shows the BD-rate gain brought by this method for video sequences (i.e. with a competition with inter-prediction modes). In [86], a Locally Linear Embedded-based prediction method (LLE) is proposed. The principle is similar to template matching. A search is performed in the causal zone in order to find a predictor block that has the neighboring pixels that match the best with the neighboring pixels of the block to predict. The same search can be performed at the decoder side, so that the amount of prediction information to transmit is reduced.

In [87] a scalable coding scheme is described as follows: the layer 0 corresponds to an extracted view, the layer 1 corresponds to a set of extracted views and the layer 2 is the integral image. Layers 0 and 1 are encoded respectively with reference HEVC and MV-HEVC encoders. For layer 2 the self-similarity method is used. An inter-layer prediction is also proposed, in which a sparse integral image is reconstructed from the set of views of the layer 1 and is inserted in the reference picture buffer to encode layer 2.

In response to the recent ICME 2016 Grand Challenge on Light-Field image compression [46], the aforementioned methods (i.e. SS [88][89], LLE [90], PVS [91][92]) have been proposed to overcome JPEG performances on plenoptic image compression. The wavelet-based and 3D-DCT based methods are conceptually far from the hybrid video encoder schemes (H.264/AVC and HEVC) and more adequate for still image coding than for video, whereas the self-similarity and multi-view methods are more easily included in the structure of these reference video encoders. The layered scheme offers an interesting display scalable feature but requires an additional increase in bitrate. In this chapter, we propose a new efficient coding scheme for integral images. As mentioned in Sec. 3.1, although it provides some level of display scalability, it targets compression efficiency. Additionally, this scheme is not restricted to still images.

3.3 Proposed scheme

In this section, the proposed compression scheme (Figure 3.1 and Figure 3.2) is described. In this scheme, a residual integral image II_R is encoded with HEVC. This corresponds to the *residual stream* in Figure 3.1. II_R is the difference between the original image II and a reconstructed image II^*. II^* is reconstructed from viewpoint images extracted from the original integral image II. Extracted views are encoded with 3D-HEVC. This is the *views stream* in Figure 3.1. The number of views is not limited. Due to their small resolution, views represent a small number of bits to encode compared to II. Moreover, they have a *natural* image aspect that is less costly to encode than the MI based structure of II. To obtain views with such a smooth aspect, advanced extraction methods are used, which use blending and varying size patches (see Chapter 2), both however preventing from perfect reconstruction with the exact original pixel values. The corresponding missing information, the difference between II and II^*, is recovered in II_R. By definition, for a reconstructed image II^* close to the original II, the subtraction is expected to provide absolute values close to zero. Therefore, under this condition, II_R has a *flat* aspect with low variations, which is easier to encode with HEVC than II. In practice, II_R can have a noisy aspect because of reconstruction errors, as discussed and illustrated in Section 3.5.1.4.

During the reconstruction performed in this scheme, the patches in the viewpoint image(s) are copied to their original position within the MIs in the reconstructed image

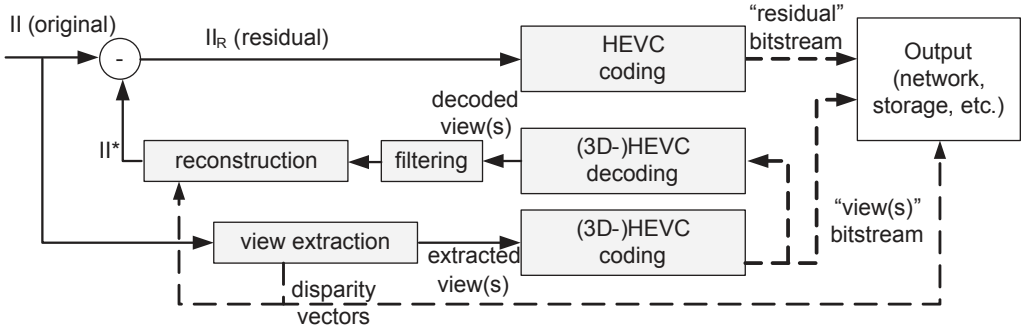


Figure 3.1: Proposed scheme - *encoder side*

II*, as illustrated by the *left* and *central* part of Figure 3.3 (note that for clarity purpose Fig. 3.3 represents content with horizontal parallax only). With this first step, the MIs in II* are partly filled, i.e. only pixels corresponding to the patch are filled, the rest of the MI is empty. Surrounding pixels in the view contained in a zone of the same size as the MI (illustrated by the dotted rectangles in the *right* part of Fig. 3.3) are also copied to fill the MIs in II*. These two steps are similar to the sparse reconstruction step and the micro-image refilling step described in [93]. Therefore, when reconstructing II* from the extracted view(s), some missing pixels, coming from different angles of view, are replaced by adjacent pixels from the same view (as shown in Fig. 3.3 with one view). However, the transformation of an object when changing the angle of view is not limited to a simple translation (disparity) but also involves angular differences. Hence errors are introduced. A low-pass filtering (e.g. average filter) is applied on the decoded views before the reconstruction to help smoothing these errors. High frequencies in the views are filtered while preserving the shape of the objects.

Disparity values computed at the extraction step are necessary for the reconstruction, therefore they must be transmitted to the decoder, among with the view(s) and the residual image $\mathrm{II_R}$. At the decoder side (Figure 3.2), the views are decoded and used to reconstruct II*, and $\mathrm{II_R}$ is decoded and added to II* to obtain the output decoded image.

There is a tradeoff between rate and quality of the views and the rate of $\mathrm{II_R}$. II* must be as close as possible to II in order to minimize the cost of $\mathrm{II_R}$, without increasing too much the cost of the views. Several combinations are possible for the following parameters: the Quantization Parameter (QP) used to encode the views ($\mathrm{QP_V}$), the QP used to encode the residual image ($\mathrm{QP_R}$), and the size M×M (in pixels) of the average filter applied to the decoded view (M). As in practice most of the bitrate is dedicated to $\mathrm{II_R}$, the value of the parameter $\mathrm{QP_R}$ is set according to the target bitrate (or quality), and $\mathrm{QP_V}$ and M are considered as parameters to optimize for a given $\mathrm{QP_R}$. The number of extracted views, their positions (i.e. angle of view) and the size of the patches used at the extraction step also have an impact on the performance. In the following we explore methods to tune these different parameters of the scheme and balance this tradeoff.

3.4 Anchor selection and performance evaluation method

As mentioned in Chapter 2, integral images represent the light-field as a 2D image composed of micro-images, and therefore they can be encoded using current compression standards and associated encoders. Most of the time, HEVC Intra is used as an anchor to

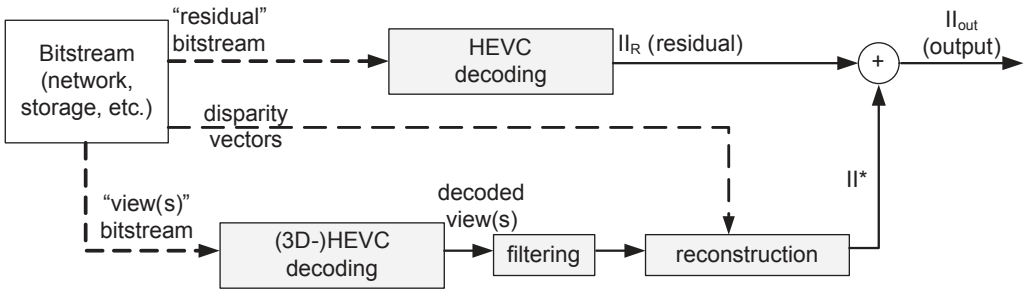


Figure 3.2: Proposed scheme - *decoder side*

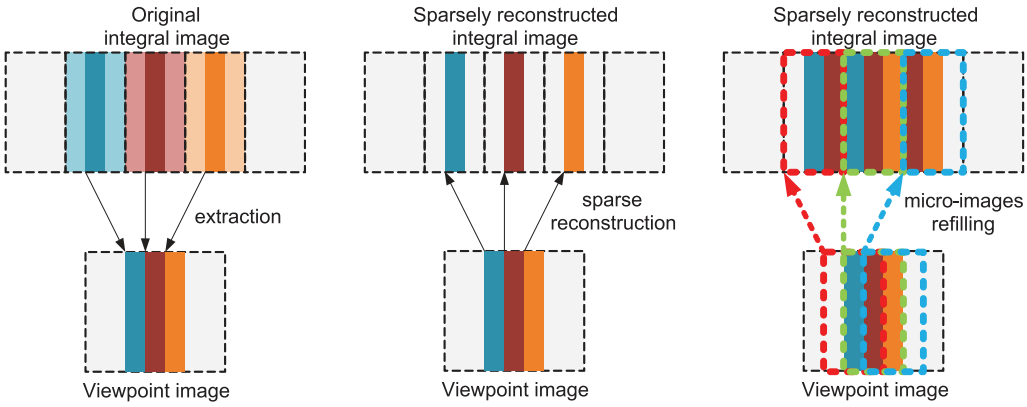


Figure 3.3: Reconstruction process

(a) Grid on (b) Grid removed

Figure 3.4: Grid/border pixels removed (*Fredo*)

	HEVC vs. JPEG	
	high bitrate	low bitrate
Fountain	-39%	-46%
Fredo	-51%	-58%
Jeff	-44%	-51%
Laura	-35%	-40%
Seagull	-45%	-54%
Sergio	-40%	-47%
Zenhgyun1	-49%	-57%
Average	**-43%**	**-51%**

Table 3.1: BD-Rate results for HEVC vs. JPEG. Negative values are gains for HEVC

compare the latest methods proposed in the literature (see Sec. 3.2). As mentioned above, in the recent ICME 2016 Grand Challenge on Light-Field image compression [46], JPEG is also used as an anchor. Although the performances of HEVC are known to be much higher on natural images, using JPEG as an anchor also makes sense as it is by far the most widespread and used standard for most of the still images produced today. In this section we compare the compression efficiency of these two encoders on integral images. The experimental conditions are as follows.

Seven still images [45] (listed in Table 3.1) are used in our experiments. Images were cropped to remove incomplete MIs and cleaned from grid pixels corresponding to the boundaries of the micro-lenses [94]. This grid removal process is automatically performed, i.e. regularly spaced bands of pixels are removed, as illustrated in Figure 3.4, by a method that takes as parameters the resolution of the picture, the resolution of the micro-lens array (i.e. number of MIs) and the width of the grid to remove. Therefore the inverse operation can easily be performed (i.e. adding grid pixels), for example after decoding, in a case where the target display takes as input a picture that includes the grid. HEVC reference software (HM14.0) with the *Intra main* configuration [95] is used on the QP range {20,25,30,35} for higher bitrates and {25,30,35,40} for lower bitrates. For JPEG, the FFMPEG implementation [96] is used on the Quality Factor range {0,4,7,14} for higher bitrates and {4,7,14,27} for lower bitrates, in order to provide similar PSNR target values for both encoders. Compression results in Table 3.1 are provided using the Bjøntegaard Delta (BD) rate metric [97]. Negative values represent gains of HEVC over JPEG. The performance for HEVC is much higher than for JPEG, with average BD-rate gains of 43% and 51% reported on the higher and lower tested quality ranges respectively, with gains up to 58% (for *Fredo* in lower bitrates).

These results are obtained by computing PSNR values on all the pixels of the decoded (i.e. output) integral image against the original integral image. Another evaluation procedure for the quality of integral images has been proposed in [98]. The quality corresponds to the average of the PSNR values computed on views that are extracted at several positions (multiple perspectives case) and/or at several focal distances (multiple focal points case). It is straightforward to understand that it makes sense to evaluate the quality on natural images that can be directly visualized by a user (i.e. extracted views). However this statement stands for subjective quality evaluation, but is questionable for objective metrics. Indeed, the selection of the perspectives and focal points is arbitrary and could bias the results for other perspectives and other focal points. For example, a scheme that would encode only the views that are used for the metric would provide a much better

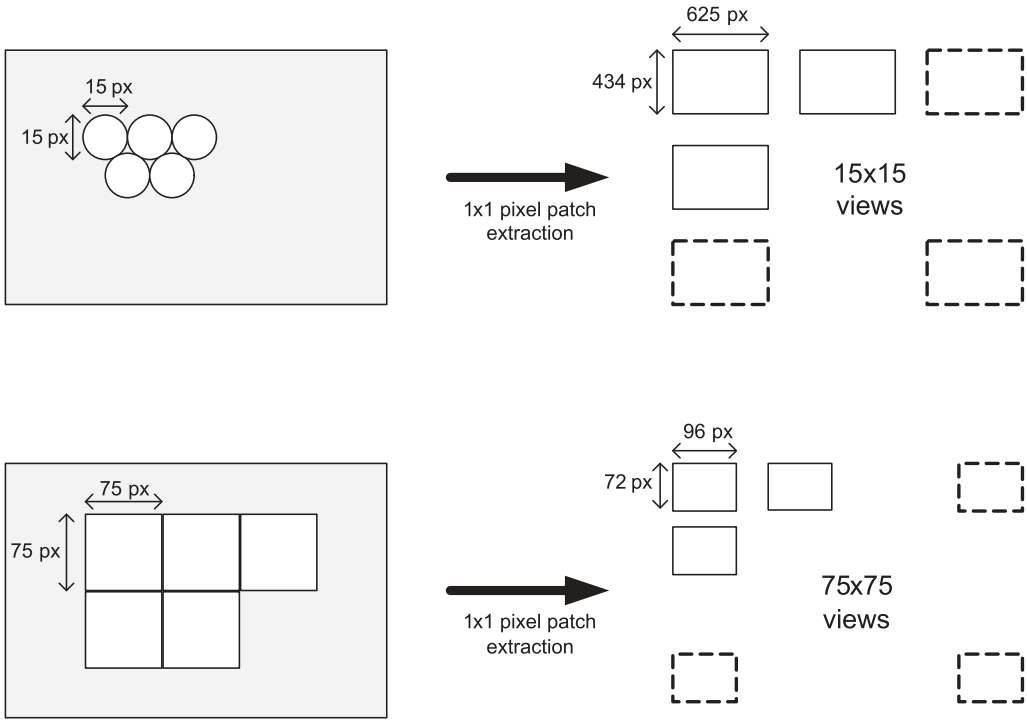


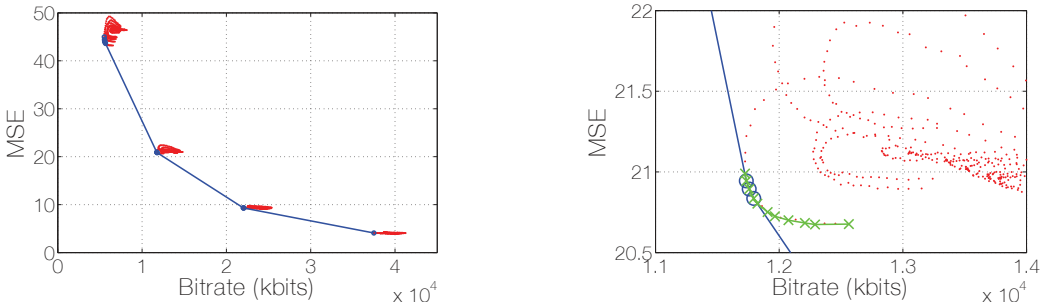
Figure 3.5: Extraction with one pixel per MI: (top) Lytro Illum content (bottom) Georgiev content

compression performance, but a large part of the content would be missing. In a less extreme case, a scheme that would be tuned or optimized for these particular views would provide good results according to this metric, while the part of the content that is not evaluated could have virtually any level of distortion. A similar approach is used in [99] in order to provide a more in-depth study of the impact of compression on integral images depending on the target application.

In the requirements of the aforementioned ICME 2016 Grand Challenge on Light-Field image compression [46], integral images captured by a Lytro Illum [10] camera are used as a test set. The raw integral images (i.e. as taken directly from the sensor of the Lytro Illum camera) are color corrected with a demosaicing step and converted to the YUV 4:4:4 format, then the chroma components are downsampled to YUV 4:2:0. The data in this format are finally reorganized as a 4D structure consisting of 15×15 views, corresponding to views extracted by taking one pixel (i.e. one patch of size 1×1 pixel) from each MI. PSNR is computed on those extracted views before and after compression.

As illustrated in Figure 3.5, in the case of the Grand Challenge test set, the restructured views have a low but acceptable resolution because there is a large number of small MIs. However, in the Georgiev test set, there is a smaller number of significantly larger MIs, therefore views extracted with 1 pixel per MI are too small to have a natural image aspect. Because all the views are taken into account in the PSNR computation in the Grand Challenge requirements, all the pixels of the integral image are evaluated. Hence this is nearly equivalent to the method that is used to obtain the results reported in Table 3.1, i.e. the distortion is measured on all the pixels. The same data is evaluated, only structured differently, which has no significant impact on the results.

In the following of this chapter, HEVC is therefore used as an anchor and the quality



(a) All configurations (b) Close up on QP_R 20

Figure 3.6: Rate-distortion points for all configurations (*Fountain*)

of the integral images is measured on all the pixels for the compression performance evaluation. Additional comparison with JPEG as an anchor are given in Section 3.8 in order to present the results as for the ICME Grand Challenge (although with a different test set).

3.5 Proposed methods with one extracted view

In this section, we study the performance of the scheme using a single extracted view to reconstruct II^*. We first propose several methods to tune the values of QP_V and M for a given QP_R, trading-off rate-distortion performance and complexity. In a second time, we also assess the impact of the position and size of the extracted patches on the compression performance.

3.5.1 Iterative methods to tune the scheme

We first propose to perform an exhaustive search, by testing a large number of combinations of values for QP_R, QP_V and M in predefined intervals, and to select the combinations that provide the best compression performance (using the Bjøntegaard Delta (BD) rate metric [97]), in order to obtain an optimal result. Results provided by this preliminary study are used in Section 3.5.1.1 to determine a rate-distortion optimization (RDO) process that allows selecting the best QP_V and M values for a given QP_R. Several iterative methods based on this criterion are proposed in Section 3.5.1.2. Experimental conditions and results are provided in Section 3.5.1.3 and Section 3.5.1.4 respectively.

3.5.1.1 Determination of the rate-distortion criterion

Figure 3.6 illustrates the RD values (red dots) provided by the exhaustive search for the best combinations for QP_R, QP_V and M (among hundreds of combinations tested). We define the global convex hull (GCH, illustrated in blue) as the convex hull of all points, and the local convex hull (LCH, illustrated in green) as the convex hull of a set of points with a same QP_R value. For a given QP_R, optimal configurations are represented by the set of points located at the intersection S of LCH and GCH. From our experimental data, it can be observed that this intersection is not empty (i.e. for each QP_R, there is at least one point that belongs to GCH).

Interval of QP_R	*Fountain*	**average**
25-20	3734	3836
20-15	1128	1140
15-10	337	339

Table 3.2: Experimental λ values ($\times 10^{-6}$)

Name	Criterion QP_V	Criterion M	Iterations on M
method_1	RDO		all
method_2.1	RDO		first QP_V iteration
method_2.2	RDO	MSE	first QP_V iteration
method_3.1	fixed	MSE	single QP_V iteration
method_3.2	fixed		none

Table 3.3: Summary of the iterative methods to tune QP_V and M

Figure 3.6 shows that using only the LCH provides sub-optimal configurations, as illustrated by the points marked by a green cross that are located on the right side of GCH. However, GCH cannot be plotted without encoding the image with several QP_R values, which multiplies the number of tested combinations. The idea in this section is to be able to select the configuration (for a given QP_R) that provides rate and distortion values (R and D respectively) minimizing a cost $D + \lambda R$, where λ is the slope of LCH in S (hence of GCH). In Figure 3.6, this is equivalent to find among the points marked by a cross the points that are also marked by a circle.

For each of the three segments of the blue curve in Figure 3.6, which connect two points with a different QP_R, the slope $\frac{\Delta D}{\Delta R}$, corresponding to a value of λ is calculated. Table 3.2 shows experimental values obtained in average on the seven images of our test set, and the values obtained for one image (*Fountain*) to illustrate the stability of the result from one image to another.

The slope of GCH between two consecutive values of QP_R increases exponentially according to QP_R. Hence an estimation of $\lambda = f(QP_R)$ is possible. By linear regression using the least square method, we obtain the function defined in Equation 3.1, with $a = 0.34$ and $b = -15.8$, which has an excellent fit with the data.

$$\lambda = 2^{a\mathrm{QP_R}+\mathrm{b}} \tag{3.1}$$

3.5.1.2 Description of the methods

Table 3.3 summarizes the methods proposed in the following. A first iterative method is proposed, *method_1*, where combinations of QP_V and M are successively processed for a given QP_R. The combination that provides rate and distortion values minimizing a cost $D + \lambda R$ (with D the distortion, R the bitrate, and λ a lagrangian multiplier as determined in Section 3.5.1.1) is selected. Iterations on QP_V and M induce a large total encoding runtime, therefore two variant methods are proposed in the following in order to reduce the number of iterations.

In *method_2.1* and *method_2.2*, the iterations on M are processed only for one QP_V value (e.g. for the first one, $QP_V = 10$, in our experiment) and the best M value is kept for the remaining QP_V iterations, in order to reduce the number of tested combinations. The best value is the one that minimizes the cost $D + \lambda R$ in *method_2.1* (same RDO process as for *method_1*), and the one that minimizes the mean square error (MSE) of II* against II in *method_2.2*.

Finally, we define two more methods, *method_3.1* and *method_3.2*, where a value of QP_V is empirically set and associated to a value of QP_R. In *method_3.1*, M is iteratively selected according to the MSE of II* against II (as for *method_2.2*), while in *method_3.2*, M is also fixed.

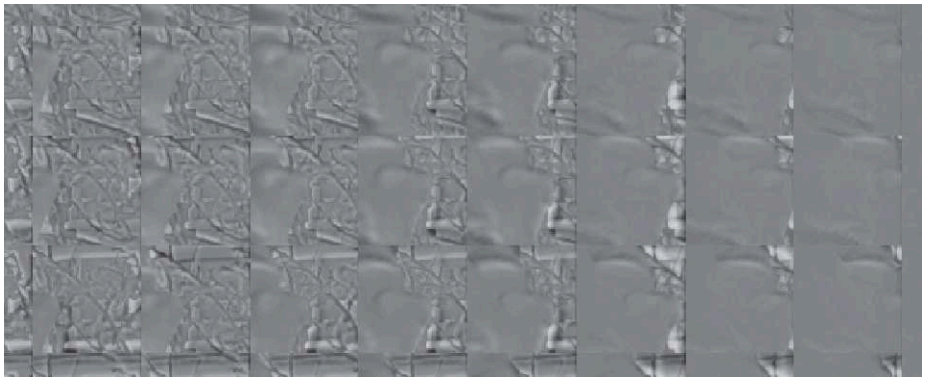
3.5.1.3 Experimental conditions

The disparity-assisted patch blending method [54] is used for the extraction of one single view. The residual image is encoded with a bit depth of 10 bits, as it can take values in the range [-255;+255]. Encodings of the view and the residual image are performed under HEVC reference software (HM14.0) using the *Intra main* configuration [95], and disparity values are coded with 4 bits per MI (1 value per MI in the range $\{1,\ldots,15\}$). For each target QP_R in the range {10,15,20,25}, combinations of values for the parameters QP_V and M (in the ranges $\{10,11,\ldots,50\}$ and $\{1,2,\ldots,11\}$ respectively) are tested, providing 1804 ($4 \times 41 \times 11$) possible rate-distortion (RD) points. Compression results are provided using the Bjøntegaard Delta (BD) rate metric [97]. II encoded with HEVC on the QP range {25,30,35,40} is the anchor, and negative values represent improvement over the anchor.

3.5.1.4 Experimental results

3.5.1.4.1 Exhaustive search for optimal configuration: for each image, Table 3.4 shows the configuration that provides the best BD-rate results. An average BD-rate gain of 15.7% (up to 31.3% for *Fredo*) is reported when using optimal parameter combinations. QP_V values increase according to QP_R, providing a tradeoff between the bitrate for the views and for II_R. Optimal values for QP_V and M depend on the tested image. Approximately 97% of the total bitrate is dedicated to II_R in average, mainly because of its very large resolution compared to the view (e.g. for *Fountain* 6512×4880 against 960×720), which represents the remaining 3% (disparity values used for extraction and reconstruction cost only 0.3%). The aspect of the residual image II_R is illustrated in Figure 3.7, showing regions that are well reconstructed like the face of the character (i.e. where II* is close to II); and region that are not correctly reconstructed like the background, where the artifacts in II* (i.e. differences with II) appear in the residual image. Figure 3.8 plots the PSNR against bitrate curves obtained for *Fountain* with the anchor method and with the proposed scheme. These curves show the compression gain provided by the proposed scheme between 30dB and 40dB, which are commonly considered as corresponding to acceptable to good qualities for 2D images. It can be observed that the gain increases as the bitrate decreases.

3.5.1.4.2 Rate distortion optimization: Table 3.5 shows the BD-rate and coding time variations with *method_1* (for each image and in average) in reference to the anchor. Combinations selected by *method_1* are very close to the best configurations determined by exhaustive search (same M values and only slight differences for a few QP_V values),



(a) Residual image II_R

(b) Corresponding original image

Figure 3.7: Aspect of the residual image (*Laura*)

Image	BD-rate (%)	Param. for each QP_R in {10,15,20,25}	
		QP_V	M
Fountain	-17.0	19 21 23 29	3 3 3 3
Fredo	-31.3	18 21 25 32	3 3 3 3
Jeff	-5.9	25 30 30 32	9 9 9 7
Laura	-11.2	22 25 27 31	4 4 4 4
Seagull	-13.7	20 21 25 29	3 3 3 3
Sergio	-23.6	19 19 24 32	4 2 2 2
Zenhgyun1	-7.5	25 26 30 32	9 9 9 7
Average	-15.7		

Table 3.4: BD-Rate results with best configurations QP_V and M for each QP_R. Negative values are gains over the anchor

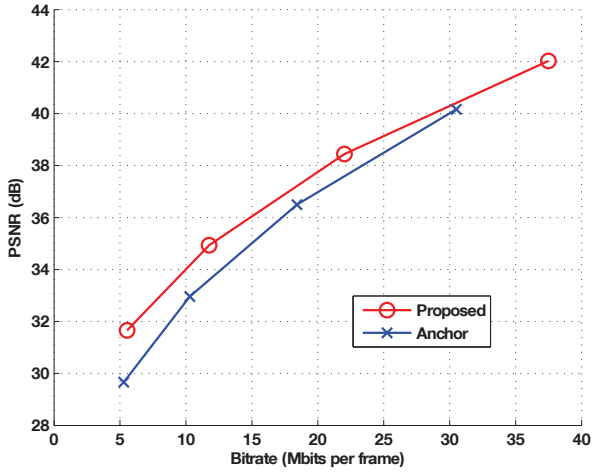


Figure 3.8: PSNR against bitrate - *Fountain*

and average BD-rate gains of 15.7% are preserved, which shows the robustness of the estimation of $\lambda = f(\mathrm{QP_R})$. The total encoding runtime for all the iterations is large, with a multiplication of the anchor encoding time by 484 in average. It should be noted that the ranges of tested values for $\mathrm{QP_V}$ and M are not fully used and can be tightened to decrease the number of iterations.

Table 3.6 shows the average BD-rate results and average coding time variations for all the proposed methods. Decoding time does not depend on the number of iterations performed at the encoder and is therefore specifically discussed in Section 3.5.1.5. For *method_2.1*, the total encoding runtime is significantly reduced (down to 55 times the anchor) because the number of iterations is reduced to 51 (instead of 451 with *method_1*). BD-rate gains of 15.7% are preserved because M does not vary significantly according to $\mathrm{QP_V}$. Results for *method_2.2* show that the encoding runtime can be further reduced (down to 44 times) by selecting M without encoding the residual image for each iteration, with an average BD-rate gain almost as good (15.3% in average, e.g. with a decrease of 1.7% for *Seagull*, and 0.8% for *Sergio*). This shows that the MSE of II^* against II is a good indicator of the encoding performance, as for a reconstructed image close to the original image, the residual image is easier to encode. It should be noted that the number of iterations on $\mathrm{QP_V}$ can be further reduced by avoiding the full search on the range {10,11,...,50}. For example, it can generally be observed that the cost $D + \lambda R$ has one local minimum according to $\mathrm{QP_V}$, for M and $\mathrm{QP_R}$ given (as illustrated in Figure 3.9). Hence the iterations on $\mathrm{QP_V}$ can stop when the cost starts to increase.

3.5.1.4.3 Empirical selection of the parameters: The number of images in our test set is limited to only seven. Therefore, in order to provide fair results, we use a *leave-one-out cross-validation* method to empirically select the parameter values for *method_3.1* and *method_3.2*. For each tested image, parameters that provide the best average results on the six other images are selected for the experiment.

In Table 3.6, results for *method_3.1* show that assigning one $\mathrm{QP_V}$ to one $\mathrm{QP_R}$ largely reduces the encoding runtime (only 1.4 times the anchor) and still provides 15.5% BD-rate gains in average, which is close to optimal. Although the number of available images

Image	BD-rate (%)	Coding time (%)	Param. for each QP_R in {10,15,20,25}	
			QP_V	M
Fountain	-17.0	48284	19 21 23 27	3 3 3 3
Fredo	-31.1	47067	18 21 25 28	3 3 3 3
Jeff	-5.9	48729	25 30 30 32	9 9 9 7
Laura	-11.2	49065	22 25 27 30	4 4 4 4
Seagull	-13.7	48836	19 21 25 29	3 3 3 3
Sergio	-23.5	48036	20 21 24 28	4 2 2 2
Zenhgyun1	-7.5	48554	25 26 31 30	9 9 9 7
Average	-15.7	48367		

Table 3.5: BD-Rate results, coding time variations and associated configurations for method *method_1*

Method	BD-Rate (%)	Coding time (%)
method_1	-15.7	48367
method_2.1	-15.7	5526
method_2.2	-15.3	4443
method_3.1	-15.5	136
method_3.2	-8.5	120

Table 3.6: Average BD-Rate results and coding time variations

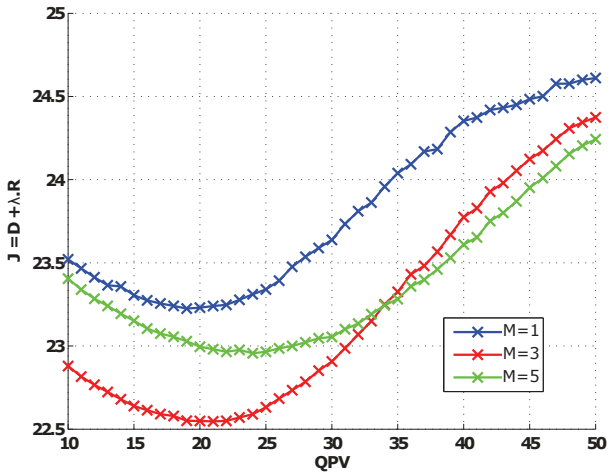


Figure 3.9: RD cost ($J = D + \lambda R$) according to QP_V (*Fountain* with $QP_R = 15$)

Runtime (%)	against anchor	Extr.	Rec.	HEVC View	HEVC II_R	Others
Encoding	130	7	8	2	79	4
Decoding	240	/	31	1	46	22

Table 3.7: *Fountain* - Runtime variation against anchor with *method_3.1*, and percentage of the total time for each task including: extraction, reconstruction, view and residual encoding/decoding, and blur, subtraction and sum as *others*

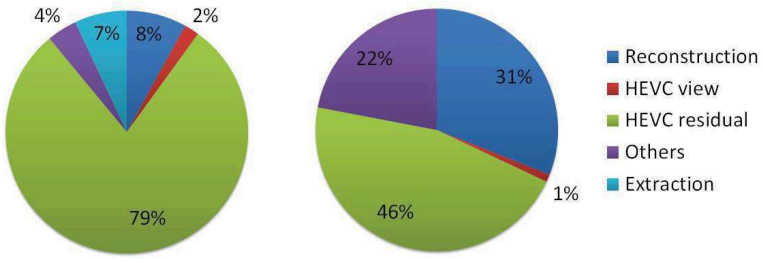


Figure 3.10: Percentage of the total time for each task, *left*) encoding, *right*) decoding

is limited, parameters only slightly differ from one image to another. This robustness suggests similar gains on other integral images. For *method_3.2*, the coding time is only 1.2 times the anchor time. However, the BD-rate gain drops to 8.5%, with losses for *Jeff* and *Zenhgyun1*. It shows that the adequate value for M strongly depends on the image and that iterations on this parameter can significantly improve the coding performance, with only a slight increase of the encoding runtime.

3.5.1.5 Runtime of the proposed scheme (*method_3.1*)

Table 3.7 shows the encoding and decoding runtime variations against the anchor for *Fountain* with *method_3.1*, and the percentage of the runtime dedicated to each task. The repartition of the time for each task is also illustrated in Figure 3.10. Runtime values have been obtained in the following conditions. Every task has been performed five times, maximum and minimum values have been discarded, and the average runtime values provided in the table are processed on the three remaining values (trimmed mean).

Encoding runtime of the proposed scheme is 1.3 times the encoding time of II with HEVC (anchor), with encoding of II_R representing 79% of the total time. The eleven iterations of blur, reconstruction, and subtraction steps represent 12%. View extraction represents 7%, mainly because of the time-consuming disparity estimation. Decoding runtime does not depend on the number of iterations at the encoder side. It is 2.4 times the anchor, with 46% for the decoding of II_R. Reconstruction (31%) and sum (22%) represent a larger percentage at the decoder because HEVC decoding process is much faster than encoding. The increase is larger in lower bitrates where HEVC decoding time is further reduced while the reconstruction and sum runtime do not vary.

3.5.2 Impact of the position and size of the extracted patch

In the experiments described in Section 3.5.1, the disparity-assisted patch blending method [54] is used for the extraction of the central view. Therefore, the center of the extracted patches are aligned on the center of the MIs, and the size of the patches varies for each MI depending on the estimated disparity. In this section, we study the impact of the position and the size of the extracted patch on the compression performance.

3.5.2.1 Position of the extracted patch/view

We compare the performance of the scheme with views extracted at different positions (i.e. extracted patch centered at different positions within the MI). Several positions for the center of the patch are tested with horizontal and vertical distances from the center of the MI (in pixels) in the range {-20,-15,-10,-5,0,5,10,15,20}.

For the view extraction step in our experiments, we have implemented the disparity-assisted patch blending method (described in Chapter 2) according to [54]. However, in this work, the author only focuses and performs experiments on the most central point of view. The method has to be adapted to the extraction of side views. First, for the disparity estimation step, the adaptation basically consists in shifting the position of the block used for the block matching algorithm, according to the target point of view. For the patch extraction step, pixels surrounding the borders of the patches are kept and blended with a weighted averaging, in order to avoid blocky artifacts in the view (see Chapter 2). As the patches have varying sizes (depending on the estimated disparity values), they are resized (with surrounding pixels in the MI) in order to match the maximum patch size. The distance from the patch to the center of the MI must be resized accordingly, so that the pixels used for the disparity estimation correspond to the pixels extracted in the patch, and to avoid a mismatch of angle of views between patches extracted from MIs with different disparity values.

Blending the pixels surrounding the border of the patches reduces blocking artifacts. However, for the side views that are far for the center, pixels close to the border of the MIs are included in the blending, introducing grid artifacts in the extracted views. Therefore, in this section, we also perform the experiments with views that are extracted with a smaller blending zone, presenting less artifacts as illustrated in Figure 3.11.

We have already shown in Section 3.5.1 that a single value of QP_V can be empirically associated to a value of QP_R, providing consistent gains when using a *leave-one-out cross-validation* method to select the experimental values (for *method_3.1*). Therefore, in this experiment, each tested QP_R value in the range {10,15,20,25} is associated to a QP_V value in the range {20,22,25,31}. M values providing the best results with the central view (e.g. with *method_3.1*) are kept for each tested image. The remaining of the test conditions is as described in Section 3.5.1.3.

Figure 3.12 plots the BD-rate gains over the HEVC anchor according to the position of the patch in the MI. Table 3.8 shows the average BD-rate results for the tested positions in the central row, column, and diagonal. Figure 3.12 and Table 3.8 illustrate that the compression performance generally increases when the extracted patch is closer to the center of the MI. Although views with a position shifted in the horizontal dimension can also provide good results (with results for position -5 and 5 equal or very close to the result at the center), the central view provides the better performance in average.

(a) central view, normal blending zone

(b) side view, normal blending zone

(c) central view, reduced blending zone

(d) side view, reduced blending zone

Figure 3.11: Close-up on views extracted at 2 different positions, with 2 different sizes of blending zone (image: *Fountain*)

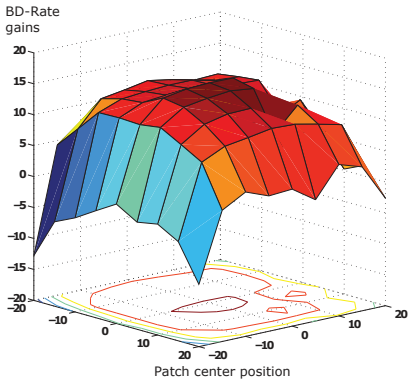


Figure 3.12: Average BD-rate gains according to the position of the extracted patch in the MI

Position	-20	-15	-10	-5	0	5	10	15	20
horizontal	-0.3	-12.6	-16.0	**-16.7**	**-16.8**	**-16.8**	-11.0	-9.5	-5.3
vertical	-10.2	-14.4	-14.6	-15.0	**-16.8**	**-16.5**	-14.6	-12.2	-3.2
diagonal	11.8	-9.1	-14.0	-14.8	**-16.8**	-16.0	-13.4	-10.4	1.6

Table 3.8: Average BD-rate results according to the position of the extracted patch in the central row, column, and diagonal

Image	9	10	11	12	15	varying (*method_3.1*)
Fountain	25.8	-3.2	129.6	124.3	79.5	-16.7
Fredo	49.0	-7.9	286.7	295.5	187.9	-31.2
Jeff	40.3	-3.4	185.2	190.5	147.1	-5.4
Laura	9.9	-3.2	79.3	74.4	50.0	-11.1
Seagull	3.8	0.7	131.9	132.3	82.2	-13.7
Sergio	31.0	-7.2	164.4	168.2	105.8	-23.3
Zenhgyun1	66.2	-6.3	293.8	301.8	190.7	-7.2
Average	32.3	-4.4	181.6	183.9	120.5	-15.5

Table 3.9: BD-Rate results with fixed patch size

3.5.2.2 Size of the extracted patch/view

We also assess the performance of the scheme with views extracted with a fixed patch size. Patch sizes in the range {10,11,12,15} are tested. These sizes correspond to the maximum values obtained on the images of our test set with the varying patch size extraction, i.e. with lower values patches are downsized and the visual data is altered. QP_V and QP_R values are the same as in the previous experiment. M values in the range {2,...,9} are tested. The remaining of the test conditions is as described in Section 3.5.1.3.

Table 3.9 shows the best results obtained with a fixed patch size. Patch sizes larger than 10 provide very large BD-rate losses from 120% to 180% in average, with the larger value always selected for M ($B = 9$). A patch size of 10 with $M = 4$ provides the best results of this experiment, with an average BD-rate gain of 4.4%. However, this best results is still far below (more than 10%) the results provided with a varying patch size. With a fixed patch size, artifacts occur on the pixels that are out-of-focus (i.e. that require a different patch size), as described in Chapter 2, and therefore degrade the quality of the reconstructed image.

3.6 Improvement of the filtering step

In the experiments described in Section 3.5, the views are filtered with an average filter before the reconstruction. Although this filtering step smooths the reconstruction errors and improves the quality of II* (when compared to II), the average filter is not the optimal filter. Moreover, iterations on the filter size must be performed to select a value that provides the best compression efficiency. In this section, we propose to compute the filter's coefficients using the Wiener filter technique [100].

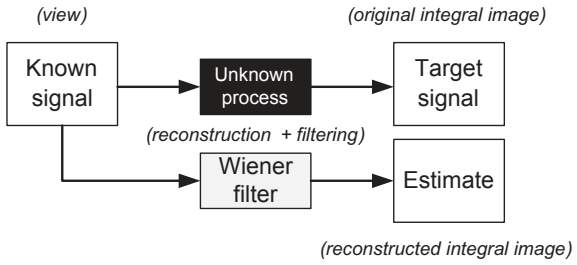


Figure 3.13: Illustration of the Wiener filter problem

3.6.1 Wiener Filter in integral image reconstruction

The Wiener filter is the solution to the problem illustrated in Figure 3.13. It is used to produce an estimate of a target signal by linear time-invariant filtering of a known signal, minimizing the mean square error between the estimated and the target signals. In the case of our compression scheme, the known signal to filter is the view, the original image II is the target, and the reconstructed image II* is the estimate of II.

The integral image reconstruction step is not a time-invariant process (e.g. shifting the view by one pixel before reconstruction is not equivalent to shifting II* by one pixel after reconstruction). However, the reconstruction step is locally time-invariant at the MIs level, therefore the coefficients computation and the filtering can be processed MI-wise, minimizing the mean square error between the MIs that are copied from the view to the reconstructed image II* and the original MIs at the same position in II.

One of the solutions to the problem illustrated in Figure 3.13 is the causal finite impulse response (FIR) Wiener filter. It can be written in the matrix form $Ta = v$ (known as the Wiener-Hopf equations), where T is a symmetric Toeplitz matrix populated with estimates of the auto-correlation of the input signal (i.e. MIs taken from the view), v is a vector populated with estimates of the cross-correlation between the input and output signals (i.e. MIs taken from the view and from II), and a is a vector populated with the coefficients of the filter.

The Wiener-Hopf equations are computed and solved at the encoder side where all the images are available, and the resulting Wiener filter coefficients are coded in the bitstream to be read by the decoder.

3.6.2 Proposed Wiener filter based methods

We first propose to compute a single set of coefficient for the entire image (i.e. all the MIs). This first method is mentioned as the Wiener filter based method in the following. In a second time, we propose to further improve the Wiener filter based method by adapting the filter according to the disparity. In the disparity-assisted patch blending method [54] used in our experiments, patches are resized according to the disparity of the MIs to fit in the extracted view. Therefore, one filter per disparity value existing in the image is processed in this second method, referred to as disparity-adaptive Wiener filter based method in the following.

3.6.3 Experimental results

As shown in Section 3.5, a single value of $\mathrm{QP_R}$ can be empirically associated to a value of $\mathrm{QP_R}$, hence for this experiment, each tested $\mathrm{QP_R}$ value in the range {10,15,20,25} is

Image	BD-rate (%)	Comparison with blur based method	
		BD-rate (%)	Δ PSNR (dB) II* vs. II
Fountain	-15.1	1.8	0.31
Fredo	-27.7	4.4	0.04
Jeff	-11.6	-6.3	0.28
Laura	-13.0	-2.4	0.16
Seagull	-13.1	0.3	0.14
Sergio	-28.6	-6.8	0.23
Zenhgyun1	-11.1	-4.0	0.65
Average	-17.2	-1.9	0.26

Table 3.10: Wiener filter based method. BD-rate over HEVC anchor, over blur based method, and associated II* PSNR difference (positive Δ PSNR represent improvement)

associated to a QP_V value in the range {20,22,25,31} as for *method_3.1*. The remaining of the experimental conditions is as described in Section 3.5.1.3.

Table 3.10 shows the BD-rate gains provided by the proposed Wiener filter based method over the HEVC anchor, and over the blur based method, with the associated variations of the PSNR of II* against II (for $QP_V = 20$) . An average BD-rate gain of 17.2% (up to 28.6% for *Sergio*) is reported over the HEVC anchor, corresponding to a BD-rate gain of 1.9% over the blur based method. Filter coefficients are coded on 12 bits, resulting in a cost of less than 1 kbit for a filter of size 9×9, which approximately represents 0.01% of the total bitrate in average. Results show that compression efficiency can be significantly improved with an advanced filtering adapted to the integral image reconstruction step. It should be noted that the BD-rate variation for *Seagull* is not significant despite the improvement of the PSNR of II* against II (0.14 dB), and that there are even losses for *Fountain* and *Fredo* despite the PSNR improvement. Although the PSNR of II* against II can provide a significant hint on the coding performance under some conditons (e.g. to compare blur filter sizes, as shown in Section 3.5), the results confirm that it is not a perfect indicator of the efficiency of the residual image encoding, which also depends on other characteristics (e.g. smoothness of the residual image).

Decoding runtime is not impacted by the Wiener filter coefficients computation. For the encoding runtime, the number of operations related to the filtering increases because the Wiener filter is applied to the MIs (i.e. on 6512×4880 pixels), while the averaging filter is performed on the view (e.g. 960×720 pixels for *Fountain*) in the blur based methods. However, the Wiener filter based method does not use iterations, allowing a significant reduction of the time attributed to reconstruction. We estimate that the encoding runtime for both types of methods are in the same order.

Table 3.11 shows the results for the disparity adaptive Wiener filter based method. Average BD-rate gains are increased to 17.4% (up to 28.9% for *Sergio*). The disparity adaptive method is not significantly more complex in terms of runtime. One filter per disparity level (e.g. 4 in *Laura*) is processed instead of one for the entire image, but the filters are computed using the same set of data as input (estimates of the auto and cross correlations), only labeled differently.

Image	BD-rate (%)	Comparison with blur based method	
		BD-rate (%)	Δ PSNR (dB) II* vs. II
Fountain	-16.2	0.6	0.54
Fredo	-27.2	5.1	0.06
Jeff	-12.3	-6.9	0.31
Laura	-12.8	-2.1	0.20
Seagull	-13.3	0.1	0.19
Sergio	-28.9	-7.2	0.27
Zenhgyun1	-11.2	-4.1	0.74
Average	-17.4	-2.1	0.33

Table 3.11: Disparity adaptive Wiener filter based method. BD-rate over HEVC anchor, over blur based method, and associated II* PSNR difference (positive Δ PSNR represent improvements)

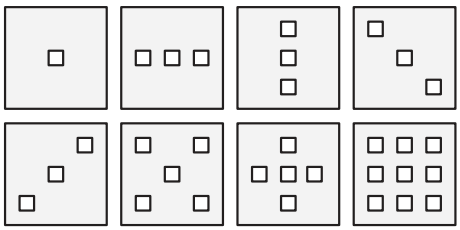


Figure 3.14: Examples of multi-view extraction configurations tested. Left-to-right, top-to-bottom: 1 view, 3 views horizontal, vertical, and diagonal, 5 views diagonal and straight, 9 views

3.7 Proposed methods with several views

In this section, we study the impact of the number of extracted views on the compression performance. As described in Section 3.3, the efficiency of our scheme depends on a tradeoff between the bitrate of the views and the bitrate of the residual image. The goal here is to increase the number of extracted views in order to improve the reconstruction of II*, and therefore reduce the bitrate required to encode the residual image, without increasing too much the bitrate required for the views.

3.7.1 Experimental conditions

It should be noted that although the Wiener filter based method presented in Sec. 3.6 provides additional coding gains, its implementation in Matlab is costly in terms of memory and time consuming. Therefore it could not be enabled for the experiments presented in this section. As in Section 3.6.3, for this experiment, a configuration with fixed values for QP_R and QP_V is used. Each tested QP_R value in the range {10,15,20,25} is associated to a QP_V value in the range {20,22,25,31}. M values in the range {1,...,9} are tested. Several configurations with 3, 5 and 9 views are tested in order to be compared with the single view case. Figure 3.14 shows the relative positions of the patches within the MI for the tested configurations. At the reconstruction of II*, for each MI, the pixels taken from all the views (within the patch and its surroundings) are all blended with an equal weight. In preliminary experiments, the reconstruction was performed by applying a gaussian weighted averaging centered on the patches, however these experiments provided a largely

degraded compression performance. One possible reason for this unexpected result is that blending all the views smoothes the reconstruction unaccuracies and errors in the same way the low-pass filter does. The views are extracted with the disparity-assisted patch blending method [54]. Preliminary experiments also showed that, like in the single view case, using a fixed patch size alters severely the performance (see Section 3.5.2.2). Encodings of the views are performed with the 3D-HEVC reference software (HTM13.0).

3.7.2 Experimental results

Table 3.12 and Table 3.13 show the BD-rate results provided for each of the tested multi-view configurations, with a large and small blending zone respectively (as described in Section 3.5.2.1 and shown in Figure 3.11). Comparison between the two tables shows that using a smaller blending zone improves performance as expected.

The configuration with 3 horizontal views provides the best results with an average BD-rate gain of 22.2% over the HEVC anchor. All the multi-view configurations with smaller blending zone overcome the performance of the single view configuration when considering the average BD-rate. However, some of them provide inconsistent results overall with significant losses for some images. For example the configuration with 3 vertical views is the second best in average but the BD-rate for *Fountain* drops from a 17% gain to a 10.4% gain only. Similarly, one of the 3 diagonal views configuration has a large average BD-rate gain of 19.9% but the gain for *Fredo* drops from 31.2% (which is our higher gain so far) to 26.0%.

In our best configuration with 3 horizontal views, the improvement is quite consistent over the test set, with only slight losses reported for *Fountain* and *Fredo* (from 17.0% to 16.4% and from 31.2% to 28.9% respectively), similar results for *Seagull* and *Laura* (slight gain), and with large gains for the two images that provided the less impressive results with the single view configuration (from 5.4% to 25.8% for *Jeff*, and from 7.1% to 25.9% for *Zenhgyun*). Gains for *Sergio* are also significant (from 23.5% to 31.1%). For images with large improvements, using one view is not sufficient to obtain an accurate reconstructed image, therefore significant angular information is contained in the residual image. Results show that this information is less costly when contained in two additional extracted views, providing a smoother reconstructed image with less errors. These results show that the increase of bitrate to encode several views can be compensated by the improvements of quality for II^* and therefore the decrease of bitrate for II_R.

In some cases, even though the reconstruction of II^* is improved, the impact on the encoding of II_R is not sufficient to compensate the additional cost of the views. For the image *Seagull*, using 3 vertical views improves the PSNR of II^* against II (from 23.9 dB to 24.3 dB approximately) and provides a gain of 1.5% over the single view case when only the bitrate of the residual image II_R is taken into account, but the gain drops to 0 when the bitrates of the views are included. In some other cases, like *Fountain* in the same 3 vertical views case, the improvement of the PSNR of II^* against II (from 19.3 dB to 19.8 dB approximately) does not provide BD-rate gain, even without counting the views (5% loss approximately). This result shows that in these test conditions, the PSNR of II^* against II is not as relevant as in the single view case to predict the compression performance.

Table 3.14 provides the encoding and decoding runtime variations (for *Fountain*) against the anchor in the case of 3 horizontal views, and the percentage of the runtime dedicated to each task. The division of the time for each task is also illustrated in

Image	1 view	3 views				5 views		9 views
		ver.	hor.	diag.		diag.	straight	
Fountain	-17.0	-11.8	-14.4	-8.6	-7.9	-12.2	-12.5	-7.2
Fredo	-31.2	-25.9	-20.3	-15.2	-14.5	-19.3	-20.0	-6.1
Jeff	-5.4	-26.0	-24.4	-10.4	-21.2	-21.0	-24.1	-15.9
Laura	-11.1	-15.2	-13.5	-13.0	-12.5	-10.6	-13.9	-10.2
Seagull	-13.7	-14.5	-13.8	-13.1	-12.9	-10.9	-13.5	-10.7
Sergio	-23.5	-31.2	-27.3	-25.2	-24.0	-25.3	-28.0	-20.8
Zenhgyun1	-7.1	-27.0	-23.8	-18.5	-21.3	-21.9	-24.4	-15.6
Average	-15.6	-21.7	-19.6	-14.8	-16.3	-17.3	-19.5	-12.4

Table 3.12: BD-rate results comparison between multi-view and single view based methods

Image	1 view	3 views				5 views		9 views
		ver.	hor.	diag.		diag.	straight	
Fountain	-17.0	-10.4	-16.4	-9.2	-8.8	-10.5	-12.9	-8.5
Fredo	-31.2	-29.1	-28.9	-27.4	-26.0	-22.6	-27.9	-22.0
Jeff	-5.4	-25.9	-25.8	-10.8	-23.3	-20.2	-25.0	-18.2
Laura	-11.1	-14.0	-13.9	-13.4	-13.2	-8.4	-12.9	-10.5
Seagull	-13.7	-13.4	-13.7	-13.3	-13.3	-8.0	-11.1	-10.9
Sergio	-23.5	-31.1	-31.1	-30.7	-29.4	-24.2	-28.5	-27.2
Zenhgyun1	-7.1	-26.3	-25.9	-18.0	-25.0	-20.8	-24.4	-21.1
Average	-15.6	-21.5	-22.2	-17.6	-19.9	-16.4	-20.4	-16.9

Table 3.13: BD-rate results comparison between multi-view and single view based methods (with smaller blending zones for multi-view configurations only)

Figure 3.15. These results are compared to the results previously given in Table 3.7 and Figure 3.10 for the single view case. Encoding and decoding runtimes are respectively 1.3 and 2.4 times the anchor runtimes when using one extracted view. In the multi-view case, only the runtime for the steps related to the views is impacted. Extraction and filtering consist basically of the same operations repeated for each views, therefore the runtime is multiplied by the number of views (i.e. 3 in our best configuration). Reconstruction runtime is also multiplied, although by slightly less than the number of views as some operations are common for all the views (e.g. normalization). Encoding and decoding times for the first coded view are the same as for one extracted view (as it is also an I frame). Additional runtime is required to encode side views (i.e. two P frames here). In our experimental conditions, encoding time for P frames is approximately 4 times larger than for I frames, while decoding time is similar. Total encoding and decoding time for the scheme using 3 horizontal views are respectively 1.8 and 3.9 times the anchor runtime.

3.8 Combination and comparison with state-of-the-art methods

In this section we compare the results of the proposed scheme to the Self-Similarity method (using HEVC with the Intra Block Copy mode enabled), which is one the most efficient state-of-the-art methods presented in Section 3.2. Additionally we also provide further

	Runtime (%)	against anchor	Extr.	Rec.	HEVC View	HEVC II_R	Others
3 hor. views	Encoding	180	15	18	6	58	3
	Decoding	390	/	57	1	28	14

Table 3.14: *Fountain* - Runtime variation against anchor, and percentage of the total time for each task including: extraction, reconstruction, view and residual encoding/decoding, and blur, subtraction and sum as *others*

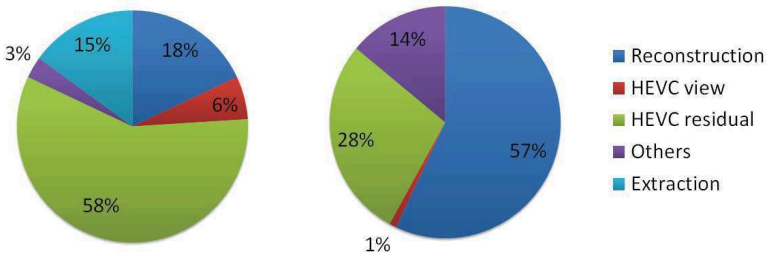


Figure 3.15: Percentage of the total time for each task, *left*) encoding, *right*) decoding

improved results by combining the proposed scheme with the Self-Similarity method.

Table 3.15 provides BD-rate results against HEVC Intra (as anchor). The two first columns correspond to state-of-the-art methods. The Self-Similarity method is tested using HEVC Range extension [101] (HM-RExt7.0, based on HM14.0) with the mode Intra Block Copy (IBC) enabled (see Sec. 3.2). For the Multi-View method, 49 views are extracted using patches of 10×10 pixels, and the views are encoded with 3D-HEVC (HTM13.0). The output integral image is reconstructed from the decoded views. Results for the proposed method with one extracted view (*method_3.1* as presented in this chapter) and three horizontal views are reported in the third and fourth columns. Finally, the last columns show the results for the combination of the proposed scheme with the Self-Similarity method, i.e. the residual image II_R is encoded with the IBC mode enabled, with one view and with three horizontal views extracted.

The proposed method with one extracted view provides significantly better results than the Self-Similarity method for only two images (*Fredo* and *Sergio*), hence the average result is lower with 15.5% against 19.1% gains respectively. However, when improved

anchor: HEVC Intra	Multi-view	HEVC Intra+IBC	Prop. scheme		Prop. scheme + IBC	
			(1 view)	(3 hor. views)	(1 view)	(3 hor. views)
fountain	-19.1	-16.6	-16.7	-16.4	-24.3	-21.1
fredo	-40.2	-23.4	-31.2	-28.9	-39.8	-36.3
jeff	-18.0	-16.4	-5.4	-25.8	-18.4	-29.5
laura	-15.7	-15.0	-11.1	-13.9	-20.2	-18.9
seagull	-26.8	-22.8	-13.7	-13.7	-27.4	-25.5
sergio	-32.8	-21.9	-23.3	-31.1	-32.3	-34.9
zenhgyun1	-26.6	-17.5	-7.2	-25.9	-19.7	-29.8
average	-25.6	-19.1	-15.5	-22.2	-26.0	-28.0

Table 3.15: BD-rate results (%) comparison for the combination of the proposed scheme with state-of-the-art methods

anchor: HEVC Intra+IBC	Prop. scheme + IBC	
	(1 view)	(3 hor. views)
fountain	-10.0	-6.3
fredo	-21.5	-17.1
jeff	-3.0	-16.3
laura	-7.0	-5.7
seagull	-6.3	-4.0
sergio	-13.7	-17.0
zenhgyun1	-3.3	-15.3
average	-9.3	-11.7

Table 3.16: BD-rate results (%) with the Self-Similarity method as anchor

with three horizontal extracted views, the average gain for the proposed scheme is larger (increased to 22.2%). The decomposition of the original image in a multi-view content to be encoded with 3D-HEVC provides large BD-rate gains, with 25.6% in average. However, preliminary experiments performed on 3 sequences (*PlaneAndToy*, *DemichelisCut*, and *DemichelisSpark*) show that this multi-view approach provides very large losses from 25% up to 75% when compared to the 2D approach (i.e. II encoded with HEVC intra), in the case of intra coding only. When temporal prediction is enabled (inter coding), the performance further decreases with losses from 90% up to 130%. As mentioned in this chapter, our scheme is not limited to still images, but it cannot perform efficiently on the available test sequences. On *PlaneAndToy*, the disparity estimation at the view extraction step is disturbed by optical artifacts in a large number of MIs. On *DemichelisCut* and *DemichelisSpark*, the round MIs prevent us from removing the grid pixels efficiently. In the case of our test set composed of still images, the multi-view approach benefits from inter-view correlations that cannot be exploited by the 2D approach, hence resulting in significant BD-rate gains. However, in the case of sequences, this is compensated by the temporal correlations. Although our test set is only composed of still integral images due to the lack of exploitable video content, the purpose of the proposed scheme is also the encoding of sequences, therefore the 2D approach (i.e. HEVC Intra) remains the most relevant anchor.

The combination of the two methods (proposed scheme and Self-Similarity) provide the largest gains with 26% and 28% in average, respectively with one and three extracted views. The Self-Similarity method performs well also on the residual integral image II_R of the proposed scheme because it presents non-local spatial redundancies similarly to the original integral image II. Table 3.16 shows the BD-rate gains for the proposed method with the Self-Similarity method as an anchor. It is interesting to note that even with this efficient anchor, the proposed scheme still provides average BD-rate gains up to 11.7%. For additional comparison, Table 3.17 shows the BD-rate gains for the proposed method with JPEG as an anchor, in order to present the results as in the ICME Grand Challenge (although with a different test set). Gains are very large over JPEG (around 60%).

anchor: JPEG	Prop. scheme		Prop. scheme + IBC	
	(1 view)	(3 hor. views)	(1 view)	(3 hor. views)
fountain	-53.4	-53.2	-57.7	-55.9
fredo	-69.6	-68.5	-73.4	-71.9
jeff	-51.8	-61.8	-58.5	-63.8
laura	-45.7	-47.5	-51.6	-50.8
seagull	-57.8	-57.7	-64.6	-63.7
sergio	-57.8	-61.9	-62.8	-64.1
zenhgyun1	-58.0	-66.0	-63.6	-67.8
average	-56.3	-59.5	-61.8	-62.6

Table 3.17: BD-rate results (%) with JPEG as anchor

3.9 Perspectives

3.9.1 CU level competition with intra mode

In this chapter, the proposed scheme is applied at the frame level, and compared to HEVC Intra. In the HEVC anchor encoding, the whole frame (II) is encoded with intra prediction, and in the proposed scheme, it is the whole residual image II_R that is encoded, additionally to the extracted view(s). However, a combination is also possible at CU level, with a competition between the original intra prediction and the proposed scheme for each CU.

In the first case, the CU is encoded with intra prediction. Neighboring pixels used as patterns for the prediction come from the reconstructed integral image (II_{out} in Fig. 3.2 from Sec. 3.3), hence from the reconstructed neighboring CUs. In the second case, when the CU is encoded with the proposed scheme, the corresponding CU (i.e. at the same position) in II_R must be intra coded. However, that case is not straightforward because if the neighboring CUs have not been encoded with the proposed scheme, corresponding pixels from II_R are not available as patterns for prediction. The structure of the proposed scheme should therefore be modified. An additional reconstructed residual image has to be computed, being the difference between II^* (the integral image reconstructed from the views) and II_{out} (the actual reconstructed/decoded integral image), so that no additional information should be transmitted to the decoder for the prediction. This is expected to alter the quality of prediction for the CUs in II_R.

The only additional cost required is a flag that signals the choice for the coding mode between the original intra mode and the proposed method. This syntax element should be coded using HEVC mechanisms based on prediction and CABAC (Context-Adaptive Binary Arithmetic Coding) contexts in order to reduce the bitrate. This method is expected to be efficient if the gains provided by the possible original intra prediction of some CUs can overcome the additional cost required. Moreover, this new scheme is also compatible with the Self-Similarity method for both modes (i.e. prediction of the original CU or the CU from II_R).

3.9.2 View extraction with dense disparity map

In this chapter, the view extraction step in the proposed scheme uses a disparity map with one disparity value per micro-image. However, when a patch contains (at least) two objects with different depths, (at least) one of the object will be resized with a wrong

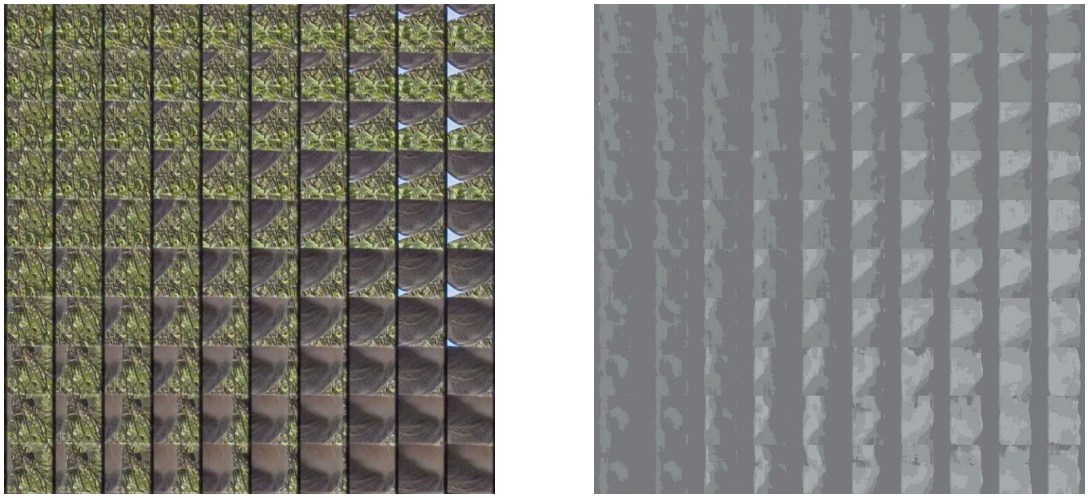


Figure 3.16: Example of dense disparity map for *Laura*

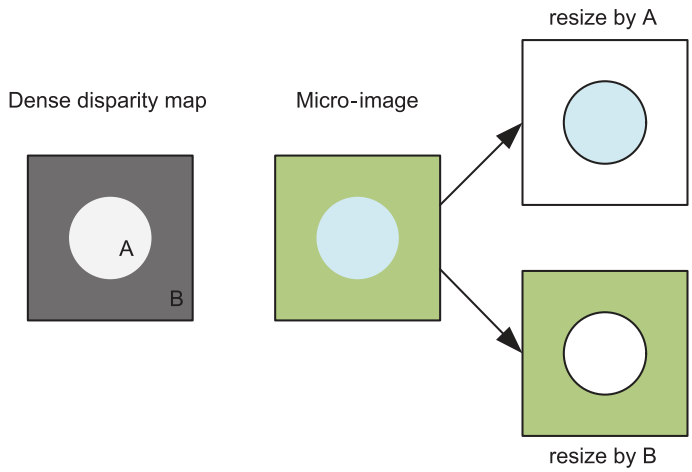


Figure 3.17: Micro-image resizing step with dense disparity map

value, because the single disparity value for the MI corresponds to only one depth value. In this section, we mention side work from another study related to the improvement of the view extraction step. We propose to use dense disparity maps as illustrated in Figure 3.16. In that case, illustrated in Figure 3.17, each layer of depth is resized with corresponding value in the disparity map (e.g. A and B in Figure 3.17). Therefore, the extracted view will present less artefacts, as each object is rendered with correct size. In future work, this method should be tested within the compression scheme proposed in this chapter to improve coding performance. The results will depend on the tradeoff between how much the dense disparity based extraction can improve the quality of the view and of the reconstructed integral image II^* and how much the dense disparity map cost will increase the bitrate.

3.9.3 Display/format scalable feature

Backward JPEG compatible coding schemes have recently been proposed for omnidirectional images and for GIF animations in [102] and [103]. The principle is to offer a scheme that is scalable in terms of format. The process is based on the JPEG marker *APP11*, that permits to insert additional information in the bitstream that is bypassed by a common JPEG decoder, therefore without altering the data regarding the standard. The content can be decoded with a dedicated decoder, for example to be read by a 360° viewer in the case of omnidirectional content or by a GIF viewer in the case of animations. When decoded with a common standard JPEG decoder, however, a JPEG frame is obtained, representing either a view of the scene represented by the omnidirectional content or one frame of the animation (e.g. the first one).

This structure is directly compatible with the structure of the coding scheme proposed in this chapter. The (3D-)HEVC blocks can be replaced by JPEG blocks, and one extracted view from the original integral image can therefore be encoded/decoded by JPEG, while the rest of the content (i.e. the residual stream containing II_R) can be hidden behind a *APP11* flag.

Moreover, this principle can be applied while keeping the HEVC based encoding, for example by replacing the JPEG *APP11* marker by a dedicated flag in a Supplemental Enhancement Information (SEI) message in the HEVC stream. In that case the display scalable feature is available, but the backward compatibility is not because the decoder must be modified to bypass the content behind the flag, and the bistream is not compliant with the current standard anymore. This structure offers a stream that can be either decoded as a natural 2D image (i.e. a view) by a 2D decoder (modified so that the flag should be known and treated by this decoder), either as an integral image for dedicated displays or applications. It should be noted that this display scalable feature is cited in the Call for Proposal (CfP) from the JPEG Pleno group dedicated to integral images compression. In the case described in this section, this feature is completed by the high coding efficiency demonstrated by the experimental results presented in this chapter.

3.9.4 Other perspectives

Finally, in addition to the specific perspectives proposed in this section, several other parts of the scheme can be improved. The encoding of the residual image $\mathrm{II_R}$ is the first aspect that can be though of. As illustrated in Fig. 3.7 (Sec. 3.5.1.4), the residual image has a particular aspect that can specifically be handled by dedicated coding tools.

As mentioned in the introduction chapter, the proposed image coding scheme is completely adapted to video sequences as it is based on HEVC encoding blocks. The extraction and reconstruction blocks should, however, be improved to provide temporal consistency in order to improve coding efficiency.

3.10 Conclusion

In this chapter we propose an efficient integral image compression scheme where a residual integral image and an extracted view are encoded. The residual image is the difference between the original image and an image reconstructed from the view. An average BD-rate gain of 15.7% up to 31.3% over the HEVC anchor is reported. Coding performance largely depends on the configuration of the QP used to encode the view and the size of a low-pass filter applied to the view. A robust iterative RDO process is modeled to select the best configuration, preserving optimal BD-rate gains. We show that the number of iterations can be limited to reduce the runtime while preserving BD-rate gains. We prove that we can assign one single QP for the view to a given QP for the residual with minimal loss, and that the low-pass filter size can be selected using reduced iterations. We show that using the central view extracted with varying patch size provides the best performance. Moreover, iterations on the averaging filter size can be replaced by advanced adaptive filtering techniques to further improve the compression efficiency. We propose to increase the number of extracted views in order to improve the quality of the reconstructed integral image, and therefore reduce the bitrate required to encode the residual image. Compression efficiency is increased with an average BD-rate gain of 22.2% (up to 31.1%) over the HEVC anchor, at the cost of a realistic increase in runtime. Finally, average BD-rate gains are brought up to 28% when the proposed scheme is combined with the state-of-the-art Self-Similarity method. This complete study results in a codec with realistic coding performance and runtime, and with several efficient configurations possible.

Part II is dedicated to the compression of integral (or plenoptic) imaging content. As discussed in Chapter 2, this technology currently provides a dense sampling of the light-field with views that are constrained within a narrow angle of view. Target applications such as Free Navigation can however benefit, or even require, a larger angle of view. Therefore in Part III, we study the compression of Super Multi-View content, captured by camera arrays with a large baseline, providing a set of views that is less dense. This study also includes the process of view synthesis as a tool to create intermediate views (that may or may not have been captured).

Part III

Super Multi-View

Chapter 4

Subjective evaluation of super multi-view compressed contents on light-field displays

4.1 Introduction

Efficient compression of Super Multi-View (SMV) content is a key factor for enabling future light-field video services. In this context, the in-depth understanding of the interactions between video compression and display is of prime interest. However, evaluating the quality of 3D content is a challenging issue [104, 105]. In the context of SMV content, the increased number of views, the increased number of synthesized views (depending on the configuration), and the novel characteristics of the target display systems make it even more challenging. The main goal of this chapter is to assess the impact of compression on perceived quality for light-field video content and displays. To the best of our knowledge, the work presented in this chapter is the first to carry out subjective experiments and to report results of this kind.

First, assessing a range of bitrates required to provide an acceptable quality for compressed light-field content will give a cue on the feasibility of transmitting this kind of content on future networks. It is also needed to understand how much view synthesis disturbs the general quality (both subjectively and objectively). Moreover, depth based rendering and synthesized views make the PSNR less relevant [106], but no other metric is currently accepted as more appropriate. One of the goals in this chapter is to evaluate how much the use of the PSNR remains relevant, and if future codec developments can keep on relying on this basic indicator. Finally, as classical compression is well known to generate artifacts such as blocking, ringing, etc., one of our goals is to possibly observe new compression artifacts that may affect the specific aspects of the visualization of light-field content like the motion parallax, the perception of depth, etc. Our experiments provide first results showing that improvements of compression efficiency, depth estimation and view synthesis algorithms are required. However, results also show that the target bitrates for the use of SMV appears realistic according to next generation compression technology requirements, when display systems will be fully ready.

This chapter is organized as follows. The principles of the target light-field display system Holografika's Holovizio [14] are described in Section 4.2. In Section 4.3, preliminary experiments are conducted in order to select the most relevant coding configurations for

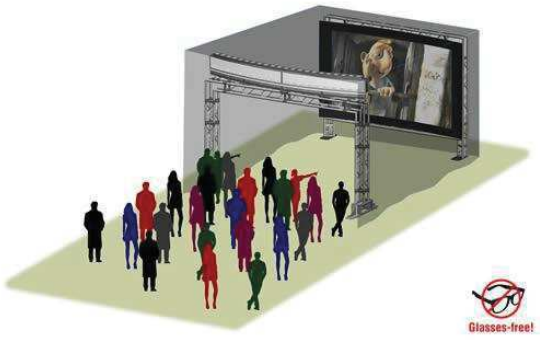


Figure 4.1: Holovizio C80 cinema system [14]

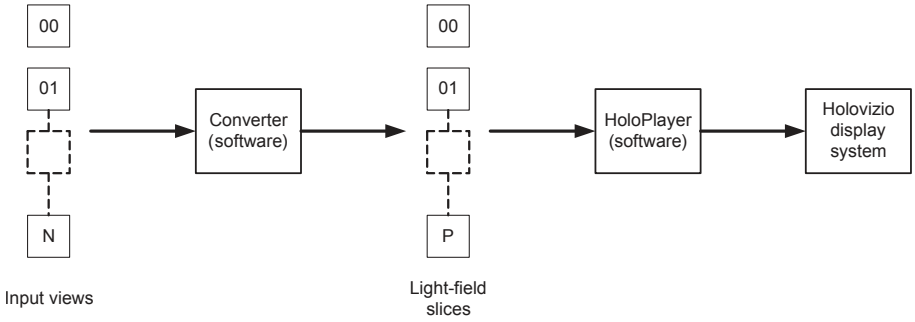


Figure 4.2: Conversion step of the input views before the display

SMV content. Several configurations with varying ratios of coded/synthesized views are compared in Section 4.4 and objective results are shown. Subjective evaluation of the tested configurations is described in Section 4.5, and subjective results are presented and analyzed. Conclusions and perspectives are drawn in Section 4.6.

4.2 Super Multi-View display system used in our experiments

4.2.1 Example of light-field display system

As described in Section 2.3, SMV display systems, also refered to as light-field displays, take tens to hundreds of views as input. Several display systems are based on a front or rear projection [48]. Each projection unit projects from a different angle onto a screen. The screen surface has anisotropic properties (which can be obtained optically or with a holographic diffuser for example), so that the light rays can only be seen from one direction which depends on the projection direction. The Holovizio C80 display system, which has been used in our experiments, is illustrated in Figure 4.1 and consists of a large screen (3×1.8 meters) and of 80 projection units with a 1024×768 resolution, controlled by a rendering cluster. It offers a viewing angle of approximately 40°. Technical specifications and details are available at [14].

Name	Fps	Duration (s)	Resolution	Cam. Setup	Type
ChampagneTower	30	6	1280x960	Linear	Natural
Pantomime	30	6	1280x960		
Dog	30	6	1280x960		
T-Rex	30	6	1920x1080	Linear	CG
Bunny	24	5	1280x768		

Table 4.1: Natural and computer generated (CG) content used in our experiments

4.2.2 Light-field conversion

As illustrated in Figure 4.2, the input SMV content needs to be converted to be displayed on the Holovizio system. In the experiments described in the following of this chapter, there are 80 views as input (N=80 in Figure 4.2), captured by 80 cameras horizontally aligned in a linear arrangement. Measures performed by Holografika on each of their systems showed that for the display system used in our experiments (C80) it is useless to have more than 80 or 90 views. These views as well as the parameters of the camera rig (baseline, distance from the center of the scene, dimensions of the region of interest, etc.) are provided to the converter. Most of the common video and image formats (e.g. jpg, png, avi, etc.) are supported (as input and output) by the converter. It should be noted that the number of input views N is not fixed and can be more or less than 80. The converter outputs P=80 light-field slices, which are provided to the player (software) at the display step. The whole image projected by a single projection unit cannot be seen from a single viewing position [107], therefore one projection unit represents a light-field slice, which is composed of many image fragments that will be perceived from different viewing positions. The number of light-field slices P is fixed for a given display system as it corresponds to the number of projection units. Hence N should not necessarily be equal to P.

4.3 Preliminary encoding configurations experiments

In the following, we report the results of preliminary tests performed in order to select the most relevant parameters, encoding configurations and encoding structures to encode the content included in the following subjective quality evaluation (Section 4.5).

4.3.1 Experimental content

The experiments in this chapter include the SMV content described in Table 4.1. Dog, Pantomime, and Champagne Tower sequences [108] have been captured with the same camera system. Big Buck Bunny [109] and T-Rex [14] are sequences generated from 3D scenes (with Blender [110] and 3ds Max [111] respectively). A significant difference is reported for the coding performance of content acquired with linear or arc camera arrangement [112]. As a consequence, only linear content is exploited in this work, in order to avoid that camera setup variations affect our conclusions. The comparison between the two kinds of contents is mentioned in Chapter 6. As the coding efficiency and the quality provided by the light-field display system also depend on the number of input views, 80 views are used for each sequence.

Configuration			PSNR Y (dB)	
Precision	Search Level	Filter	VSRS4.0	HTM10.0 Renderer
1	1	1	33,8	34,0
4	4	2	32,0	32,2
Provided depth maps			33,8	34,3

Table 4.2: Preliminary results for DERS configuration

4.3.2 Depth estimation

As described in Chapter 2, depth maps can be captured, estimated from the texture views, or automatically generated for CG content. The depth maps used for experiments in this paper are estimated with DERS6.0 (Depth Estimation Reference Software [60]). Preliminary experiments are performed in order to compare several values for the following parameters of this software:

- Precision (1: Integer-Pel, 2: Half-Pel, or 4: Quarter-Pel), corresponding to the level of precision chosen to find correspondences,

- Search Level (1: Integer-Pel, 2: Half-Pel, or 4: Quarter-Pel), corresponding to the level of precision of candidate disparities,

- Filter (0: Bi-linear, 1: Bi-Cubic, or 2: MPEG-4 AVC 6-tap), corresponding to the upsampling filter used to generate image signals at sub-pixel positions.

In these preliminary experiments, the depth maps for views 37 and 39 of Champagne sequence are estimated on 30 frames. The view 38 is then synthesized with VSRS4.0 (View Synthesis Reference Software [64]) and with the HTM10.0 renderer [19] (see next section). The PSNR of this synthesized view is computed against the original view 38. The depth maps provided with the Champagne sequence [108] (which are estimated semi-automatically with DERS) are also tested for comparison. Table 4.2 shows that the lower values for the tested parameters provide a better PSNR for the synthesized view, and that this result is closer to the result obtained with the semi-automatically estimated depth maps provided with the sequence. Selecting higher values for the tested parameters implies the use of more advanced tools (e.g. with higher precision). These values provide depth maps with a smoother aspect and apparently less artifacts, however this involves a decrease of the PSNR of the synthesized view (i.e. more synthesis artifacts appear). The significant filtering obtained with high precision values can improve the visual aspect of the depth maps, however this smoothing operation alters the information when the depth maps are used as tools for view synthesis. In our experiments, depth maps are encoded with the views in order to be used at the decoder side to synthesize views that are not encoded (i.e. skipped at the encoder side). As view synthesis is the main purpose of the depth map estimation here, the configuration with lower parameter values is used to estimate all the depth maps included in our experiment phase.

4.3.3 View synthesis

We have performed experiments to compare the 3D-HEVC Renderer [19] and VSRS4.0 [64] with several configurations obtained by assigning different values for the following parameters:

Configuration					PSNR Y	Time
Precision	Filter	Boundary noise	Mode	Blend	(dB)	(s)
2	1	1	1	1	37,4	33
2	1	0	0	1	38,3	60
2	1	0	1	1	37,8	26
2	1	1	0	1	36,4	61
2	1	0	0	0	38,3	58
2	**0**	**0**	**0**	**1**	**38,5**	**57**
2	2	0	0	1	38,4	58
1	1	0	0	1	37,7	91
4	1	0	0	1	38,3	61
4	**0**	**0**	**0**	**1**	**38,5**	**60**
4	2	0	0	1	38,4	59
3D-HEVC Renderer (HTM10.0)					**38,6**	**17**

Table 4.3: Preliminary results for view synthesis software configuration

- Precision (1: Integer-Pel, 2: Half-Pel, or 4: Quarter-Pel), and
- Filter (0: Bi-linear, 1: Bi-Cubic, or 2: MPEG-4 AVC), which are both used for values at sub-pixel positions,
- Boundary Noise removal (0: disable, 1: enable), which process artifacts on edges,
- Mode (0: General, 1: 1D Parallel), corresponding to the type of camera arrangement,
- Blend (0: disable, 1: enable), used to blend the right and left input views.

View 38 of Pantomime sequence is synthesized on 30 frames and the PSNR is computed against the original view 38. Table 4.3 shows the PSNR results and the processing time for this preliminary experiment. HTM10.0 Renderer provides better results than VSRS4.0 in our experiments conditions and is also faster (approximately one third of the time of most VSRS configurations). Hence HTM10.0 Renderer (with default configuration) is used to synthesize all the intermediate views in our experiment phase.

4.3.4 Group of views (GOV)

Encoding 80 dependent views is very demanding in terms of memory (Random Access Memory - RAM). To avoid memory limitations, a configuration with Groups Of Views (GOV) is used. For example in Figure 4.3, the views from V_0 to V_{x-1} are dependent because of the IPP structure, and it is also the case for views from V_x to V_{2x-1}. However, these two GOVs are independent from each other, as no view from one group is predicted by a view from the other group. Therefore in practice these two groups can be processed and especially stored separately. Table 4.4 compares the performance of encoding 80 views with groups of 16 views against groups of 9 views. For this experiment, MV-HEVC [19, 113] reference software version 10 is used (HTM10.0 with macro HEVC_EXT=1). 180 frames of Champagne, Dog and Pantomime sequences are encoded with QPs (Quantization Parameters) 20-25-30-35. IPP inter-view reference coding structure is used (see Sec. 4.3.5). The results are provided using the Bjøntegaard Delta rate (BD-rate) metric [97], which

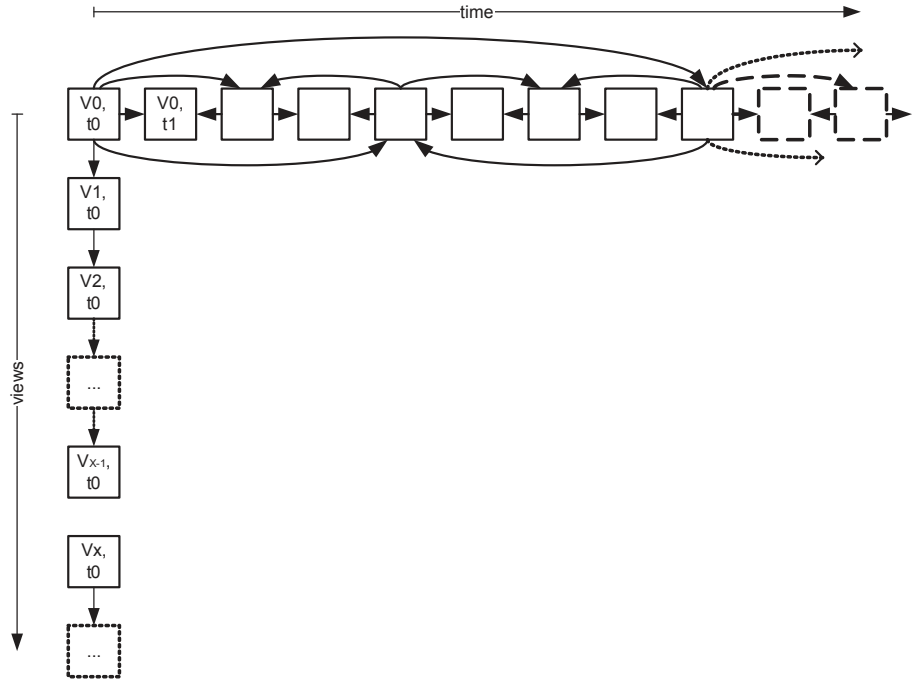


Figure 4.3: Group of X views with hierarchical temporal prediction structure and IPP inter-view prediction structure

GOV 16 vs. GOV 9 (mean PSNR on 80 views)	
ChampagneTower	-0,9%
Dog	-5,2%
Pantomime	-3,1%
Mean	-3,1%

Table 4.4: BD-rate performance of GOV size 16 against GOV size 9

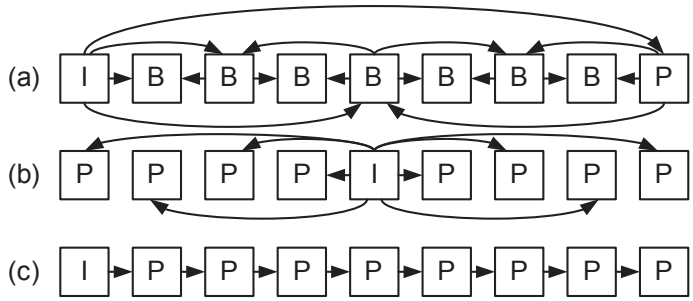


Figure 4.4: Inter-view reference structures within a GOV:
(a) Hierarchical, (b) Central, (c) IPP

Ref: Central (b) (average PSNR on 80 views)		
Sequence	IPP (c)	Hierarchical (a)
ChampagneTower	-7,5%	-0,9%
Dog	-6,1%	-5,5%
Pantomime	-2,9%	2,3%
Mean	-5,2%	-1,0%

Table 4.5: BD-rate performance depending on the inter-view reference structure within GOVs

computes the average bitrate saving (in percentage) for a given quality between two rate-distortion curves, as described in [114]. Table 4.4 shows that using a larger group (from 9 to 16 views) provides an average BD-rate gain of 3.1%. The insertion of I-frames to create GOVs has a non-negligible impact on the BD-rate results. However, this limitation in the configuration is relevant for future use cases because the memory limitation is a practical reality. Moreover GOVs allow parallel processing at both the encoder and decoder side, prevent from the loss of all the views when losing one view due to network errors for example, and provide some limits on error propagation into other views when losing one view.

4.3.5 Inter-view reference pictures structure

In this section we compare 3 inter-view reference structures inside the GOVs, illustrated in Figure 4.4 as follows: Hierarchical (a), Central (b), and IPP (c). Table 4.5 shows the BD-rate performance for these 3 structures with groups of 9 views. IPP is the most efficient inter-view reference structure for this experiment. The experiment is extended with some views skipped to simulate the encoding of a subset of the views as in configurations including view synthesis. The main goal is to verify that the IPP structure remains the most efficient in these configurations. 9 views are encoded with an increasing baseline from 1 view skipped (referred to as skip1) to 9 views skipped (referred to as skip9) between two coded views. Results are shown in Tables 4.6, 4.7, and 4.8. As expected when the baseline increases (more distance between the coded and the reference views), IPP remains the most efficient structure.

Baseline: skip1	Ref: Central (b)	
Sequence	IPP (c)	Hierarchical (a)
ChampagneTower	-8,1%	-1,3%
Dog	-2,9%	1,1%
Pantomime	-8,4%	2,0%
Mean	-6,5%	-1,3%

Table 4.6: BD-rate performance with different inter-view reference structures (with 1 view skipped)

Baseline: skip3	Ref: Central (b)	
Sequence	IPP (c)	Hierarchical (a)
ChampagneTower	-8,9%	-5,7%
Dog	-9,2%	-4,4%
Pantomime	-15,8%	-6,2%
Mean	-12,6%	-5,6%

Table 4.7: BD-rate performance with different inter-view reference structures (with 3 views skipped)

Baseline: skip9	Ref: Central (b)	
Sequence	IPP (c)	Hierarchical (a)
ChampagneTower	-7,4%	-4,9%
Dog	-8,2%	-1,3%
Pantomime	-11,3%	-3,9%
Mean	-9,6%	-3,4%

Table 4.8: BD-rate performance with different inter-view reference structures (with 9 views skipped)

4.4 Objective experimental results

Based on the preliminary results obtained in Section 4.3, the content is encoded with IPP inter-view reference structure and groups of 16 views (i.e. one intra frame every 16 views). For the configuration where all the views are encoded (i.e. without skipped views), QPs 15, 17, 20 to 30, 32, 35, 37 and 40 are used in order to provide a large and dense range of bitrates. For the configurations with views skipped at the encoder, QPs 20, 25, 30, 35 are used. Resulting PSNR-bitrate curves are illustrated in Fig. 4.5, 4.6, 4.7, 4.8, and 4.9. PSNR values between 30 and 40 dB are generally considered as corresponding to acceptable to good qualities for 2D images. Most of the curves in this experiment are above this 30 dB limit, except for some skip9 curves (Dog, Champagne).

For the content captured from real scenes (Dog, Champagne, Pantomime), the PSNR decreases as the number skipped/synthesized views increases (for a given rate). For example, with the Dog sequence (Figure 4.5), the gap between the different curves for a given bitrate is approximately from 2 to 3 dB. This is mainly due to the limitations of the PSNR, which is severe with synthesis artifacts (leading to a significant decrease of objective quality) that are hardly perceptible by human observers (see Section 4.5).

For the computer generated content (T-Rex, Bunny), the decrease in PSNR from one configuration to another is less severe. The skip1 and even skip3 configurations can even provide better results than the configuration without synthesis. One of the reasons can be the noise-less aspect of the content. Because of this characteristic, first the depth estimation (which is based on block-matching algorithms) is more accurate, and thus allows a better synthesis. Secondly, the PSNR metric is also less disturbed by hardly perceptible noisy variations. These results show that the weaknesses of the synthesis algorithms affect the coding scheme.

4.5 Subjective evaluation

In this section, we describe the subjective evaluation of SMV compressed content encoded as described in Section 4.4. Experimental conditions and evaluation methodology are first described, and subjective results are then presented and analyzed.

4.5.1 Experimental conditions

4.5.1.1 Raw video files constraint for the display step

As described in Section 4.2.2, the captured views are converted into light-field slices before the display step. Each light-field slice is associated to a projection unit of the Holovizio display system (the projection units are illustrated in Figure 4.10). The encodings performed in our experiments are done with raw video files (YUV4:2:0 raw data contained in .yuv files) as input of the encoder and as output of the decoder (and renderer), so that the video data suffers no degradation, apart from the compression effect which is aimed to be observed in the experiments. Additional software has been developed by Holografika in order to handle the YUV raw video format in the converter and the player. The use of a raw video format induces very large file sizes. During our experiments, in order to have a smooth playback on the display, the light-field slices in raw video format had to be copied directly to the ramdisk (more specifically on a compressed ramdisk) of their associated nodes (e.g. light-field slices 72 to 79 are copied on Node #09's ramdisk, as illustrated in Figure 4.10).

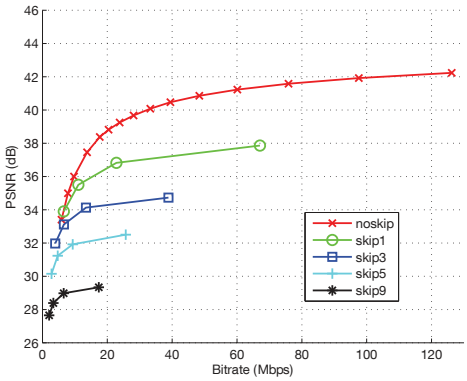


Figure 4.5: PSNR-bitrate (Dog)

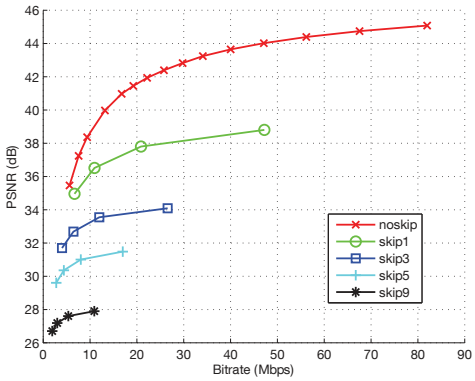


Figure 4.6: PSNR-bitrate (Champagne)

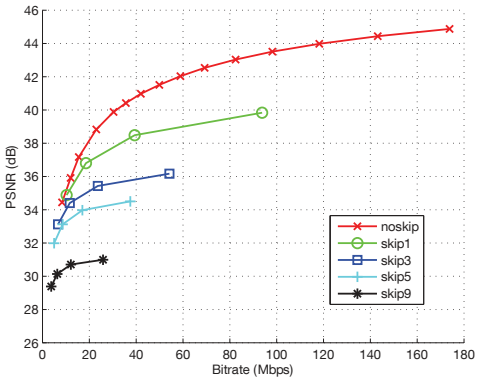


Figure 4.7: PSNR-bitrate (Pantomime)

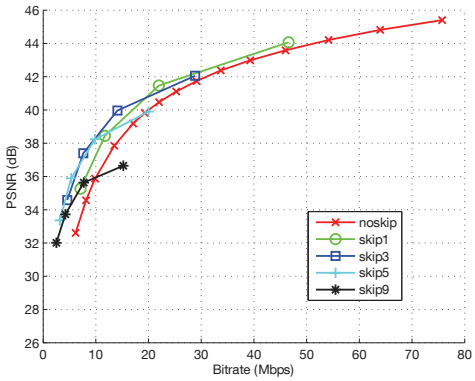


Figure 4.8: PSNR-bitrate (T-Rex)

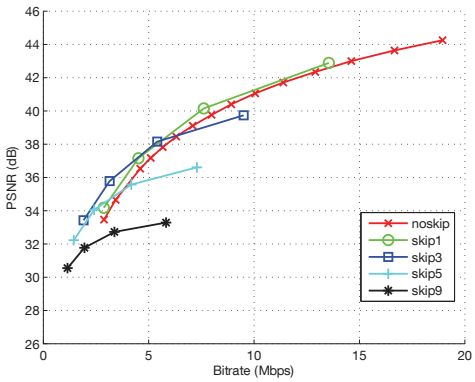


Figure 4.9: PSNR-bitrate (Bunny)

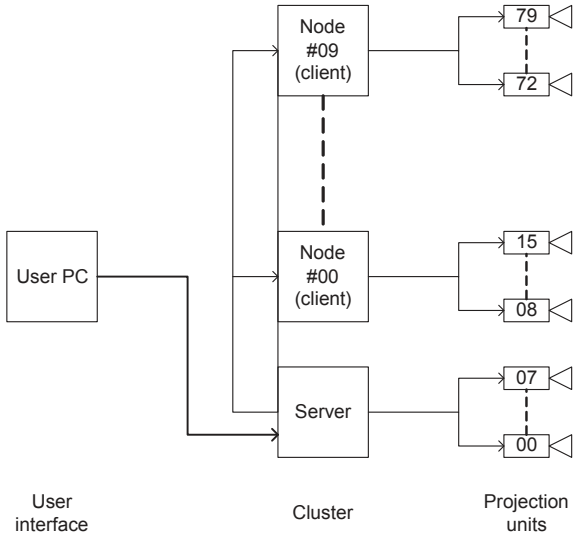


Figure 4.10: Display system structure

4.5.1.2 Content selection

Three sequences are included in the subjective evaluation: Dog, Champagne, and Bunny. In Dog, a woman plays with a dog in front of a colored curtain with dots. Champagne represents a woman serving champagne in glasses in front of a black curtain. Bunny is a computer generated scene with a rabbit coming out of a burrow with grass in the background. Example frames from the sequences are shown in Figure 4.11.

Each Holovizio light-field display system has a field of depth in which the content of the scene must be included to be displayed correctly. The objects in the scene that are outside of these depth bounds (i.e. too far or too close) present ghost-like artifacts. In our experiments, it is the case for the objects in the background of the Champagne sequence as well as for the background of the Dog sequence. A small part of Bunny is also too close in the foreground but in a slighter way which does not impact significantly the visualization.

The sequences encoded in the preparation phase as described in Section 4.4 have been evaluated in a preliminary subjective evaluation session in order to select relevant configurations to be included in the limited time of one test session (see Section 4.5.1.3). Based on this preliminary visualization the following configurations are included in the evaluation: QPs 25-30-35 for the noskip configurations, QPs 20-30 for skip1 and skip3, and only QP 20 for skip5 and skip9. The uncompressed content is also included in the evaluation as a reference to assess the quality of the compressed content.

4.5.1.3 Subjective evaluation methodology

The evaluation methodology used in our experiments is the double-stimulus impairment scale (DSIS) method [115]. The double-stimulus method is cyclic. The assessor is first presented with an unimpaired reference, and then with the same picture impaired. In our experiments the first picture is the original (uncompressed) sequence, and the second picture is compressed and possibly synthesized. We followed the variant #2 of the DSIS method for which the pair is showed twice to the assessor. Following this, he is asked to vote on the second, keeping in mind the first unimpaired sequence. The rating scale is showed in Table 4.9. The choice of the DSIS method is motivated by the fact that the

(a) Dog (b) Champagne (c) Bunny

Figure 4.11: Example frame for each sequence.

Score	Impairments
5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Table 4.9: ITU-R impairment scale [115]

tested content and the display system already present flaws or artifacts that could prevent them from being rated as excellent. By using a comparative method like DSIS, we can ignore these aspects and only focus the evaluation on the compression/synthesis artifacts (which are the causes of the rated impairments).

4.5.1.4 Experimental set-up

Subjective experiments have been performed at Holografika's facilities (a surface of approximately 100m^2 with 3 meters height ceiling and a black curtain that halves the room) where the system is usually stored and used for demonstrations. The light-field display has been calibrated for geometry and intensity before the tests using proprietary Holografika tools, the same way calibration is performed for a new installation. Lighting conditions were the same as for demonstrations, i.e. in a dark room with sparse sources of diffuse light (e.g. distant windows with curtains). No particular attention was paid to obtain a specific brightness measure (in cd/m^2) as no recommendation fits our experimental conditions on this point to our knowledge. Figure 4.12 illustrates the experimental set-up. For each viewing session there was between one and six subjects. Subjects were sitting at

Figure 4.12: Experimental setup

a viewing distance of approximately 6 meters from the screen (which has 3×1.8 meters dimensions), in the 40° angle recommended for the C80 display system. The experiments were performed with 16 subjects. The subjects are employees of Holografika and students from Budapest universities. Various categories of Holografika's employees participated in the evaluation (technical staff, engineers, programmers, etc.), which are not all working directly on light-field imaging, neither with the display systems, and there are no relevant differences between the results when separating the employees from Holografika and the others in two Expert/Non-expert groups. This can be explained by the fact that even if the panel contains light-field experts, the light-field related artifacts and characteristics in the sequences were present both in the original and compressed one. Thanks to the DSIS method, only compression artifacts are taken into account during the evaluation, therefore we have an homogenous panel of subjects who are non-experts in compression. Subjects have normal or corrected to normal vision.

Each session lasted roughly 30 minutes (as recommended in [115]). The duration required for 3DTV subjective evaluation is discussed in [116]. For 2D video, a 10 seconds duration is recommended in [115]. For 3D video, there are two conflicting arguments: i) as 3DTV is closer to the human natural viewing behavior, less time is needed to judge the quality, ii) more time is needed since more information is contained in the additional dimension. This discussion also applies for light-field video. In [117], the presentation time only had little effect on subjective evaluation results with durations of 5 and 10 seconds tested. Based on these considerations, and to limit the session duration as well as the encoding and conversion processing time, the duration of the tested content is limited to 5-6 seconds. From our observations, this 5-6 seconds duration did not appear to be too short. Pairs of sequences were displayed two times and artifacts, when present, were noticeable with one visualization.

Pencils and paper sheets with a scoring table to fill were provided. Before each evaluation session, written instructions describing the process in details as in Sec. 4.5.1.3 and [115] were provided and discussed with the subjects. One pair of sequences was first shown as a training phase, so that the subjects could experience the evaluation process and see examples of degradation types.

4.5.1.5 Light-field display specific aspects

In the evaluation of Super Multi-View compressed contents on light-field displays, different types of artifacts are observed at several steps in the process including capture, coding and display (transmission and/or storage are out of the scope here). The most significant ones are listed and discussed in the following. The very first artifacts in the content are created at acquisition (e.g. out-of-focus, low contrast, noise, etc.), however they should be considered as characteristics of the content (because in some cases, distinguishing out-of-focus from artistic blur can be completely arbitrary for example) and should not impact the score in our case. Typical 2D compression artifacts (such as block artifacts) are introduced during the encoding. In configurations with synthesis, additional artifacts are present in the synthesized views. They are mainly block artifacts and synthesis errors flickering on objects edges. For multi-view (and SMV by extension) content, the differences between cameras (e.g. color calibration, brightness, etc.) can also be sources of artifacts. Additionally, the light-field conversion and the light-field display system itself are also potential sources for new artifacts that need to be listed and studied. Moreover, the impact of these artifacts (new ones and common ones listed above) on the perception

of elements specific to light-field display and 3D imaging (e.g. the motion parallax, the depth, etc.) also needs to be studied.

The study in this chapter is focused only on the compression, therefore it does not take into account the variety of all the artifacts that occurs in the evaluated content. As mentioned above, this subjective evaluation based on the DSIS method provides results that are not impacted by the artifacts that are not related to compression or synthesis, because they are present in the original and the compressed content and therefore not scored. Our work provides preliminary hints and observations concerning the impact of light-field conversion and the perception of motion parallax in Sec. 4.5.6 and Sec. 4.5.7 respectively.

4.5.1.6 Statistical analysis methodology

According to Chapter 2.3 in [118], it should be noted that since the panel size is relatively small (16 subjects), it is more relevant to compute the 95% confidence interval (CI) assuming that the scores follow a t-Student distribution, rather than a normal distribution as suggested in the ITU recommendation [115].

Two methods are used in our experiments to detect potential outliers. The first method is described in the recommendation [115] (for the DSIS evaluation). The principle of this method for screening the subjects is as follows. First, the β_2 test is used to determine if the distribution of scores for a given tested configuration t is normal or not. The kurtosis coefficient (β_2) of the function (i.e. the ratio between the fourth order moment m_4 and the square of the second order moment m_2) is calculated as in Equations (4.1) and (4.2), with N the number of observers/scores, and with u_i the i^{th} score. If β_2 is between 2 and 4, the distribution may be considered to be normal.

$$\beta_2 = \frac{m_4}{(m_2)^2} \tag{4.1}$$

$$\text{where } m_x = \frac{\sum_{i=1}^{N}(u_i - u_{mean})^x}{N} \tag{4.2}$$

Then for each tested configuration t, two values are processed as in Eq. (4.3) and (4.4): P_t corresponding to the mean value plus the associated standard deviation S_t times 2 (if normal) or times $\sqrt{20}$ (if non-normal), and Q_t corresponding to the mean value minus the associated standard deviation S_t times 2 or times $\sqrt{20}$.

$$\begin{aligned} P_t &= u_{mean} + 2 \times S_t \text{ (if normal)} \\ \text{or } P_t &= u_{mean} + \sqrt{20} \times S_t \text{ (if non-normal)} \end{aligned} \tag{4.3}$$

$$\begin{aligned} Q_t &= u_{mean} - 2 \times S_t \text{ (if normal)} \\ \text{or } Q_t &= u_{mean} - \sqrt{20} \times S_t \text{ (if non-normal)} \end{aligned} \tag{4.4}$$

Then for each observer i, every time a score is found above P_t a counter P_i associated with this observer is incremented. Similarly, every time a score is found below Q_t a counter Q_i associated with this observer is incremented.

The following two ratios must be calculated: $P_i + Q_i$ divided by the total number of scores T for each observer, and $P_i - Q_i$ divided by $P_i + Q_i$ as an absolute value. If the first ratio is greater than 5% and the second ratio is less than 30%, then observer i must

be eliminated as shown in Eq. (4.5).

$$\begin{aligned} &\text{if } \frac{(P_i + Q_i)}{T} > 0.05 \qquad \text{and} \qquad |\frac{(P_i - Q_i)}{(P_i + Q_i)}| < 0.3 \\ &\text{then reject observer } i \end{aligned} \tag{4.5}$$

The second method is described in Chapter 2.3 of [118] as follows. For each tested configuration t, the interquartile range corresponds to the difference between the 25^{th} and the 75^{th} percentile. For a given tested configuration t, if the score $score_{i,t}$ of an observer i falls out of the interquartile range by more than 1.5 times, then this score is considered as an outlier score, as shown in Eq. (4.6). An observer is considered as outlier (i.e. is eliminated) if at least 20% of his scores are considered as outlier scores according to Eq. (4.6).

$$\begin{aligned} &\text{if } score_{i,t} < q_{t,25^{th}} - 1.5 \times (q_{t,75^{th}} - q_{t,25^{th}}) \\ &\text{or } score_{i,t} > q_{t,75^{th}} + 1.5 \times (q_{t,75^{th}} - q_{t,25^{th}}) \\ &\text{then } score_{i,t} \text{ is an outlier score} \end{aligned} \tag{4.6}$$

4.5.2 Subjective results

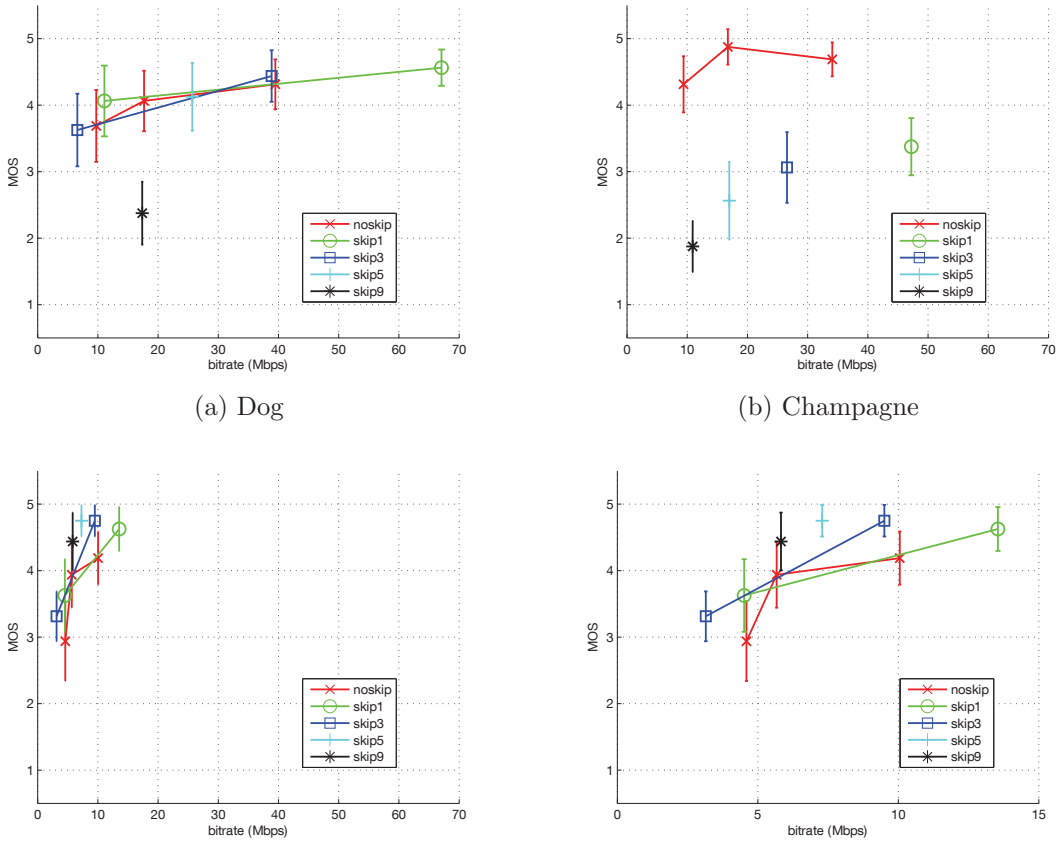
The raw data obtained after the subjective evaluation sessions is an array of 400 scores (25 sequences × 16 subjects). Table 4.10 shows the mean opinion score (MOS) for each tested configuration and its associated bitrate. Figure 4.13 (a), (b) and (c) show the results for Dog, Champagne and Bunny sequences respectively. The subjective scores (MOS) and associated confidence intervals (CI) are presented on the y-axis and the associated bitrates on the x-axis. The bitrate values are given in megabits per second (Mbps). Table 4.10 and MOS-bitrate curves show that the mean scores are globally coherent and increase/decrease as expected relatively to the QPs and configurations. The only obvious incoherence is the score (of 4.9) for Champagne sequence with the noskip configuration at approximately 16.8 Mbps (with QP 30) which is larger than the score (of 4.7) attributed to the same sequence also with noskip configuration at approximately 34.1 Mbps (with QP 25 which is less severe). However this could be explained by the fact that these two scores are very close to each other and to the highest score. Moreover, a statistical analysis of the distribution of the scores (e.g. t-test [118]) would not define them as different scores because the CIs are superimposed. No outliers were detected in our panel using both methods. This, in addition to the reasonable size of the CIs, shows the reliability of the results of this evaluation.

For the Dog sequence, the curves are close to each other for noskip, skip1, skip3 and skip5 configurations. For example, the configurations skip3 and noskip (with QP 20 and 25 respectively) obtain approximately the same score (*Perceptible but not annoying*) at a bitrate of approximately 40 Mbps (rightmost point on the dark blue curve, and rightmost point on the red curve respectively). There is a tradeoff here because the reduction of bitrate due to reduced number of coded views allows a less severe compression, but induces synthesis artifacts. For this sequence with the skip9 configuration, impairments are rated between *Slightly annoying* and *Annoying* even with a bitrate of 17 Mbps for which the compression is not very severe (QP 20). This means that skip9 cannot be considered realistic here, because this low score is mainly due to the synthesis artifacts.

For the Champagne sequence, the noskip configuration is rated *Perceptible but not annoying* even at 10 Mbps (with QP 35). The curve shows that the limit with *Slightly annoying* score should be obtained at an even smaller bitrate value. The configurations

Sequence	Configuration	#	QP	Bitrate (Mbps)	MOS
Dog	noskip	1	25	39,5	4,3
		2	30	17,7	4,1
		3	35	9,7	3,7
	skip1	4	20	67,1	4,6
		5	30	11,1	4,1
	skip3	6	20	38,9	4,4
		7	30	6,6	3,6
	skip5	8	20	25,7	4,1
	skip9	9	20	17,4	2,4
Champagne	noskip	10	25	34,1	4,7
		11	30	16,8	4,9
		12	35	9,4	4,3
	skip1	13	20	47,2	3,4
	skip3	14	20	26,6	3,1
	skip5	15	20	17,0	2,6
	skip9	16	20	10,9	1,9
Bunny	noskip	17	25	10,1	4,2
		18	30	5,7	3,9
		19	35	4,6	2,9
	skip1	20	20	13,5	4,6
		21	30	4,5	3,6
	skip3	22	20	9,5	4,8
		23	30	3,2	3,3
	skip5	24	20	7,3	4,8
	skip9	25	20	5,8	4,4

Table 4.10: Mean Opinion Scores (MOS) for each tested configuration and associated bitrate



(a) Dog (b) Champagne

(c) Bunny (d) Bunny (close-up)

Figure 4.13: MOS scores and associated bitrates.

with synthesis (only QP 20 on the curves) are not rated over *Slightly annoying* here, except for skip1 at 47 Mbps (QP 20), but this point is anyway associated to a bitrate already larger than for noskip rated *Imperceptible* at approximately 35 Mbps (QP 25). For this sequence, the configurations with view synthesis cannot be considered effective nor realistic (in our experimental conditions).

For the Bunny sequence, all the configurations with synthesis and QP 20 are rated between *Slightly annoying* and *Imperceptible*: skip1 at 13.5 Mbps, skip3 at 9.5 Mbps, and skip5 at 7 Mbps are very close to *Imperceptible*, noskip at 10 Mbps is closer to *Perceptible but not annoying* because of the compression artifacts (with QP 25), and skip9 at 6 Mbps also, because of the synthesis artifacts that appear. For this sequence, several configurations are close to *Slightly annoying* at a bitrate of approximately 4 Mbps. It should be noted that the curve for noskip configuration on Bunny is steeper than the other curves. This might be due to the fact that the Bunny sequence presents fewer flaws than the Dog and Champagne sequences do, and so the compression distortions are more perceptible.

4.5.3 Impact of depth estimation and view synthesis

The experimental results in Section 4.5.2 first highlight the limitations of the configurations based on view synthesis. The results show that the efficiency and quality of the

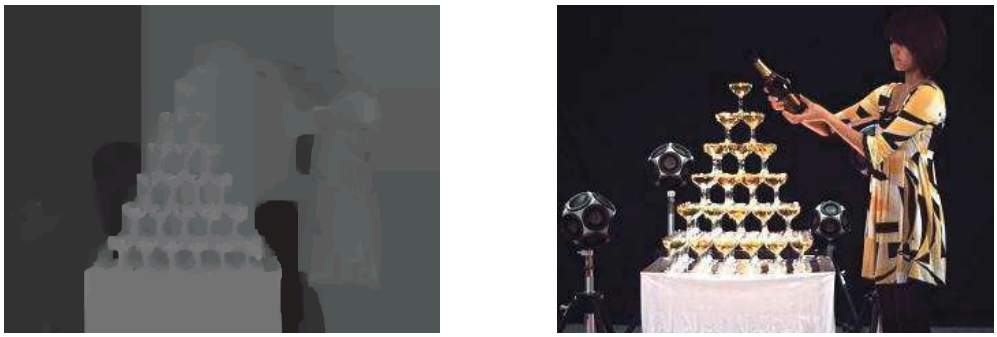


Figure 4.14: Estimated depth map and associated frame for Champagne (view 42)

view synthesis greatly depend on the content of the sequence and on the quality of the associated depth maps. The 3 sequences used for the evaluation are quite representative of this dependency. For Champagne, the estimated depth maps present a lot of flickering from frame to frame, and do not show well-shaped objects (e.g. the lady is mingled with the background for most parts, as illustrated in Figure 4.14). The resulting synthesized 2D views present a lot of synthesis artifacts on objects edges. For Dog, the estimated depth maps are visually better, and the resulting synthesized views present fewer artifacts. The most severe artifacts generally appear only in skip5 and skip9 configurations. For Bunny, estimated depth maps and synthesized views are significantly better than for Dog or Champagne sequences. Even with the skip9 configuration, the synthesis artifacts are rare and hardly perceptible. The main reason might be that Bunny is a computer generated sequence (CG) that has less misalignment issues (camera position, color calibration, capture noise, etc.) between the views, in comparison to a real world captured content (or natural content). This allows the depth estimation and view synthesis algorithms to perform better (see Section 4.4). The depth estimation and view synthesis algorithms performance is dependent on the content as expected. However, in our experiments this inconsistency among sequences goes to an extent where for one sequence (Champagne) it is problematic even to synthesize only one view while for another (Bunny) it is possible to synthesize up to 9 views with a good quality. This shows that we cannot only rely on current depth estimation and view synthesis technologies for SMV video coding because they do not provide sufficient quality for some content.

4.5.4 Range of bitrate values for compressed light-field content

A second conclusion concerns the measured range of bitrate values and associated qualities. In our experiments the minimum bitrate values associated with *Slightly annoying* impairments are approximately 6.6 Mbps for Dog, 9.4 Mbps for Champagne, and 4.5 Mbps for Bunny (respectively with skip1 QP 30, noskip QP 35, and skip1 QP 30). Bitrate values associated to *Perceptible but not annoying* are about 11 Mbps for Dog (with skip1 configuration), less than 10 Mbps for Champagne (with noskip configuration), and about 5 Mbps for Bunny (with skip5 or skip9 configurations). The target bitrate values for encoding 4K content with HEVC are estimated at 10 to 15 Mbps, and 2 or 3 times more for 8K content. Moreover, encoding multi-view content with 3D-HEVC can provide BD-rate gains from 20% to 25% over MV-HEVC in a configuration including 3 coded views (and associated depth maps) and 6 synthesized views. According to these values, the use of SMV content associated with MV-HEVC/3D-HEVC based encoding appears realistic for

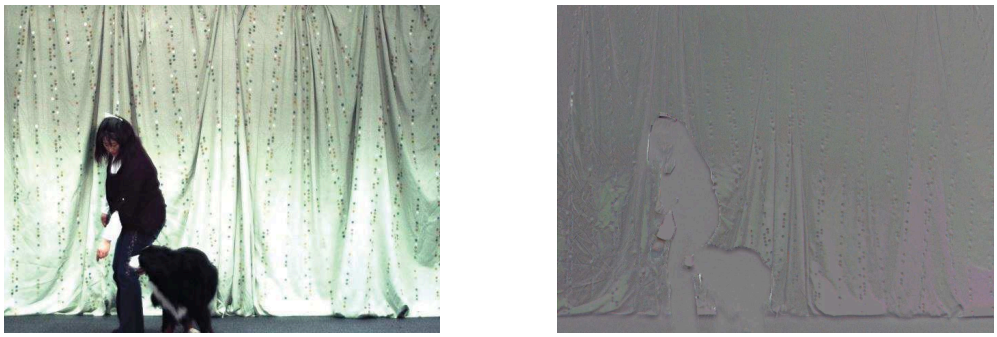


Figure 4.15: Synthesized view and residual synthesis artifacts (skip1, Dog)

future light-field video services and broadcast. This conclusion cannot be considered as definitive because it is limited by the conditions of our experiments which largely depend on the characteristics of the display (like spatial and angular resolutions, field of depth, etc.) and of the tested content (resolution, camera arrangement, etc.). However these results provide a significant first hint on the feasibility of the light-field video using SMV with current compression technologies.

4.5.5 Comparison between objective and subjective results

During the experiments we observed that some of the compression artifacts and synthesis artifacts are generally observable in the same way on the light-field display as they are when visualizing the 2D views separately, e.g. the typical compression block artifacts have the same recognizable aspect. Hence at this point, the experiments do not show any reason to prevent the measure of the quality for SMV light-field content by measuring the objective quality of the input views.

In Section 4.4, we provide objective results by comparing synthesized views against original views at the same position. The comparison of the performance for the different synthesis configurations (noskip, skip1, etc.) is not identical in the objective results and in the subjective results. In the objective results, the PSNR decreases as the number of synthesized views in the configuration increases for Dog and Champagne. The subjective results show the same decrease for Champagne, while for Dog, the noskip, skip3 and skip5 curves are very close and the skip3 curve is slightly better. For Bunny, the noskip, skip1 and skip3 curves are very close in the objective results and skip5 and skip9 are lower, while in the subjective results skip5 and skip9 are better. The PSNR is severe with synthesis artifacts that are not (or hardly) perceptible and do not impact the subjective quality. Figure 4.15, Figure 4.16, and Figure 4.17 show the residual images obtained by subtracting a view (#41) synthesized from the original uncompressed views (and depth maps) and the original view at the same position. Hence these captions only show the artifacts due to view synthesis. Table 4.11 shows the PSNR for views synthesized from uncompressed views computed against the original views at the same positions. These PSNR values (between 24.8 dB and 44.4 dB) are already impacted by the impairments due to the synthesis. PSNR is generally relevant for a given coding configuration, but not always with inter-configurations comparisons. Despite the limited number of points in our experiments, this aspect is coherent with the results provided.

Table 4.12 shows the Spearman and Pearson correlation coefficients [119] for the MOS and PSNR obtained on all configurations and sequences, with values of approximately 0.6

Sequence	Configuration	PSNR (dB)		
		Y	U	V
Dog	skip1	33,1	40,0	39,4
	skip3	31,8	39,9	39,0
	skip5	30,2	39,6	38,7
	skip9	27,4	39,4	38,4
Champagne	skip1	32,0	39,8	39,0
	skip3	29,4	39,1	38,3
	skip5	27,6	38,6	37,9
	skip9	24,8	37,2	36,9
Bunny	skip1	44,4	43,9	45,9
	skip3	38,7	42,8	45,9
	skip5	35,3	42,0	45,5
	skip9	31,9	40,3	44,1

Table 4.11: PSNR of uncompressed synthesized views

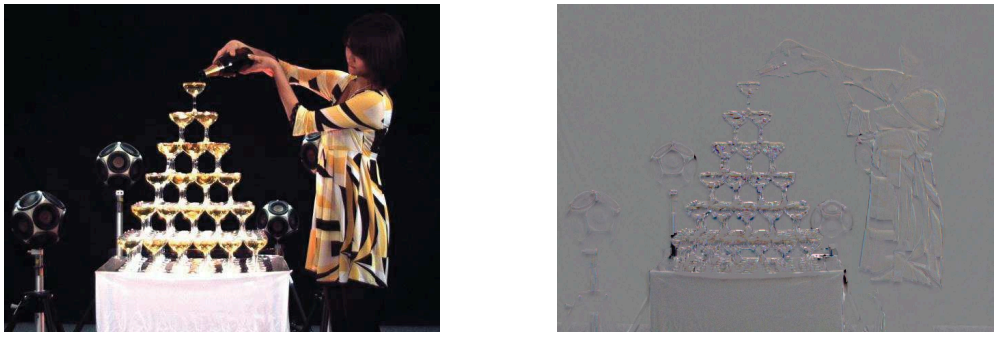


Figure 4.16: Synthesized view and residual synthesis artifacts (skip1, Champagne)

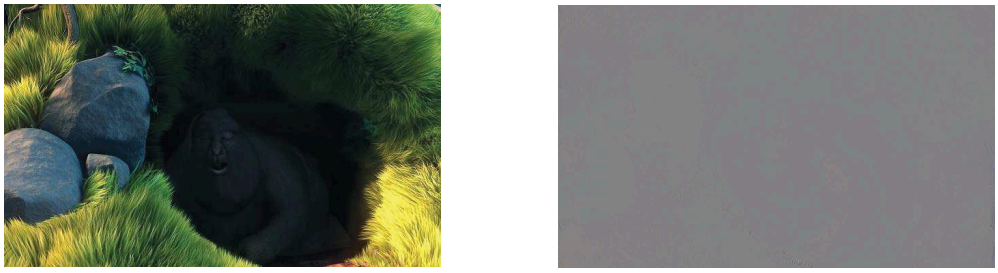
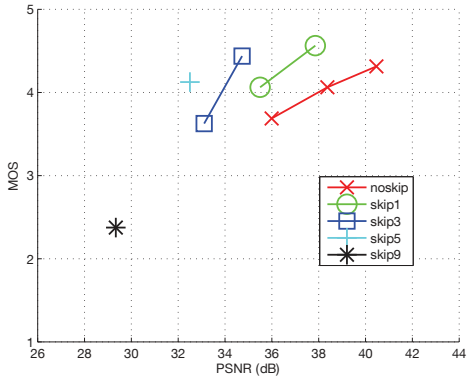


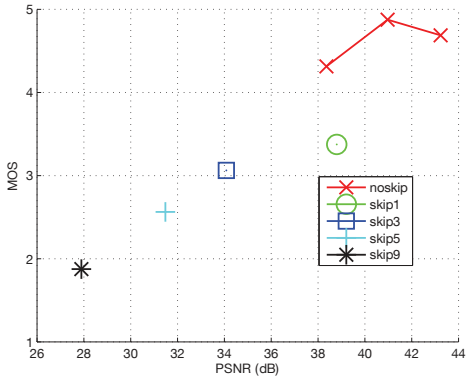
Figure 4.17: Synthesized view and residual synthesis artifacts (skip1, Bunny)

	Spearman	Pearson
Coeff	0.64	0.73

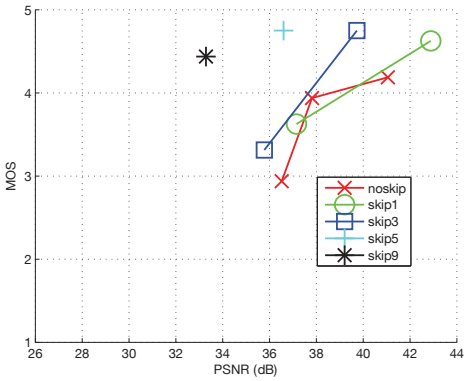
Table 4.12: Correlation coefficients between MOS and PSNR (on all sequences and tested configurations)



(a) Dog



(b) Champagne



(c) Bunny

Figure 4.18: MOS vs. PSNR

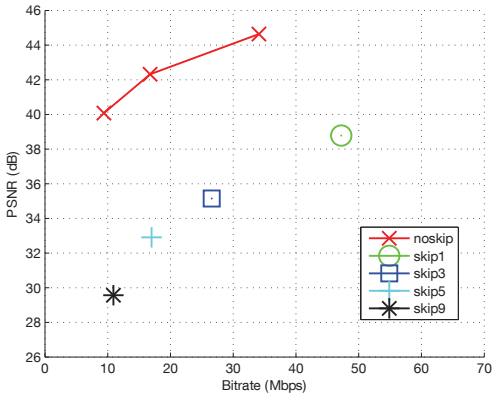


Figure 4.19: Bitrate variations with PSNR measured on light-field slices (Champagne sequence)

and 0.7 respectively, which suggest a correlation between the two variables (the correlation increases as the coefficients absolute value get closer to one). However, this correlation remains significantly smaller than in other mainstream video coding applications. Figure 4.18 plots the MOS relatively to the PSNR associated with each tested configuration. The curves are ascending functions, which shows that the PSNR is able to reflect the increase in the effective quality (even in the presence of skipped views). However, the curves also show the inconsistency of the relation between PSNR and MOS among configurations. For Dog sequence, the MOS value 4 (associated to a score where the impairments are *Perceptible but not annoying*) approximately matches: 33dB with 5 skipped views, 34dB with 3 skipped views, 35.5dB with 1 skipped views, and 38dB without skipped views. This is mainly due to the inefficiency of the PSNR metric for synthesized views (i.e. the PSNR is too severe with synthesis artifacts). However for the noskip configuration, the curves of the 3 sequences cross the MOS value 4 at a PSNR of approximately 38 dB. Hence for this configuration, the PSNR values might be aligned to the MOS values. There is a consistency across sequences in the relation between PSNR and MOS for the configuration without synthesis only, and PSNR is able to reflect an increase of the effective quality. However the order of magnitude of the effective quality variation is biased by the PSNR and changes for the different configurations.

4.5.6 Impact of the light-field conversion step

In this section, we compute the PSNR of the light-field slices (see Section 4.2.2 about light-field conversion) converted from compressed input views against the light-field slices converted from the original uncompressed views. Figure 4.19 shows that the PSNR results on light-field slices for the Champagne sequence are consistent with the PSNR results obtained on the input views (see Figure 4.6). The range of PSNR values are different but relatively close, and the order of the configurations is very similar. The conversion step in our experiments conditions does not seem to have a large impact on the compression and synthesis artifacts, hence on the objective quality of the sequence.

4.5.7 Comments on motion parallax

The effect of compression and synthesis on the motion parallax quality is discussed in this section. It should be noted that this is just based on a preliminary observation (based on one subject's comments). During one session, the subject watched the content while moving on a baseline of approximately 2 meters from left to right and right to left and commented the following aspect of the motion parallax. For compressed sequences which present many artifacts (i.e. with a low quality), a variation of the intensity of these artifacts (e.g. sizes of the blocks artifacts) has been observed when moving along the viewing angle of the display. For the sequences with only few artifacts (with impairments rated as *Imperceptible* or *Perceptible but not annoying*), the variations were not perceptible and did not disturb the perception of the motion parallax. As a first preliminary conclusion, it could be said that the perception of motion parallax is unsatisfying only when the image quality (in terms of compression artifacts and flickering synthesis artifacts) is already bad. More tests (by defining a scale rating the motion parallax from perfectly smooth to jerky for example) should be conducted to confirm these first hints.

4.6 Conclusion

The study presented in this chapter provides some initial conclusions on the feasibility of a video service that would require rendering about 80 views. We have observed that bitrates associated to impairments rated as not annoying are about 11 Mbps for the sequence Dog (with skip1 configuration), less than 10 Mbps for Champagne (with noskip configuration), and about 5 Mbps for Bunny (with skip5 or skip9 configurations). It is known that typical bitrates for encoding 4K content with HEVC are estimated to 15 Mbps and up to 80 Mbps for 8K content. We consequently conclude that bitrates required for rendering 80 views are realistic and coherent with future 4K/8K needs. In order to further improve the quality and avoid network overload, improved SMV video codec efficiency is mandatory. It should also be noted that experiments results largely depend on the characteristics of the display (like spatial and angular resolutions, field of depth, etc.) and on the tested content (2 natural and 1 synthetic sequences) which we do consider as *easy to encode* contents because they contain still backgrounds, have small resolutions and frame-rates (1280×960, 30fps), and have a linear camera arrangement. This note does not change the feasibility conclusion, yet highlights the need for a better codec.

Preliminary experiments performed during this study lead to recommended coding configuration for SMV contents. In particular, IPP inter-view prediction structure with Groups of Views (GOVs) of size 16, with hierarchical temporal prediction structure with GOPs of size 8 is suggested. IPP inter-view prediction structure is more efficient than Central (with 5% BD-rate gains reported) and Hierarchical (3% BD-rate gains reported) structures. Results are similar when the coding scheme includes view synthesis. GOVs of size 16 bring about 3% coding improvement over size 9. GOVs enable a compromise between memory limitations, coding efficiency and parallel processing.

Some conclusions are also drawn on the number of frames to skip at the encoder, and synthesize at the renderer after the decoding process. Several ratios for the number of coded and synthesized views are compared in our experiments. Subjective results suggests skipping 0, 1, 3 or 5 views for Dog, not skipping any view for Champagne, and skipping up to 5 or 9 views for Bunny. The amount of views to skip is highly sequence dependent, and varies from 0 to 9 (i.e. the minimum and maximum tested values). The

ratio coded/synthesized depends on the quality of the synthesized views, hence is linked to the quality of the depth maps and the efficiency of the view synthesizer. It obviously also depends on the complexity of the scene that needs to be synthesized.

By synthesizing intermediate views from original uncompressed views, a 25dB to 44dB PSNR is achieved (against the original uncompressed views). Apart from compression, view synthesis introduces severe distortions. View synthesis weaknesses affect the coding scheme and are tightly linked to the estimated depth maps quality. Improvement of view synthesis and depth estimation algorithms is mandatory. The curves representing the correspondence between PSNR and MOS are monotone increasing functions, which shows that the PSNR is able to reflect increase or decrease in subjective quality (even in the presence of skipped views). However, depending on the ratio of coded and synthesized views, we have observed that the order of magnitude of the effective quality variation is biased by the PSNR. PSNR is less tolerant to view synthesis artifacts than human viewers.

Finally, preliminary observations have been initiated. First, the light-field conversion step does not seem to alter the objective results for compression. Secondly, the motion parallax does not seem to be impacted by specific compression artifacts. The perception of the motion parallax is only altered by variations of the typical compression artifacts along the viewing angle, in cases where the subjective image quality is already low (i.e. cases with severe artifacts).

As mentioned above, the experiments depends on our test conditions and particularly on the tested content. As a consequence, future work should extend the evaluation towards additional content with different depth characteristics and encoding complexities. It should be noted that producing Super Multi-View content is not a trivial task and has been one of the main issue for the research community working on this technology. For example, one of the main tasks of the FTV ad-hoc group in MPEG is to gather content [120].

The study should also be extended to content captured with different camera arrangements, like the arc arrangement which is generally considered more appropriate for light-field display systems, but cannot be handled properly by current view synthesis and depth estimation algorithms, and is reported to provide less efficient coding performance with current multi-view encoders [112].

Further experiments could complete current results. Subjective evaluations with a denser range of bitrate values could allow refining the boundaries between the ranges of bitrate values associated with each quality level. Similarly, a lower range could allow determining the lowest bitrate value possible for an acceptable quality.

Using these denser ranges and limit values could allow finding a proper way to evaluate objectively the quality of compressed and synthesized SMV content by weighting efficiently the PSNR for synthesized views or by using a more convenient metric. This could allow associating ranges of subjective qualities with ranges of objective values.

The impact of the compression and synthesis artifacts on the perception of motion parallax should be further studied, as well as other specific aspects of light-field content such as the perception of depth or the angle of view for example.

Acknowledgment

This work has been carried out in the context of a Short Term Scientific Mission (STSM) granted by the COST Action 3D-ConTourNet [121]. The authors want to thank the

STSM coordinator, the Action Chair and the Management Committee of COST Action 3D-ConTourNet.

The research leading to these results has received funding from the PROLIGHT-IAPP Marie Curie Action of the People programme of the European Union's Seventh Framework Programme, REA grant agreement 32449, and from the DIVA Marie Curie Action of the People programme of the European Union's Seventh Framework Programme, REA grant agreement 290227.

Dog, Pantomime, and Champagne Tower (Nagoya University's sequences) are provided by Fujii Laboratory at Nagoya University [108]. T-Rex sequence is provided by Holografika [14]. Big Buck Bunny is copyright Blender Foundation [122], and 3D camera setup and rendering is done by Holografika [109].

Chapter 5

Full parallax super multi-view video coding

5.1 Introduction

Motion parallax is defined as the "optical change of the visual field of an observer which results from a change of his viewing position" [123], and corresponds to the "set of apparent motions of stationary objects which arise during locomotion". In other words, in the case of multi-view imaging, it corresponds to the different disparity displayed by objects depending on their depth (the closer the object, the larger the disparity). It is a psychological cue that allows the user to perceive depth, hence to gather more information about the scene when changing point of view (e.g. using interactive request in free navigation applications or just by moving in front of the screen for glasses-free light-field display systems).

Smooth motion parallax is therefore considered as a key cue in the perception of depth for natural immersive applications. The lack of smooth motion parallax in current 3D video technologies available on the consumer market particularly alters the quality and comfort of visualization [1], as the visualization is not continuous when moving in front of the display.

Super Multi-View (SMV) video with dense enough camera arrays theoretically allows a realistic visualization with a smooth motion parallax in horizontal and potentially vertical directions. However, most ligth-field display systems currently omit the vertical parallax in order to provide a better horizontal angular resolution. One consequence of not including parallax for both axes appears when showing objects increasingly distant from the plane of the display. As the viewer moves closer to or further away from the display, such objects exhibit the effects of perspective shift in one axis but not in the other one, appearing variously stretched or squashed to a viewer that is not positioned at an optimal distance from the display.

Multi-view encoder extensions are adequate to encode SMV content with horizontal parallax only. Modifications of these encoders have been proposed in the literature to encode content with full parallax. State-of-the-art methods present however limitations in the use of the two dimensions for inter-view predictions. Here we propose an efficient inter-view prediction scheme to exploit horizontal and vertical dimensions at the coding structure level. Then we propose improvements of inter-view coding tools to exploit the two dimensional structure also at the coding unit level.

This chapter is organized as follows. Section 5.2 describes state-of-the-art methods for full parallax SMV encoding. In Section 5.3, we propose an inter-view reference picture scheme and show experimental results against state-of-the-art schemes. Improved inter-view coding tools adapted to full parallax are proposed in Section 5.4, including experimental results. Section 5.5 finally concludes the chapter.

5.2 State-of-the-art

5.2.1 Multi-view video coding standards and specific coding tools

As described in Chapter 2, SMV defines multi-view video content with tens or hundreds of views, with either horizontal only or full motion parallax. The massive number of views increases the amount of data to process compared to current 3D video technologies. The amount of inter-view correlation is also increased. Current multi-view encoders have been designed for horizontal parallax content with limited number of views. MVC and MV-HEVC are the multi-view extensions of respectively H.264/AVC and HEVC standards [19]. These extensions provide additional high level syntax that allows the inter-view prediction. 3D-AVC [124] and 3D-HEVC [125] extensions provide depth-related tools and new tools at macroblocks/CU level (respectively) for side views.

In the reference software used in the experiments described in this chapter (HTM7.0), the following applies: Neighboring Block Disparity Vector (NBDV) [126] and Inter-View Motion Prediction (IVMP) [127] are specific 3D-HEVC coding tools designed for standard horizontal multi-view encoding. For the current CU, NBDV searches for a disparity vector (DV) through already coded temporal and spatial neighboring CUs (illustrated in Figure 5.1, with A0 corresponding to bottom-left, A1 to left, etc., and Col corresponding to the collocated CUs in the temporal reference frame). The DV derived by NBDV is used by IVMP to create the Inter-View Motion Candidate (IVMC). IVMC corresponds to the motion parameters (motion vectors and temporal references) of the CU pointed by the DV in the reference view. This process is shown in Figure 5.2. IVMC is introduced at the first place in the merge [25] candidate list (for textures). Finally the DV itself is inserted in the merge list as Inter-View Disparity Candidate (IVDC). The list of merge candidates is thus composed as follows: Texture (inherited only for depth coding), IVMC, A1, B1, VSP (for View Synthesis Prediction), B0, IVDC, A0, B2, Shifted IVMC, Shifted IVDC, and remaining candidates (collocated, combined, zero default) [128].

5.2.2 Improvement for full parallax configuration

The first approach considered to encode full parallax SMV content is the use of a multi-view encoder with an adaptation at the inter-view references structure level. In [79], the views are first scanned in spiral as illustrated in Figure 5.3 (a) and realigned horizontally. Then the horizontal arrangement is coded using a IBP prediction structure (b) by an MVC encoder. Figure 5.3 (c) shows the resulting scheme of equivalent IBP structure with the views represented in two dimensions. The main drawback of this approach is the introduction of unsuitable predictions, i.e. random predictions without any logical reason to be and that are generally ineffective.

In [129], it is proposed to apply horizontal IPP or IBP structures (Fig. 5.6(e) and (f)) to each line of the views array, and to add vertical inter-view prediction only for the first or central column of views as illustrated in Figure 5.4 (a),(b) and (c). The number of

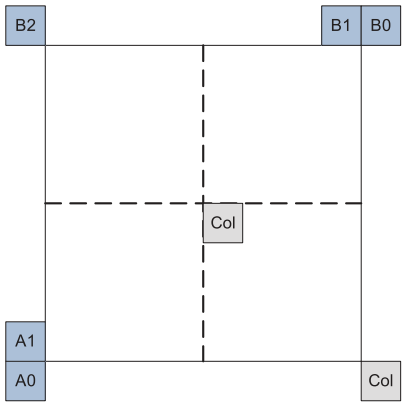


Figure 5.1: Positions accessed in NBDV or merge mode

available vertical inter-view predictions is very limited in such structures.

In [130], [131] and [132], another structure is proposed as illustrated in Figure 5.4 (d). Each line of views uses an horizontal IBP structure and additional vertical inter-view predictions are introduced, giving views of types B1 with two horizontal or vertical only references, B2 with one horizontal and two vertical references, and B3 with two references in both directions. The number of views that use both horizontal and vertical references is limited (less than half of the views are of types B2 or B3) and the distance between the coding and reference views can be large.

A second approach at the coding unit level is considered in [133] and in [130], [131] and [132]. Similar methods are proposed based on the prediction of a DV for the current coding view by interpolation of DVs from neighboring views.

5.3 Proposed inter-view reference pictures configuration

5.3.1 Reference and proposed schemes

Here the goal is first to improve the compression efficiency with non-normative modifications, i.e. using the compression standard as is, only with a new configuration. We propose a two dimension inter-view reference picture structure, *Central2D*, that can ex-

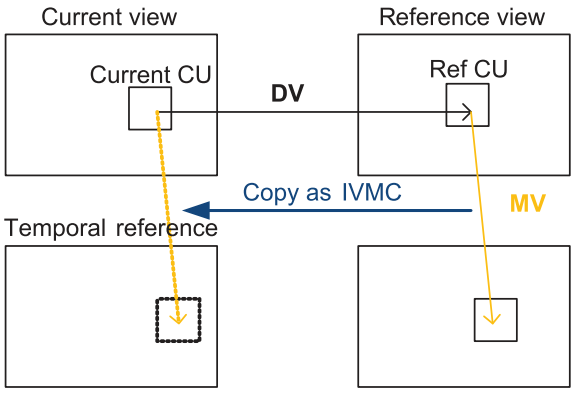


Figure 5.2: Inter-View Motion Prediction (IVMP)

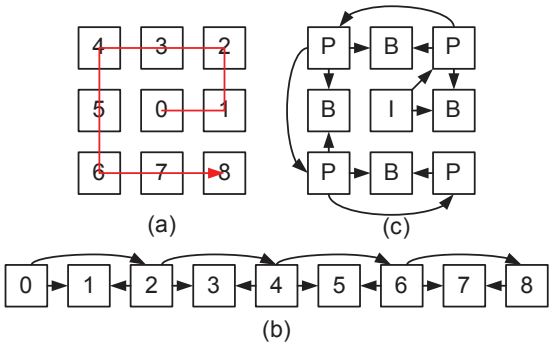


Figure 5.3: State of the art method [79] for 9 views (0...8) (a) spiral scan, (b) IBP structure for inter-view prediction, (c) equivalent IBP scheme in 2 dimensions

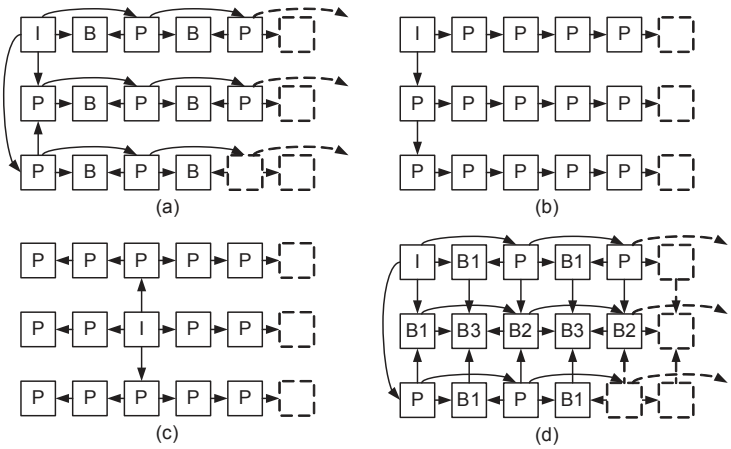


Figure 5.4: State of the art structures: (a),(b) and (c) proposed in [129], and (d) proposed in [130][131][132]

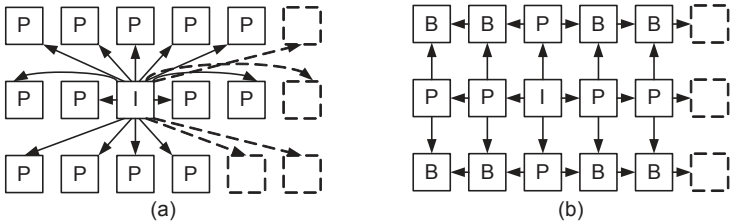


Figure 5.5: (a) basic anchor, (b) proposed *Central2D*

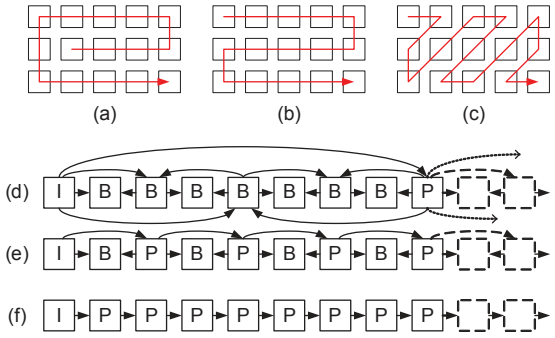


Figure 5.6: Scan orders: (a) spiral, (b) perpendicular, (c) diagonal and Horizontal inter-view reference picture structures: (d) hierarchical, (e) IBP, (f) IPP

ploit efficiently a two dimensional view alignment as illustrated in Figure 5.5 (b). For a $N \times M$ views configuration, *Central2D* scheme is built as follows. The central view is coded first and cannot use inter-view references. The $N-1$ (respectively $M-1$) views that are in the same horizontal (resp. vertical) axis as the central view are then coded using only one inter-view reference, being the nearest view in the central direction. All the other views are coded using one horizontal and one vertical inter-view references being the nearest views in the central direction, hence it allows the use of an horizontal and a vertical inter-view reference picture for a large number of views (only $M+N-1$ views do not have horizontal and vertical reference pictures). Moreover this method minimizes the distance between the coding views and their inter-view reference pictures and does not use diagonal references.

Several inter-view prediction structures have been proposed or discussed in the literature (see Sec. 5.2), but compression performance results have not always been provided. Therefore in the following section, the proposed scheme is compared to a basic anchor (see Figure 5.5 (a)) with only the central view as inter-view reference picture for all the other views, in order to fairly assess the benefit of inter-view prediction in two directions and of a small distance between the coding and the reference views. Aforementioned state-of-the-art structures are also tested in our experiments: [129] and [130] correspond to the schemes illustrated in Figure 5.4 (c) and (d). [79] corresponds to the spiral scan with IBP structure (see Figure 5.3). For comparison purpose, we also propose to extend method [79] by modifying the scan order and the structure as illustrated in Figure 5.6.

Coast 3×3			
	spiral	perpendicular	diagonal
IPP	-1.2%	-2.2%	5.1%
IBP	9.1%	7.1%	11.4%
Hierarchical	3.0%	4.4%	8.4%
Method [130]	2.1%		
Method [129]	-6.8%		
Central2D	-7.1%		
Akko 3×3			
	spiral	perpendicular	diagonal
IPP	-4.9%	-5.5%	8.8%
IBP	2.7%	-4.0%	-1.9%
Hierarchical	1.9%	2.4%	4.0%
Method [130]	7.8%		
Method [129]	-7.7%		
Central2D	-8.2%		

Table 5.1: BD-rate variations for state of the art and proposed structures compared to basic anchor - with 3×3 views

5.3.2 Experimental results

In this section, we test the state-of-the-art and proposed schemes within MV-HEVC. The temporal prediction structure is as described in the Common Test Conditions (CTC) of 3D-HEVC [134]. Experiments are performed under MV-HEVC reference software version 7.0 (HTM7.0 with QC_MVHEVC macro). Two sequences are tested: *CoastalGuard* (50 frames, computer generated, resolution 768×384) and *Akko&Kayo* (290 frames, captured, resolution 640×480). A first configuration with only 3×3 views is tested, and in second step 11×5 views are used to assess the impact of a large number of views on the performance variations. Results are measured using the Bjøntegaard Delta (BD) rate [97] on the QPs range 22-27-32-37. The reference is the basic anchor scheme (Figure 5.5 (a)). Negative values represent improvement over the anchor.

Table 5.1 shows that for both sequences with a 3×3 views configuration, the *Central2D* scheme, method [129] and IPP structure with perpendicular and spiral scan outperform the other methods. These schemes do not use diagonal inter-view reference pictures and minimize the distance between the coding views and the inter-view reference pictures. The extra gain for *Central2D* is due to the use of both horizontal and vertical inter-view reference pictures. Table 5.2 shows that *Central2D* remains the most coherent and efficient configuration with a larger number of views.

The final BD-rate gain for the proposed structure *Central2D* against the basic anchor is up to 8.2% and 29.1% in the 3×3 and 11×5 views configuration respectively. These results show that it is possible to significantly improve the compression performance for full parallax SMV content only with non-normative configuration changes, that adequately exploit the horizontal and vertical prediction directions while remaining compatible with the state-of-the-art codec and standard.

Coast 11 × 5			
	spiral	perpendicular	diagonal
IPP	-20.5%	-19.6%	16.1%
IBP	-15.9%	-14.9%	-13.9%
Hierarchical	-8.4%	-9.3%	-13.0%
Method [130]	-19.5%		
Method [129]	-24.4%		
Central2D	-29.1%		
Akko 11 × 5			
	spiral	perpendicular	diagonal
IPP	-22.9%	-24.8%	-6.5%
IBP	-20.0%	-23.4%	-2.4%
Hierarchical	-14.9%	-20.2%	-3.7%
Method [130]	-24.2%		
Method [129]	-25.9%		
Central2D	-27.6%		

Table 5.2: BD-rate variations for state-of-the-art and proposed structures compared to basic anchor - with 11 × 5 views

5.4 Adaptation and improvement of inter-view coding tools

5.4.1 Merge candidate list improvement

NBDV and IVMP are specific coding tools implemented to work in the Common Test Conditions [134], i.e with only one horizontal inter-view reference picture, which is the central baseview (with view index 0). This is not efficient for full parallax configurations, therefore in this section we adapt these tools by allowing the use of several inter-view reference pictures with a view index different from 0, and possibly horizontal or vertical. Furthermore, in order to exploit the redundancies from both horizontal and vertical prediction directions, we propose a normative modification of the NBDV and IVMP coding tools.

We improve NBDV as follows. When encoding one of the B views that use one horizontal and one vertical inter-view reference pictures, the modified NBDV searches for two DVs (one for each inter-view reference picture). The search of a second DV does not provide BD-rate gain in itself but will be used for IVMC and IVDC. The new second DV is used to introduce a second IVMC at the second place of the merge candidate list. For the IVDC merge candidate, the couple of DVs is used, allowing an inter-view bi-prediction in both directions at the same time.

5.4.2 Inter-view derivation of the second DV

We propose to increase the chances of finding a second DV with NBDV in order to improve the efficiency of modified IVMC and IVDC candidates. The steps are illustrated in Figure 5.7. For the current coding view NBDV must find a first horizontal DV pointing a reference CU in an inter-view reference picture. If this horizontal reference picture uses itself a vertical inter-view reference picture and if the reference CU is coded by inter-view prediction, the vertical DV used for the prediction is inherited/derived as a second DV for the current coding CU, and then used by IVMC and IVDC as described in the previous section. We note that this method can be used for B views with one horizontal and

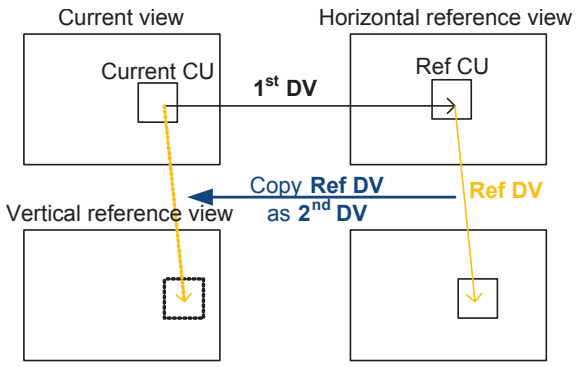


Figure 5.7: Inter-view derivation of a second DV

one vertical references, which makes the *Central2D* structure the most adequate for this coding tools.

5.4.3 Experimental results

In this section, we test the proposed modifications on NBDV and IVMP coding tools. The experiments are performed under 3D-HEVC reference software version 7.0 (HTM7.0). The test conditions are the same as in Sec. 5.3.2, (i.e. allowing two dimensional structures configuration). Previously proposed *Central2D* structure is used in all following experiments. The reference is HTM7.0 without software modifications.

Table 5.3 shows the results for the following proposed modifications: adaptation of NBDV and IVMP to a two dimensions structure (adaptation only), bi-prediction for the IVDC merge candidate (BiIVDC), insertion of a second IVMC in the merge candidate list (2 IVMC), and the combination of both tools (adaptation with BiIVDC and 2 IVMC). Table 5.4 shows the results for BiIVDC and 2 IVMC, also both separately and combined, with the proposed inter-view derivation for the second DV.

Table 5.3 shows that the adaptation of NBDV and IVMP to a two dimensions structure brings BD-rate gains up to 3.3%, confirming the impact of the use of horizontal and vertical dimensions at the inter-view references structure level. The insertion of a second IVMC in the merge candidate list and the bi-prediction for the IVDC merge candidate separately increase the gains up to 2.4% for the 3×3 views configuration and 3.7% for the 11×5 configuration. The combination of both improvements provides a gain up to 2.5% and 3.9% respectively with 3×3 and with 11×5 views. The results for the combination of both tools are slightly higher than the sum of each taken separately because the bi-prediction allows NBDV to find more often a second DV, hence increasing the chances to have a relevant second IVMC candidate.

Table 5.4 shows that the proposed derivation for the second DV is efficient and increases the encoding performance of the complete proposed method (including the adaptation of NDBV and IVMP to a full parallax structure, the two IVMC, the IVDC bi-prediction and the inter-view derivation of the second DV) up to 2.9% and 4.2% for the sequence *Akko&Kayo* respectively with 3×3 and with 11×5 views.

Reference: 3D-HEVC (HTM7.0 without modifications)				
	3×3 views		11×5 views	
	Coast	Akko	Coast	Akko
Adaptation only	-1.1%	-2.3%	-2.4%	-3.3%
BiIVDC	-1.2%	-2.4%	-2.7%	-3.7%
2 IVMC	-1.1%	-2.3%	-2.8%	-3.5%
Combination	-1.3%	-2.5%	-3.1%	-3.9%

Table 5.3: BD-rate variations for improved NBDV and IVMP using one DV for each inter-view reference picture

Reference: 3D-HEVC (HTM7.0 without modifications)				
	3×3 views		11×5 views	
	Coast	Akko	Coast	Akko
BiIVDC	-1.9%	-2.9%	-3.4%	-3.9%
2 IVMC	-1.3%	-2.4%	-2.8%	-3.5%
Combination	-2.0%	-2.9%	-3.9%	-4.2%

Table 5.4: BD-rate variations for improved NBDV and IVMP using one DV for each inter-view reference picture, with inter-view derivation of the second DV

5.5 Conclusion

In this chapter we propose an inter-view reference picture structure adapted to SMV light-field video content with full motion parallax (horizontal and vertical view alignment). Its main features are the minimal distance between the coded and the reference views, and the use of both horizontal and vertical inter-view references. The proposed scheme outperforms a basic anchor by up to 29.1% (BD-rate gains), showing the impact of an efficient use of both horizontal and vertical directions in the inter-view reference picture scheme. We also propose to improve 3D-HEVC coding tools NBDV and IVMP in order to exploit both horizontal and vertical directions in a full parallax configuration, providing BD-rate gains up to 4.2%. The results of the proposed methods show that exploiting efficiently both horizontal and vertical dimensions of full parallax SMV content at the coding structure and coding tools level significantly improves the compression performance.

Acknowledgement

The Coast sequence is provided by Orange Labs. The Akko&Kayo sequence is provided by Fujii Laboratory at Nagoya University.

Chapter 6

On the interest of arc specific disparity prediction tools

6.1 Motivations

According to [107], for light-field displays with extremely wide FOV, equidistant linear camera arrays cannot capture the visual information necessary to represent the scene from all around, and arcs of cameras are more suitable in this case. This difference is illustrated by the scheme in Figure 6.1. However, coding gains from inter-view prediction are reported to be smaller for arc than for linear camera arrays [112]. Therefore arc content appears to be better for display but challenging for existing coding tools. In this chapter we assess how arc camera arrangements impact on the compression performance. Results show no significant performance difference between arc and linear on the test set used in the proposed experiments. Hence we propose perspectives to improve specifically the performance in the arc case (without degrading it for the linear case).

6.2 State-of-the-art

6.2.1 Anchor results

An evaluation of compression performances on a circular camera arrangement sequence (*PoznanBlocks*) is reported in [112]. MV-HEVC, 3D-HEVC and simulcast HEVC are compared, and three views are encoded (only the texture without depth). Results report that MV-HEVC provides gains around 10% over HEVC simulcast and that 3D-HEVC gives additional 3%. Gains of MV-HEVC and 3D-HEVC appear therefore smaller than in the case of linear camera arrangements, for example compared to gains that are commonly reported in the literature (e.g. around 30% for MV-HEVC against HEVC in a case with two views, and around 20% for 3D-HEVC against MV-HEVC in a case with 3 views, as discussed in Chapter 1). Content overlap (i.e. redundancies) is generally smaller in arc content, and coding tools in 3D-HEVC are not adapted to such camera setups. However, these results are not provided for the same content, therefore they do not allow to draw direct conclusions. In Section 6.3, we compare the coding efficiency for scenes generated both with arc and with linear camera arrangements, and show that there is no significant difference in performance.

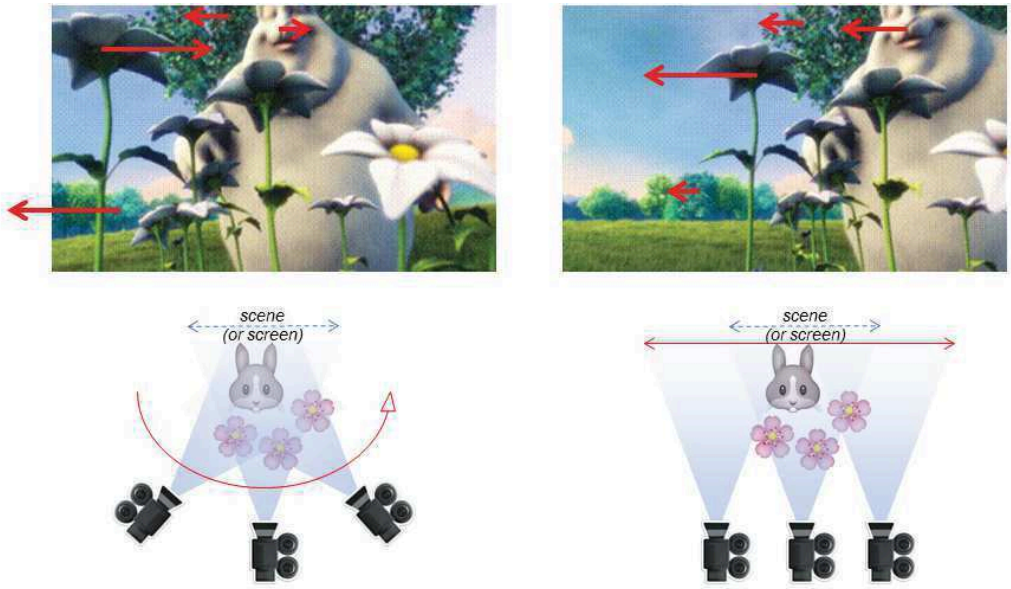


Figure 6.1: Comparison of disparity in arc (left) and linear (right) content

6.2.2 Generalization of 3D-HEVC coding tools

All the coding tools related to depth maps are generalized in [135]. Restrictions concerning view arrangements are removed, support for arbitrary camera location described by camera parameters is enabled, and disparity vectors are calculated by using projection matrices for both the reference view and the view being coded. The modified disparity vector derivation process impacts the following tools: Disparity Compensated Prediction (DCP), Neighboring Block Disparity Vector (NBDV), Depth oriented NBDV (DoNBDV), View Synthesis Prediction (VSP), Inter-View Motion Prediction (IVMP), Illumination Compensation (IC). In addition to the already transmitted (in the current encoder and standard) camera parameters (focal length f_x, optical center c_x of the camera, and position of the camera's optical center along x axis T_x), additional intrinsic parameters (second focal length along vertical direction f_y, position of the optical center along a second axis c_y) and extrinsic parameters (rotation matrix R, and remaining coordinates of camera position T_y and T_z, forming vector T).

An average BD-rate gain of 6% over 3D-HEVC is reported in the case where 3 views and 3 depth maps are encoded. Hence gains of this improved 3D-HEVC over MV-HEVC on arc content reach approximately the gains of 3D-HEVC over MV-HEVC on linear content. This shows that further improvement of the 3D-HEVC encoder is possible by relaxing explicit constraints on camera setups.

6.3 Comparison of coding performances between arc and linear content

In this section, we report experiments performed to reproduce results comparing linear and arc camera arrangements [112]. The arc sequences available for this experiment are listed in Table 6.1. First, the comparison of HEVC, MV-HEVC and 3D-HEVC (as described in Sec. 6.2.1) is performed with 10 views of *PoznanBlocks*. Results reported in Table 6.2 are

Sequence	Resolution	Views	Arrangement	Details
Butterfly	1280×768	80	Arc and Linear	Computer Generated
Flowers				
Rabbit				
Poznan Blocks	1920×1080	10	Arc	Natural

Table 6.1: Available content

3D-HEVC vs. MV-HEVC	3D-HEVC vs. HEVC	MV-HEVC vs. HEVC
-2%	-13%	-11%

Table 6.2: Comparison of HEVC simulcast, MV-HEVC, and 3D-HEVC (*PoznanBlocks* – 10 views)

similar to the results given in [112]. The same comparison is also performed on the *Bunny* sequences (i.e. *Butterfly*, *Flowers*, and *Rabbit*, Computer Generated with 80 views), and the results between arc and linear camera arrangements are also compared. Sequences are encoded with 80 views for the first case, then views are skipped in order to assess the impact of the number of views, and of the distance between these views, and to simulate cases with synthesis. As shown in Table 6.3, results are very close between arc and linear camera arrangements. This is different from the result expected from the conclusions drawn in [112]. However, in [112] the results are obtained only on one sequence captured with an arc camera arrangement, and compared to results obtained on other contents captured with a linear camera arrangement. The main goal of our experiment is to compare the coding performance on the same content/scene captured with both types of arrangement. The fact that the sequences in our test set are computer generated allows to compare directly arc and linear arrangements. In our experiment, gains for MV/3D-HEVC are smaller when the number of views decreases (sparser views with larger distance between cameras), but there is still no significant difference between arc and linear. This behavior remains even with a number of views decreased down to 10 views. It should be noted that with 5 views, there is an unexpectedly large gain for the arc case on *Butterfly*. This might be explained by the fact that this sequence is quite easy to encode and predict because of the very large and homogeneous sky background. Moreover in the case with 5 views only, content overlap between views is much smaller because of the distance, hence the results appear less consistent compared to the other cases.

6.4 Analysis of the content

6.4.1 Disparity in arc content

The disparity is different in arc content and in linear content as illustrated in Figure 6.1. In linear content, disparity is always in the same direction (i.e. exclusively left or right), and the length of the vector depends on the distance from the camera: the closer the object, the larger the disparity. In arc content, disparity can be directed to the left or to the right, depending on the position of the object (in front or behind) regarding to the convergence center of the cameras, and the length depends on the distance from this convergence point: the further the object, the larger the disparity. This is due to the fact that disparity is caused by the translation of the camera in both arc and linear content, but also by the rotation of the camera in the case of arc content.

	MV-HEVC vs. HEVC		3D-HEVC vs. HEVC		
	Arc	Linear	Arc	Linear	
Butterfly	-74%	-76%	-79%	-80%	80 views
Flowers	-49%	-53%	-58%	-62%	
Rabbit	-76%	-74%	-76%	-75%	
Average	-66%	-68%	-71%	-72%	
Butterfly	-64%	-66%	-70%	-71%	40 views
Flowers	-37%	-41%	-45%	-50%	
Rabbit	-63%	-61%	-64%	-63%	
Average	-55%	-56%	-60%	-61%	
Butterfly	-52%	-53%	-59%	-60%	20 views
Flowers	-26%	-30%	-33%	-37%	
Rabbit	-46%	-43%	-47%	-45%	
Average	-41%	-42%	-46%	-47%	
Butterfly	-35%	-36%	-43%	-43%	10 views
Flowers	-14%	-18%	-19%	-23%	
Rabbit	-26%	-24%	-27%	-27%	
Average	-25%	-26%	-29%	-31%	
Butterfly	-17%	-8%	-22%	-9%	5 views
Flowers	-4%	-7%	-6%	-9%	
Rabbit	-9%	-7%	-11%	-9%	
Average	-10%	-8%	-13%	-9%	

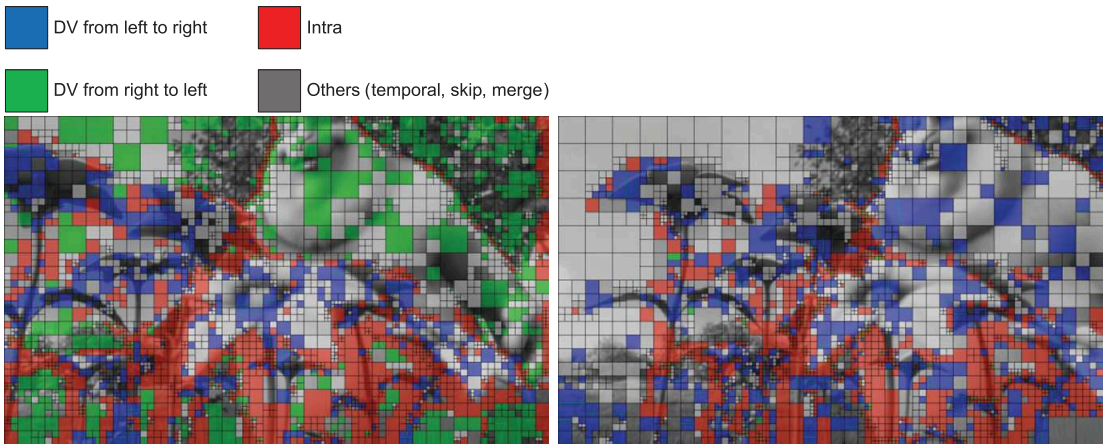
Table 6.3: Comparison of HEVC simulcast, MV-HEVC, and 3D-HEVC

In this section we assess the impact of this specificity of arc content on inter-view prediction by displaying the direction of the disparity vectors in a decoded frame. In Figure 6.2 and Figure 6.3, the content is encoded and decoded/reconstructed, and the CUs in the frame are colored as follows according to their coding mode and to the direction of the DVs: red for INTRA, white for SKIP, green for a DV from right to left, blue for a DV from left to right, and black for *others* (INTER with temporal prediction, and MERGE).

It should be noted that the directions (i.e. blue and green colors) are mostly observable on the first encoded frames (i.e. first POCs) because further frames have mostly CUs tagged as SKIP and *others*. Results are as expected concerning the directions. Figure 6.2 shows a frame for view 23. In the arc case there are mostly DVs from left to right on foreground objects (i.e. blue on the flowers), and from right to left on background elements (i.e. green on the rabbit, trees, sky, etc.), while for the linear case there are mostly DVs from left to right (i.e. CUs in blue) and almost no DV from right to left (i.e. CUs in green). For view 63 illustrated in Figure 6.3 the same applies with opposite prediction directions (i.e. opposed colors in the figure) as the reference view is on the other side. Figures 6.4, 6.5, and 6.6 show the same behavior for the two other *Bunny* sequences (i.e. *Butterfly* and *Rabbit*) and for *PoznanBlocks*.

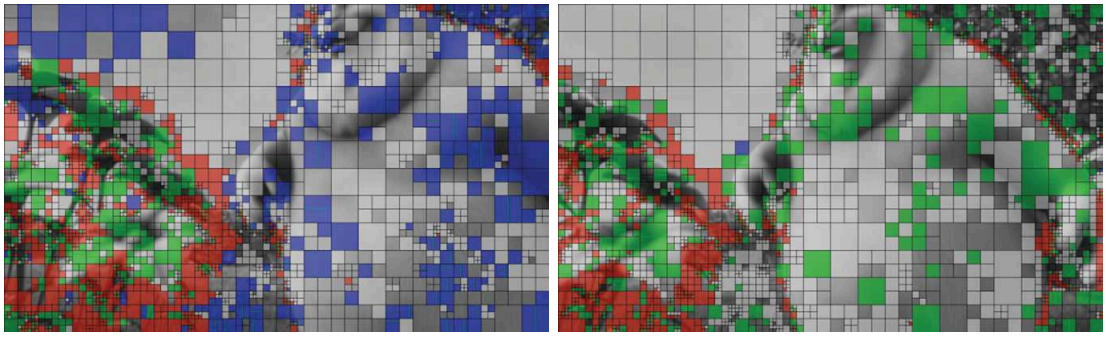
6.4.2 Percentage of the total bitrate dedicated to motion/disparity

Results from Section 6.4.1 hint that the prediction of the DVs can be improved for arc content by taking the opposite directions into account. As mentioned in Section 6.3, there is no significant performance difference between arc and linear on the test set used in these experiments. Hence the goal here is not to allow the arc case to reach the performance of the linear case, as in [135], but to actually improve specifically the performance in the arc case (without degrading it for the linear case).



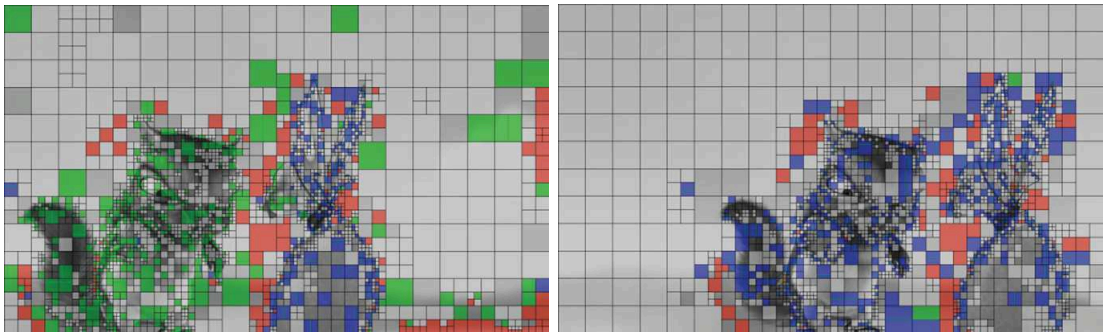
Arc Linear

Figure 6.2: Flowers (View 23, Frame 0, QP 20)



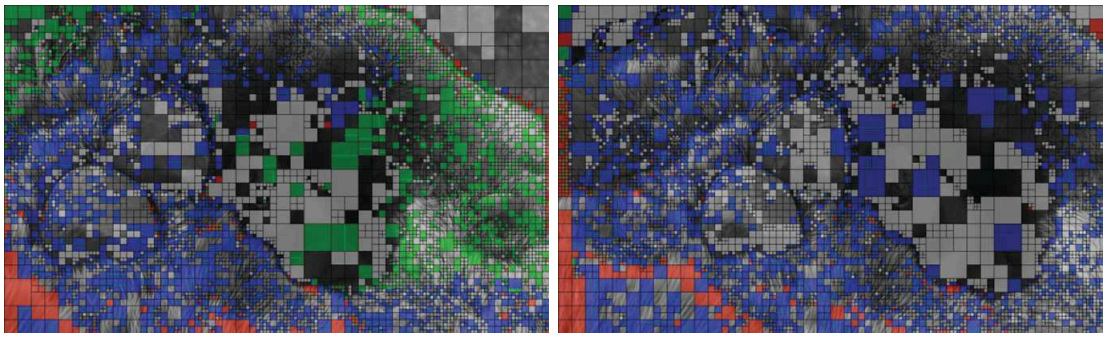
Arc Linear

Figure 6.3: Flowers (View 63, Frame 0, QP 20)



Arc Linear

Figure 6.4: Butterfly (View 23, Frame 0, QP 20)



Arc Linear

Figure 6.5: Rabbit (View 23, Frame 0, QP 20)

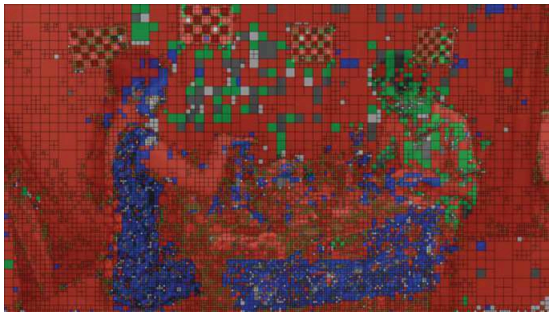


Figure 6.6: PoznanBlocks (Arc, View 2, Frame 0, QP 20)

By improving the prediction of the DVs, one impacts on the encoding of the motion parameters. And because the DVs are predicted from neighboring CUs, the partitioning is also subject to changes. In the following, we study the percentage of the total bitrate dedicated to the syntax elements related to motion/disparity, and to partitioning, i.e. the elements that are supposed to be impacted by the foreseen modifications of the encoder. Table 6.4 and Table 6.5 show the percentage respectively for the partitioning and motion syntax elements of HEVC's reference software in the encoding of *Flowers* at QP 20 from the experiments described in Section 6.4.1. It should be noted that in these tables, elements with names ending by *SC* correspond to elements encoded with CABAC contexts, and others are encoded with equiprobability. The syntax elements taken into account are described in the following:

- PartSize, PartSizeSC and SplitFlagSC describe the size and partitioning of the CUs and PUs (i.e. 64×64, 32×32, etc.)
- MVPIdxSC is the index of the candidate for the advanced prediction of the MV/DV
- MvdSign, MvdAbs and MvdSC describe the MV/DV residual after prediction (i.e. sign and residual value)
- RefFrmIdx, RefPicSC and InterDirSC describe the index of the reference pictures and reference lists
- MergeFlagExtSC, MergeIdxExtSC and MrgIdx describe the use of the merge index and the index of the merge candidate

Syntax element	Arc		Linear	
	Bits/elem	%	Bits/elem	%
SplitFlagSC	3672017	3,5	3311732	3,7
PartSizeSC	2498886	2,4	2112942	2,3
PartSize	191043	0,2	165866	0,2
Total	**6361946** on 104751749	**6,1**	**5590539** on 90673387	**6,2**

Table 6.4: Percentage of the bitrate for partitioning syntax elements - *Flowers* QP20

Syntax element	Arc		Linear	
	Bits/elem	%	Bits/elem	%
MergeFlagExtSC	1928146	1,8%	1597668	1,8%
MergeIdxExtSC	4549767	4,3%	4176510	4,6%
InterDirSC	492855	0,5%	412371	0,5%
RefPicSC	647411	0,6%	533100	0,6%
MvdSC	2243857	2,1%	1806732	2,0%
MVPIdxSC	778442	0,7%	613338	0,7%
MvdSign	1188523	1,1%	936203	1,0%
MvdAbs	2544983	2,4%	2047871	2,3%
RefFrmIdx	9494	0,0%	8154	0,0%
MrgIdx	4183394	4,0%	3698622	4,1%
Total	**18566872** on 104751749	**17,7**	**15830569** on 90673387	**17,5**

Table 6.5: Percentage of the bitrate for motion syntax elements - *Flowers* QP20

These tables provide an average result on all the encoded layers (i.e. views) and POCs. The detailed results by POC/layer are consistent. Percentage values are almost identical between arc and linear. This confirms the conclusions drawn from our BD-rate results that 3D-HEVC and MV-HEVC prediction tools are not significantly altered by the arc arrangement (on this test set). For this sequence and QP, partitioning and motion information represent respectively 6.1% and 17.7% of the bitrate, which is significant. In the following, Table 6.6, Table 6.7, and Table 6.8 provide similar results on other sequences and QP values.

The percentage of the total bitrate for the studied elements increases significantly with QP values, mainly because of MvAbs and MrgIdx. Again, with these other sequences and QP values, results are almost identical between arc and linear cases. Significant differences are reported from one sequence to another (i.e. *Butterfly*, *Flowers*, and *Rabbit* sequences give 9.7%, 17.7% and 23.1% for motion elements at QP 20 in arc). Values are in the same order for *PoznanBlocks*. Results confirm that the target syntax elements are significant in the bitrate, hence hint for potential improvements.

6.5 Proposed methods and preliminary results

In this section, we discuss possible improvements of Neighboring Block Disparity Vector (NBDV) and Advanced Motion Vector Prediction (AMVP) tools that take advantage of

	Arc			Linear		
	Butterfly	*Flowers*	*Rabbit*	*Butterfly*	*Flowers*	*Rabbit*
PartSize	0,3	0,2	0,1	0,3	0,2	0,1
MvdSign	1,6	1,1	0,9	1,6	1,0	0,8
MvdAbs	2,2	2,4	0,7	2,2	2,3	0,7
RefFrmIdx	0,0	0,0	0,0	0,0	0,0	0,0
MrgIdx	4,8	4,0	1,6	4,5	4,1	1,5
SplitFlagSC	3,9	3,5	1,3	3,9	3,7	1,3
PartSizeSC	3,4	2,4	2,1	3,3	2,3	1,8
MergeFlagExtSC	2,7	1,8	1,7	2,6	1,8	1,5
MergeIdxExtSC	5,4	4,3	2,1	5,5	4,6	2,0
InterDirSC	0,6	0,5	0,0	0,6	0,5	0,0
RefPicSC	0,9	0,6	0,1	0,8	0,6	0,1
MvdSC	3,7	2,1	2,0	3,5	2,0	1,7
MVPIdxSC	1,2	0,7	0,7	1,2	0,7	0,5
Total Motion	23,1	17,7	9,7	22,5	17,5	8,7
Total Part	7,5	6,1	3,5	7,5	6,2	3,2

Table 6.6: Percentage of the bitrate for motion and partitioning syntax elements - QP20

	Arc			Linear		
	Butterfly	*Flowers*	*Rabbit*	*Butterfly*	*Flowers*	*Rabbit*
PartSize	0,4	0,3	0,4	0,4	0,3	0,4
MvdSign	1,8	1,2	1,4	1,8	1,1	1,1
MvdAbs	4,1	4,1	2,0	4,1	3,8	1,6
RefFrmIdx	0,0	0,0	0,0	0,0	0,0	0,0
MrgIdx	9,9	6,3	4,1	9,0	6,2	3,8
SplitFlagSC	8,2	6,3	3,6	8,3	6,6	3,4
PartSizeSC	4,2	2,9	4,1	4,0	2,8	3,6
MergeFlagExtSC	3,0	2,2	2,7	2,9	2,1	2,3
MergeIdxExtSC	13,3	8,4	5,1	13,7	9,1	4,9
InterDirSC	0,6	0,4	0,0	0,6	0,4	0,0
RefPicSC	1,0	0,6	0,1	0,9	0,6	0,1
MvdSC	3,7	2,1	3,0	3,7	1,9	2,2
MVPIdxSC	1,3	0,8	0,9	1,3	0,8	0,7
Total Motion	38,7	26,3	19,5	38,0	26,0	16,6
Total Part	12,8	9,4	8,2	12,8	9,7	7,4

Table 6.7: Percentage of the bitrate for motion and partitioning syntax elements - QP35

	QP 20	QP 25	QP 30	QP 35
PartSize	0,2	0,2	0,2	0,3
MvdSign	1,4	1,7	2,0	2,1
MvdAbs	2,8	3,9	5,2	6,2
RefFrmIdx	0,0	0,0	0,0	0,0
MrgIdx	2,7	2,8	3,0	3,2
SplitFlagSC	2,1	2,6	3,3	3,9
PartSizeSC	2,4	2,7	3,0	3,2
MergeFlagExtSC	1,5	1,7	2,0	2,2
MergeIdxExtSC	2,3	3,1	4,1	5,3
InterDirSC	0,6	0,6	0,5	0,4
RefPicSC	0,6	0,6	0,5	0,5
MvdSC	2,8	3,2	3,4	3,3
MVPIdxSC	1,0	1,1	1,2	1,3
Total Motion	15,8	18,8	21,9	24,6
Total Part	4,6	5,5	6,5	7,4

Table 6.8: Percentage of the bitrate for motion and partitioning syntax elements - *PoznanBlocks*

the opposite directions of disparity vectors in arc content, and we provide preliminary results.

6.5.1 Modification of NBDV

To exploit the opposite directions of the disparity for arc content in 3D-HEVC, we first propose to modify NBDV and the merge candidate list. As described in Chapter 5, in the state-of-the-art section, NBDV looks into neighboring PUs for a DV. When one is found, the research stops. The DV is used to create one IVMC candidate with the IVMP tool, where the candidate vector in the merge list is the MV from the PU pointed by the DV. Secondly, one IVDC candidate is created, where the candidate vector in the merge list is the DV itself. Similarly to the method proposed in Chapter 5, we propose here to continue the search in order to find a second DV in the opposite direction. Then, two IVMC and/or two IVDC candidates are created in the merge list.

We first analyze the percentage of cases where the method could be applied. Sequences are encoded without temporal prediction (i.e. only intra or inter-view prediction). Table 6.9 shows the percentage of PUs that are predicted: in Intra, with a DV from left to right, or with a DV from right to left. It also shows the percentage of PUs (among the total number of PUs) for which the first DV found is the correct one (good direction) and when it is not, when only one DV has been found by NBDV, and when only two DVs have been found.

Only a small number of PUs have incorrect direction for the first and only DV found (from 0.2% to 3.0%). This number is larger in arc (1.7% to 3.0%) than linear (0.2% to 1.8%). The number of cases where a correct second DV is found is also small with around 0% to 0.3% in linear, and 0.8% to 1.5% in arc. These results confirm a difference between arc and linear for the efficiency of the prediction of the direction of DVs. However, there is only a small number of cases where the method can be applied and provide benefits (i.e. when two DVs are found), and among those cases, there is a larger number of cases where

	Total PUs	Intra	Left	Right	Inter-view - NBDV found			
					one DV only		two DVs	
					correct	wrong	first	second
Butterfly Arc	1236644	15%	44%	41%	78%	3,0%	2,9%	1,5%
Flowers Arc	1617112	44%	30%	26%	52%	1,7%	1,4%	0,8%
Rabbit Arc	3811122	9%	46%	45%	84%	2,6%	3,8%	1,1%
Butterfly Lin	1042727	17%	50%	33%	79%	1,8%	2,1%	0,3%
Flowers Lin	1404742	43%	29%	28%	54%	1,7%	1,5%	0,1%
Rabbit Lin	3370083	10%	43%	46%	89%	0,2%	0,1%	0,0%

Table 6.9: Different cases for the number and accuracy of DVs found by NBDV

	arc	linear
Butterfly	0,26%	0,08%
Flowers	0,04%	0,07%
Rabbit	-0,04%	-0,03%

Table 6.10: BD-rate results with 2 IVMC and 2 IVDC candidates

the first DV is already the correct one (e.g. for Butterfly Arc, in 1.5% of the PUs, the second one has the good direction, while the first one already has the good direction in 2.9% of the cases). Therefore, low coding gains are expected. Table 6.10, Table 6.11, and Table 6.12 provide BD-rate results for the proposed method tested on 10 frames of the *Bunny* sequences (i.e. *Butterfly*, *Flowers* and *Rabbit*), using HTM 13.0. In order to focus these first results on disparity prediction, a configuration without temporal prediction is used.

No significant gains are reported and some cases even provide slight losses. This modification of the NBDV process and merge candidate list does not improve the performance. Adding a second IVMC and/or a second IVDC is costly, and the few cases where it improves the prediction cannot compensate for the cases where the candidates are added for nothing. Because it is not efficient to try to find directly the disparity vector here, in the following we propose to improve its prediction using AMVP.

6.5.2 Modification of AMVP

We propose to exploit the opposite directions of the disparity in arc content by modifying the Advanced Motion Vector Prediction (AMVP) tool. Similarly to NBDV, AMVP searches neighboring CUs for a disparity (or motion in the case of temporal prediction) vector, but in order to predict the vector used for the current CU. A list of MVP candidates is constructed and only the index of the predictor in the list and the difference between this predictor and the current vector are transmitted. The advantage, compared to the modification of NBDV proposed in Section 6.5.1, is that even if the candidate vector

	arc	linear
Butterfly	0,06%	-0,01%
Flowers	0,08%	0,02%
Rabbit	-0,03%	0,00%

Table 6.11: BD-rate results with 2 IVMC candidates

	arc	linear
Butterfly	0,26%	-0,10%
Flowers	-0,03%	0,01%
Rabbit	-0,04%	-0,03%

Table 6.12: BD-rate results with 2 IVDC candidates

	arc	linear
Butterfly	0,01%	-0,02%
Flowers	-0,25%	-0,01%
Rabbit	-0,10%	-0,02%

Table 6.13: BD-rate results with additional MVP candidates (opposite directions)

does not correspond to the current vector, using the correct direction could still improve the prediction, while in NBDV the vector is used directly (in the merge list) or not at all.

In this section we provide preliminary results for this method. In this exploratory phase, we perform experiments where candidates are added in the list without being signaled (i.e. the bitstream cannot be decoded), to check if adding candidates in the opposite direction can improve the performance.

Table 6.13 shows BD-rate results when two MVP candidates, MV_2 and MV_3, in the opposite directions of the two first candidates, MV_0 and MV_1, are added to the list (i.e. $MV_2 = -MV_0$ and $MV_3 = -MV_1$). We simulate the encoding of the index by signaling the index of the candidate modulo 2, so that the cost of the signal overhead does not (significantly) impacts on the performance evaluation. This modification has (almost) no effect in the linear case. With arc content, slight gains for Rabbit and Flowers (0.1% to 0.3%). However in this experiment, there is a cheat for signaling, therefore these gains cannot be considered as significant enough to be promising.

Table 6.14 reports BD-rate results for a second modification of AMVP where for each MVP candidate (i.e. from left, top, and temporal neighboring PUs), the search continues until one vector in each direction is found. This method does not bring significant gains or losses either.

Finally, Table 6.15 reports results for a combination of the two previous modifications, i.e. first for each MVP candidate, the search continues until one vector in each direction is searched, then the opposite values are added as MVP candidates. BD-rate gains are significantly higher (up to 0.7%). However, these results do not hint for significant gains or improvements considering that the indexes are not signaled. Moreover the results are very similar for arc and linear, which hints that the gains are mostly due to the larger number of possible candidates, and not to the specific directions of arc content.

	arc	linear
Butterfly	0,08%	-0,25%
Flowers	-0,12%	0,01%
Rabbit	-0,04%	0,06%

Table 6.14: BD-rate results with additional MVP candidates (search continued)

	arc	linear
Butterfly	-0,5%	-0,7%
Flowers	-0,4%	-0,5%
Rabbit	-0,7%	-0,6%

Table 6.15: BD-rate results with additional MVP candidates (combination)

6.6 Conclusion

In this chapter we assess how arc camera arrangements impact on the compression efficiency. We show that the direction of the disparity vectors is not the same in linear content (with always the same direction) and arc content (with two opposite directions possible). We compare the performances of existing coding technologies on linear and on arc camera arrangements. Results show no significant performance difference between arc and linear on the test set used in the proposed experiments. Hence we propose perspectives to improve specifically the performance in the arc case (without degrading it for the linear case). We propose to improve the prediction of the DVs by taking into account this possibility of opposite directions. However, there are only few cases where the modifications have an impact, i.e. at the borders of objects with opposite disparities (foreground and background). Therefore, there is no significant improvement of the performance with the proposed tools. However, the in-depth study of the encoding experiments strongly suggests that there is room for further improvements in other specific coding tools. Moreover, other complementary specific aspects of the arc content can be taken into account, such as the rotation of parts of the pictures from one view to another for example.

Chapter 7

Compression scheme for free navigation applications

7.1 Introduction

In this chapter, we study the compression of Super Multi-View content targeting Free Navigation (FN) applications. We focus on a use case where all the pictures are encoded and sent to the decoder, and the user interactively can request to the decoder (and renderer) a point of view to be displayed (e.g. on a state-of-the-art 2D display). The main goal is to study the tradeoff between the rate-distortion performance and the degree of freedom offered by the coding schemes.

For compression efficiency, the best configuration consists in exploiting all the redundancies (i.e. intra, temporal, inter-view), for example with a multi-view encoder like 3D-HEVC and a large prediction structure. However in that case, a lot of dependencies are introduced, and the freedom of navigation is limited by the decoding complexity constraints, as a large number of pictures has to be decoded in order to display one. The best configuration in terms of freedom of navigation consists in encoding all the pictures independently, using only intra prediction with HEVC for example. However, this configuration cannot exploit temporal or inter-view redundancies, hence it is not efficient in terms of compression performance.

The problem of finding a tradeoff between compression efficiency and freedom of navigation can be tackled from several angles: first starting from an *All Intra* (AI) configuration and increasing efficiency while maintaining (at most possible) the freedom of navigation; secondly starting from the multi-view configuration and increasing the freedom of navigation while maintaining (at most possible) the compression efficiency; or finally by proposing intermediary methods that perform in-between.

State-of-the-art coding methods for Free Navigation applications are presented in Section 7.2. In Section 7.3, we study the coding performances of state-of-the-art methods based on current compression standards and encoders with several configurations, regarding the tradeoff between two main criteria: compression efficiency (i.e. lowest bitrate possible for a given image quality) and degree of freedom (i.e. the ability for the user to change the viewpoint, that mainly depends on the decoder capability and the number of pictures to decode in order to display one). Additionally we propose in an appendix chapter (Sec. 8.1) a coding scheme dedicated to Free Navigation that performs redundant encodings, thus allowing the user to shift the viewpoint without decoding additional pictures,

aiming a tradeoff between the two aforementioned methods in terms of rate-distortion performance and freedom of navigation.

7.2 State-of-the-art

Free Navigation applications allow the user to change the point of view of the scene to display. Most of the time in the literature, either the encoding is performed online, depending on the request of the user, either it is performed offline and the part of the encoded content that is transmitted depends on the request of the user. The main aspect of the study in those cases concerns the tradeoff between the storage (of the encoded content at the encoder side) and the bandwidth (that is required to transmit the requested content). However, in both cases, a feedback channel is required between the decoder/user and the encoder, implying a constraint of latency that is currently not realistic for practical applications. Hence, as mentioned in Sec. 7.1, in this chapter we focus on a use case where all the pictures are encoded and sent to the decoder, and the user interactively can request to the decoder a point of view to be displayed. The main goal here is to study the tradeoff between the rate-distortion performance (i.e. corresponding to the aforementioned bandwidth) and the degree of freedom offered by the coding schemes (that depends on the decoding complexity constraints).

As mentioned in the introduction section, encoders like HEVC and 3D-HEVC can be used in the context of Free Navigation applications. The performance of these encoders with different configurations is studied in details in Sec. 7.3. One of the main purposes of the coding schemes for Free Navigation is to be able to compress as efficiently as possible the content while permitting to reduce the number of pictures to decode, and still decode the same content with an exact reconstruction in any case (i.e. not depending on the navigation path requested by the user). This problem consisting concretely in being able to decode a given picture from several references while guaranteeing the same reconstruction, whatever the available references are, has been treated in the literature.

SP/SI frames are images that are encoded at specific positions in a set of bitstreams (at least two, corresponding to two views in our case). Although SP/SI frames are part of H.264/AVC in the *Extended* profile, they are rarely used in practice, and they were not kept in HEVC. They allow to reconstruct the same picture using different reference pictures, i.e. using a picture from the current bitstream if there is no view switching, or using a picture from another bitstream that has been decoded before a view switching [136]. In the case of SP/SI frames, residual coefficients obtained from the prediction using the second reference are encoded losslessly. However, because the motion estimation is performed with another reference, motion parameters are not adapted, hence the predictor is not efficient and this residual information is costly to encode. Because of this cost, SP/SI frames can only be used in a sparse manner, and cannot allow view switching for every frame. In the method proposed in the appendix chapter (Sec. 8.1), motion parameters obtained with the second reference are transmitted to the decoder, in order to have a residual information that is less costly to encode.

S frames [137] differ from SP/SI frames. Lossless coding is removed, and a quantization step is added. Coding efficiency is improved, however the reconstructed pictures are not strictly identical. The mismatches that are introduced when switching views/bitstreams prevent the Free Navigation functionnality as it is described above.

In a similar way, merge frames (M frames) group the information coming from two separate reconstructions (i.e. from two different reference views in our case) into one unique

reconstruction that can be used as a reference to encode the next frames. A Piece Wise Constant (PWC) function is applied as a merging operator on the transformed residual coefficients from the two separate reconstructions to obtain a target reconstruction. The parameters of this PWC function are transmitted to the decoder that can perform the same operation from any of the available reference pictures [138].

The principle of redundant P frames consists in performing several encodings of the current frame, with different references, offering multiple possible decoding paths [139]. In that case, the number of frames to encode increases drastically in a way that is not realistic for Free Navigation applications targeted in this chapter with view switching available for each frame.

An interactive multi-view video system with low decoding complexity is described in [140][141], based on a bitrate allocation for the views that depends on the predicted user's behavior. Although coding efficiency is improved, in that case the encoding process depends on the user's navigation path. A different data representation method is proposed in [142] based on the notion of navigation domain, which is optimally split into several navigation segments, described with one reference image and auxiliary information. A solution for effective partitioning of the navigation domain and for selecting the best position for reference images is given, in order to improve the viewing experience with a reasonable rate and storage cost. A quality scalable method is described in [143], where views are optimally organized in layers, each one offering an incremental improvement for the generated virtual views during the navigation.

The Distributed Video Coding (DVC) approach, similar to Distributed Source Coding (DSC), is based on the use of key frames, Wyner-Ziv (WZ) frames and side information. Key frames are encoded independently, for example with intra prediction. Side information is obtained at the decoder side, for example by interpolating key frames [144][145], in order to predict WZ frames. WZ frames are encoded with channel coding, and only the correction bits are transmitted to the decoder. The decoder uses these elements to correct its own prediction from the side information. The advantage of the DVC approach is that WZ frames can be encoded separately from the reference frames. This corresponds to the Free Navigation paradigm, where reference pictures available at the decoder side are unknown at the encoder side because they depend on the navigation path requested by the user. Typical target applications for DVC coding schemes are discussed in [146].

The efficiency of DVC coding schemes strongly depends on the quality of the side information [147]. Side information generation methods are reviewed in [148], and a review of existing fusion methods for the merge of temporal and inter-view estimations is given in [149]. Several different methods to improve the side information have been proposed, for example based on a motion smoothing step in [150], or also on a motion compensated refinement of the decoded frame [151]. Other examples of approaches based on temporal estimation, inter-view estimation, and Depth Image Based Rendering are given in [152],[153] and [154]. [155] provides a review of existing Wyner-Ziv frames coding methods. A method based on the regularization of the motion field is proposed in [156].

While information theory states that the upper limit of the compression performance of distributed coding is the same as in the case of predictive coding, practical implementations of these systems have significantly lower performances. Moreover, the DVC approach tends to reduce complexity at the encoder side and to increase complexity at the decoder side (with motion estimation processes for example). Although methods to share complexity between encoder and decoder [157] have been proposed, Free Navigation applications require a decoder that is the least complex possible. Additionally, in most cases a feedback

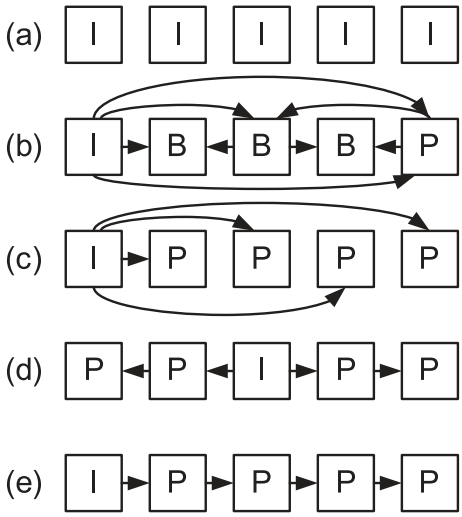


Figure 7.1: Prediction structures. (a) AI, (b) Hierarchical, (c) LDP, (d) DP central, (e) DP

from the decoder is required to know the quantity of information required to transmit. Some approaches without feedback have been proposed, but the decrease in performance is large.

Although the methods presented in this section present interesting advantages for Free Navigation applications, they also are limited by several constraints (e.g. feedback channel or low coding efficiency) that make them sub-optimal for the conditions that are studied in the chapter (i.e. offline encoding and transmission of all the pictures without feedback channel).

7.3 Performances comparison with existing encoders in different configurations

In this section, we study the coding performances of state-of-the-art methods based on current encoders with several configurations, regarding the tradeoff between compression efficiency and degree of freedom (that mainly depends on the decoding complexity and the number of pictures to decode in order to display one).

7.3.1 Tested structures

In this section, several state-of-the-art methods are compared. These methods are based on 3D-HEVC encodings with different prediction structures. Figure 7.1 illustrates the different prediction structures used [20]. The *All Intra* structure illustrated in *(a)* only has intra coded frames (AI or III...I), i.e. no dependency between pictures. *(b)* is the Hierarchical structure for common HEVC's temporal prediction (in Random Access profile). *(c)* is the Low Delay P structure (LDP or IPP...P) with several possible references, but only one used at a time. Finally *(d)* and *(e)* represent the Direct P structure (DP) where only one adjacent picture can be used as a reference. The only difference is the position of the intra frame (i.e. side or central). These structures can all be used either for temporal or for inter-view prediction.

We propose to evaluate several combinations of the temporal and inter-view prediction structures presented in Fig. 7.1, as listed in the following configurations. Intra Periods (IPs) are given for an example with 20 views corresponding to the configuration used in our experiments (as described in Section 7.3.3). Details for these configurations are sumarized in Table 7.1.

- **All Intra (AI)** All pictures are encoded in intra mode. No inter-view or inter-frame (temporal) prediction.
- **FTV (CfE) configuration** [158] Equivalent to Group of Group of Pictures (GoGoP, also called Group Of Views or GOVs) with size 24×20. Inter-view structure is IPP direct P (with IP = 20), temporal structure is Hierarchical (with IP = 24)
- **Inter-view only** All pictures are encoded in inter mode (i.e. in competition between inter and intra mode) with inter-view prediction only and no temporal prediction. Inter-view structure is IPP direct P (with IP = 20), as for FTV (CfE) configuration.
- **Temp. only hiera** All pictures are encoded in inter mode with temporal prediction only and no inter-view prediction. Temporal structure is Hierarchical (with IP = 24), as for FTV (CfE) configuration.
- **Temp. only LDP** All pictures are encoded in inter mode with temporal prediction only and no inter-view prediction. Temporal structure is Low Delay P (LDP with IP = 24).
- **Temp. only direct P** All pictures are encoded in inter mode with temporal prediction only and no inter-view prediction. Temporal structure is Direct P (DP with IP = 24).
- **Temp. only direct P, IP 4** All pictures are encoded in inter mode with temporal prediction only and no inter-view prediction. Temporal structure is Direct P (DP with IP = 4).
- **GoGoP** 5×8 Inter-view structure is IPP direct P (with IP = 5), temporal structure is Hierarchical (with IP = 8).
- **GoGoP** 10×16 Inter-view structure is IPP direct P (with IP = 10), temporal structure is Hierarchical (with IP = 16).

7.3.2 Performance evaluation

In this section, we provide a list of criteria that are relevant to evaluate the performance of a coding scheme related to Free Navigation applications. These items take into account the compression efficiency and the degree of freedom offered by the scheme.

- **BD-rate total** Coding efficiency based on the total bitrate and on the distortion (PSNR) for all the pictures.
- **BD-rate displayed** Coding efficiency based on the total bitrate and on the distortion (PSNR) of the displayed pictures only (i.e. requested by the user on a given navigation path).

Configuration	temporal		inter-view	
	structure	IP	structure	IP
All Intra (AI)	III...I	1	III...I	1
FTV (CfE) configuration	Hiera.	24	IPP...P (DP)	40
Inter-view only	III...I	1	IPP...P (DP)	40
Temp. only hiera	Hiera.	24	III...I	1
Temp. only Low Delay P	IPP...P (LDP)	24	III...I	1
Temp. only Direct P	IPP...P (DP)	24	III...I	1
Temp. only Direct P, IP4	IPP...P (DP)	4	III...I	1
GoGoP 5×8	Hiera.	8	IPP...P (DP)	5
GoGoP 10×16	Hiera.	16	IPP...P (DP)	10

Table 7.1: Summary of tested configurations and corresponding structures

	All Intra (AI)	inter-view only	temporal only				GoGoP	
			Hiera	LDP	DP	DP, IP 4	5×8	10×16
Flowers Linear	1878	529	178	237	246	609	73	20
Flowers Arc	1680	555	151	203	212	537	68	19
Champagne	2463	1376	64	80	85	622	121	38
Average	2007	820	131	174	181	589	88	26

Table 7.2: Total BD-rate (%) against FTV (CfE) configuration

- **Number of decoded per displayed pictures (decoded/displayed)** On a given navigation path, this corresponds to the average number of pictures that have to be decoded to display one, i.e. it is the total number of decoded pictures for the complete path divided by the total number of (temporal) frames.
- **Decoding time** Total time required to decode all the pictures that are necessary to display the requested navigation path. Therefore in practice it is the sum of the decoding times of all the decoded pictures included in the third criterion.
- **Encoding time** Total time required to encode all the pictures.
- **Number of encoded pictures** Total number of pictures encodings. It can be more than the total number of pictures for example in case of redundant coding.

7.3.3 Experimental conditions

The state-of-the-art methods listed in Section 7.3.1 are tested under the following experimental conditions. Two Computer Generated (CG) sequences (*Flowers Arc* and *Flowers Linear*), and one natural sequence (*Champagne*) are tested. The two CG sequences represent the same scene rendered with a linear and with an arc camera arrangement (see Chapter 6). Experiments are performed under 3D-HEVC reference software (HTM 13.0) with the modifications provided by the MPEG FTV Ad Hoc Group [61]. 20 views are encoded, each composed of 40 frames.

7.3.4 Experimental results

Table 7.2 reports total BD-rate results (i.e. bitrate and PSNR computed on all the encoded pictures) with the FTV (CfE) configuration as anchor. The All Intra (AI) configuration

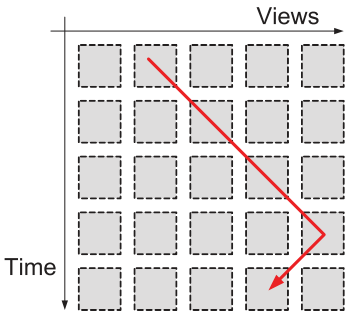


Figure 7.2: Basic path from left to right, then right to left, etc.

provides a very large BD-rate loss (of approximately 2000% in average) over the FTV (CfE) configuration (that is the most performant configuration in terms of compression efficiency). In other words, the AI configuration requires around 21 times the bitrate of the FTV (CfE) configuration. The configuration with inter-view prediction only is less performant than the configurations with temporal prediction only with IP 24. This is mainly due to the fact that the disparity of the objects between the views increases faster than the temporal distance between the frames (e.g. a larger shift is observable between view 0 and view 19 than between frame 0 and frame 19). Therefore less correlations can be exploited in inter-view. In temporal only, Low Delay P and Direct P are significantly less efficient than the hierarchical structure. Direct P with Intra Period of 4 is also much less efficient (even lower than inter-view only for *Flowers Linear*), due to the increased number of intra frames. Tab. 7.2 shows that the IP is the main parameter that impacts the BD-rate performance in this experiment. Hence the configuration GoGOP 10×16 is less performant than the FTV (CfE) anchor configuration because the number of intra frames is increased. With GoGOP 5×8, this number is increased by 4 (2 times in the view dimension and 2 times in temporal dimension).

In order to measure the performance in the conditions of a Free Navigation use case (with the criteria as described in Sec. 7.3.2), we simulate the path followed by a user exploiting to the maximum the number of view switching possible, i.e. going from left to right (and then right to left) at every frame, with a step of plus or minus one for the view index, as illustrated in Figure 7.2. All the possible starting points (i.e. starting view indexes) are considered, in order to determine which one provides the worst case for each configuration in terms of number of decoded pictures, and of decoding time (as both can possibly be different because the decoding time is not the same for I, P, and B frames). Results are presented respectively in Table 7.3 and in Table 7.4.

Results show that the number of decoded pictures and the decoding time are proportional for all the tested configurations. The reason is that complete Groups of Views are decoded, therefore the proportion of I frames and P/B frames does not vary depending on the starting point. The *worst case* path selected to evaluate the Free Navigation performance must be the same for all the configurations, in order to compare the PSNR values of the same pictures for example (in the case of the Displayed BD-rate criterion). View index 9 is used as the starting point in the following, as it is the worst case for the GoGoP configurations.

Table 7.5 presents the performance according to the aforementioned criteria (see Sec. 7.3.2), obtained as follows.

- *Decoded per displayed* is the average number of decoded pictures for one picture

Start index	All Intra (AI)	FTV (CfE) config	inter-view only	temp. only				GoGoP	
				Hiera.	LDP	DP	DP, IP4	5×8	10×16
0	40	800	800		720		156	400	720
1					680		152	440	720
2					640		156	360	720
3					600		160	440	720
4					560		156	480	720
5					520		152	400	720
6					480		156	440	720
7					456		160	400	720
8					472		156	400	720
9					512		152	480	800
10					552		156	400	640
11					592		160	400	640
12					632		156	440	640
13					672		152	400	480
14					712		156	480	640
15					736		160	440	640
16					736		156	360	560
17					736		152	440	720
18					736		156	400	720
19					720		156	400	720

Table 7.3: Number of decoded pictures depending on the starting view index for each tested configuration

Start index	All Intra (AI)	FTV config. (CfE)	inter-view only	temp. only				GoGoP	
				Hiera.	LDP	DP	DP, IP4	5×8	10×16
0	3,5	83,4	96,5	52,7	57,8	57,6	12,9	21,3	40,0
1	3,5			49,7	54,7	54,5	12,4	23,2	40,0
2	3,5			46,9	51,6	51,4	12,7	18,5	40,0
3	3,6			44,0	48,4	48,3	13,1	22,6	40,0
4	3,5			41,2	45,3	45,1	12,8	25,0	40,0
5	3,6			38,3	42,2	42,0	12,6	21,2	40,0
6	3,6			35,4	39,0	38,7	12,9	23,2	40,0
7	3,6			33,6	36,9	36,7	13,2	21,0	40,0
8	3,6			34,7	38,1	37,9	13,0	21,3	40,0
9	3,6			37,6	41,4	41,1	12,7	25,8	44,7
10	3,6			40,6	44,6	44,3	13,0	21,5	35,2
11	3,6			43,4	47,7	47,6	13,5	21,3	35,2
12	3,6			46,3	50,8	50,7	13,2	23,3	35,2
13	3,6			49,3	54,0	53,7	12,7	21,5	27,3
14	3,6			52,0	57,2	56,9	13,0	26,0	36,8
15	3,6			53,8	59,1	58,8	13,4	23,8	36,8
16	3,6			53,8	59,1	58,8	13,0	19,1	32,2
17	3,6			53,8	59,0	58,9	12,7	23,2	40,1
18	3,6			53,8	59,0	58,9	13,0	21,2	40,1
19	3,6			52,7	57,8	57,7	12,9	21,6	40,1

Table 7.4: Decoding time (s) depending on the starting view index for each tested configuration (*Flowers Linear* QP25)

Config.	BD-rate (%) total	BD-rate (%) displayed	Coding time (%)	Dec. time (%)	Decoded per displayed
FTV (CfE) config.	*(anchor)*				20
All Intra (AI)	2007	2028	151	5	1
inter-view only	820	823	254	118	20
temp. only Hiera.	131	134	174	54	12,8
temp. only DP IP4	589	597	104	16	3,8
GoGoP 5×8	88	90	113	32	12
GoGoP 10×16	26	27	101	57	20

Table 7.5: Performance in reference to FTV config scheme (average on 3 tested sequences)

displayed, obtained by dividing the total number of decoded pictures for the experiment (i.e. displayed pictures plus all the pictures in the corresponding GOVs) by the number of frames (i.e. the number of displayed pictures).

- *Dec. time* is the sum of the decoding times for each of these decoded pictures. *Coding time* corresponds to the total runtime required to encode all the pictures.
- *Total* and *Displayed* BD-rate values (as described in Sec. 7.3.2) are computed against the FTV (CfE) configuration as an anchor.

It should be noted that the BD-rate values *total* and *displayed* are very close for all of these configurations, mostly because the encoding quality is very consistent overall within the GOVs. The results confirm that the prediction structure is not the main factor for the performance, because the Intra Period has a much larger impact, as complete Group of Views have to be decoded in every cases. These results are further analysed in Section 7.3.5.

7.3.5 Results analysis

In order to analyze the results presented in Tab. 7.5, we first set the following constraints, considered to be realistic for future services:

- Decoding times can be 10 times larger (1000%) when compared to a 2D frame with the same resolution, i.e. equivalent to decoding 10 frames for one displayed frame.
- Assuming an offline coding first, there is no runtime constraint for the encoding time.
- Bitrate values can be about three times the bitrates expected for 8K frames requirements.

As mentioned in Chapter 4, bitrates required for 4K content are reported to be from 4 to 30 Mbps, and from 15 to 80 Mbps for 8K content. We provide experimental results in Table 7.6 for comparison. This 2D anchor corresponds to the encoding of a 4K resolution sequence (*CatRobot*, 3840×2160, 60 frames 60fps, 10 bits) with the JEM2.0 reference software [29] in Random Access profile (see Chap. 1, Sec. 1.4). Table 7.6 confirms that, for this sequence, results with PSNR between 38 and 40 dB (hence that are considered as an acceptable/good quality) are associated to bitrates approximately between 4 and 30 Mbps.

QP	Bitrate (Mbps)	PSNR (dB)
22	36.0	40.4
27	9.1	39.3
32	4.3	37.9
37	2.4	36.2

Table 7.6: Examples of results for a 4K sequence encoding in 2D with JEM2.0 (*CatRobot*, 3840 × 2160, 60 frames, 60fps, 10bits)

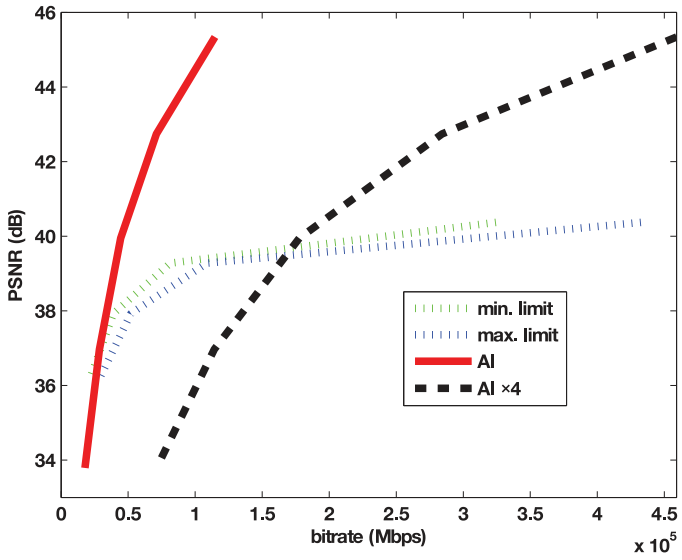


Figure 7.3: *AI* results (also multiplied by 4 to simulate 80 views) compared to estimated limit bitrates - *Flower* Arc

In the following, we roughly approximate bitrate values for 8K frames by the bitrate required for this example 4K sequence (36.0 Mbps for the point at QP 22, with a PSNR around 40dB) multiplied by 3 or 4, providing bitrates that range from 108 to 144 Mbps. We mentioned above that we consider a bitrate constraint of 3 times the bitrates required for 8K frames. This estimation provides a limit between 324 Mbps and 433 Mbps.

In our experiments, for the *AI* configuration, bitrates are measured from 136 Mbps (for *Champagne*, at QP 45, with PSNR 35.5 dB) to 1148 Mbps (*Flowers Arc*, QP 20, PSNR 45.3 dB). For the FTV (CfE) configuration, bitrates are measured from 0.3 Mbps (*Champagne*, QP 45, PSNR 34.5 dB) to 5 Mbps (*Flowers* Arc, QP 20, PSNR 43.2 dB). As mentioned in our experimental conditions, we have encoded only 20 views, which is expected to be few for future use cases. This depends on the angle of view and on the number of viewpoints provided to the users in future Free Navigation applications. For example, most of the currently available SMV contents have around 80 views. Considering 80 views, the bitrate should basically be 4 times larger.

Figure 7.3 shows the bitrates of the *AI* configuration (for *Flowers* Arc) with 20 views in plain line, and also the results with bitrates multiplied by 4 to simulate 80 views. The estimated constraints (i.e. three and four times the 8K bitrates) are also presented in dotted lines. *AI* configuration meets the decoder constraint as only one picture is

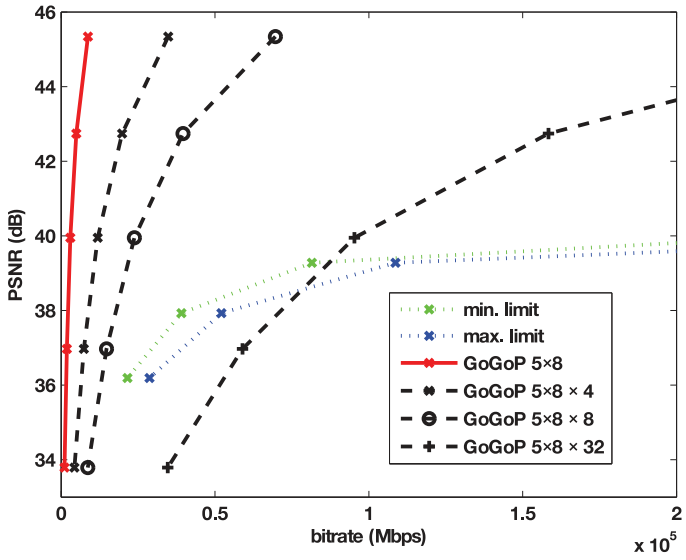


Figure 7.4: GoGoP 5×8 results compared to estimated limit bitrates - *Flower* Arc

decoded per picture displayed. However, the bitrate is too large, at least up to 39 dB, as shown in Fig. 7.3. We can also comment further on the frame rate and resolution of *Flowers* (i.e. 24fps, 1280×768) which are too low according to the expectations for future video services. Therefore it appears necessary to consider more efficient configurations. FTV (CfE) configuration is the most performant configuration in terms of compression efficiency. However it does not meet our decoder constraint (because all the pictures have to be decoded in most of the cases). Our experimental results show that the GoGoP 5×8 configuration approximately meets the decoder constraint with 12 pictures decoded per picture displayed (which close enough considering the rough estimation of the constraints), and only provides 88% BD-rate loss in average against the FTV (CfE) configuration (see Tab. 7.5). Figure 7.4 shows the GoGoP 5×8 bitrates from these experiments (which are largely below the estimated limit of three times the 8K frames bitrates). However, as mentioned above, the resolution, framerate and number of views are low compared to expectations for future services. Hence Fig. 7.4 also shows in dotted lines the GoGoP 5×8 bitrates successively multiplied by:

- 4 : to simulate 80 views
- 8 : to simulate 80 views, with a frame rate increased by two
- 32 : to simulate 80 views, with frame rate increased by two, and a larger resolution (with a width of approximately 2K pixels in this example)

Fig. 7.4 shows that even when multiplied by four or eight (i.e. to simulate 80 views and a higher frame rate), the bitrates for GoGoP 5×8 configuration are significantly below the estimated limit. However, when simulating also a larger resolution, the values become too large up to approximately 39 dB (very similarly to *AI* configuration results). Bitrates are approximately two times too large at 36 dB. The gap decreases for higher bitrates (as the quality increases up to 39 dB). That increase (of approximately 100%) is too large to be considered realistic here, and it is even larger when considering larger resolutions

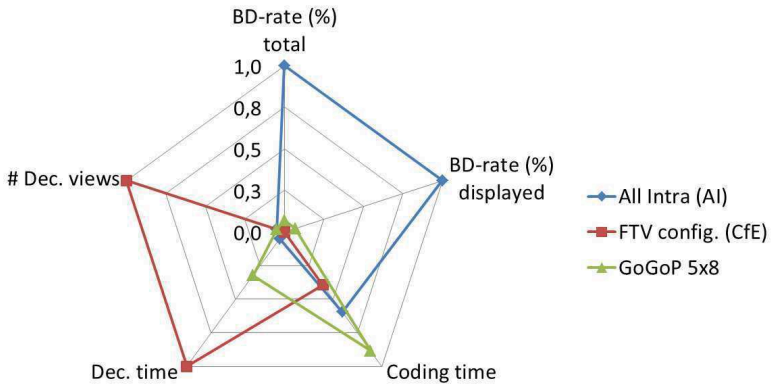


Figure 7.5: Tradeoff between the two extreme cases and an intermediary solution (values are normalized to the maximum)

(e.g. 4K or 8K). It should be noted that additionally to the bitrate constraints, an increased resolution also increases the decoding time.

Figure 7.5 and Table 7.7 provides a summary of the comparison of the anchor configurations, and shows that intermediate configurations can provide a tradeoff according to our criteria by reducing the bitrate and number of pictures to decode (at the expense of an increase in encoding time).

As a conclusion, when taking into account the expectations for future applications in terms of resolution, frame rate and number of views, the estimations based on our results show that improvements are required to reach a satisfying tradeoff between compression efficiency and freedom of navigation.

7.4 Conclusion and perspectives

In this chapter, we study the compression of Super Multi-View content targeting Free Navigation (FN) applications. We focus on a use case where all the views are encoded and sent to the decoder, and the user interactively can request to the decoder (and renderer) a point of view to be displayed (e.g. on a state-of-the-art 2D display). Several configurations and prediction structures are first compared for the encoding of Super Multi-View sequences. The performances are evaluated according to the tradeoff between the rate-distortion efficiency and the degree of freedom offered by the coding schemes. When taking into account the expectations for future applications in terms of resolution, frame rate and number of

	Bitrate	Decoding time and number of decoded pictures
All Intra (AI)	Max	Min
FTV (CfE) config.	Min	Max
GoGoP 5×8	Large	OK

Table 7.7: Summary of our conclusions on anchor comparisons

views, the estimations based on our results show that improvements are required to reach a satisfying tradeoff. Therefore in an appendix chapter, we propose as a perspective a coding scheme that allows decoding the views requested by the user using several possible references, i.e. in several cases of navigation path requested by the user, when the available reference view at the decoder is not known in advance at the encoder.

Chapter 8

Conclusion

In this chapter we first summarize all the contributions described in the manuscript, and for each chapter and associated contributions, we discuss perspectives of future work to further extend the results and conclusions of our studies. A list of the publications resulting from this work is given at the end of the manuscript (before the bibliography). In a second part we give a global vision of our work regarding the current context and the paradigm of light-field technologies development, and we discuss the current and expected status of immersive media through the scope of light-field image and video coding.

An overview of state-of-the-art light-field technologies from capture to display is provided in Chapter 2. A significant variety of capture devices, display systems, formats and associated processing tools currently coexist. Efforts are made by experts in the domain in order to emphasize on the commonalities, e.g. between the different ways of representing a light-field with object-based approaches like Point Clouds and image-based approaches like Super Multi-View and Integral Imaging. Promising target applications are emerging, exploiting different features of these technologies. For example, while Point Clouds suit quite well augmented reality or virtual reality with Head Mounted Displays, Super Multi-View would be more adequate for immersive telepresence, and Integral Imaging is currently used for plenoptic photography applications like refocusing. Display technologies are developing fast but still face several technical limitations, mainly related to spatial and angular resolution. Although current coding technologies can be used to compress light-field content (e.g. HEVC for integral images, 3D-HEVC for Super Multi-View video, with only slight syntax modifications), they are expected to provide sub-optimal performances. There is also a lack of effective processing tools dedicated to light-fields, for example current depth maps estimation and synthesis techniques are also still limited. Improvements are required to exploit the particular characteristics of light-field content and to increase compression efficiency, that will be a key factor in the development of immersive multimedia applications.

Part II of the manuscript is dedicated to our contributions on integral (or plenoptic) imaging content, that provides a dense sampling of the light-field in a narrow angle of view, with a challenging structure for compression. In Chapter 3 we propose an efficient integral image compression scheme where a residual integral image and an extracted view are encoded. The residual image is the difference between the original image and an image reconstructed from the view. An average BD-rate gain of 15.7% (up to 31.3%) over the HEVC anchor is reported. The coding performance largely depends on the configuration of several parameters. A robust iterative RDO process is modeled to select the best configuration, preserving optimal BD-rate gains. We show that the number of iterations can be

limited to reduce the runtime while preserving BD-rate gains. We study the impact of the position and size of the extracted view, and propose to use adaptive filtering techniques to further improve the compression efficiency. We finally propose to increase the number of extracted views in order to improve the quality of the reconstructed integral image, and therefore reduce the bitrate required to encode the residual image. Compression efficiency is increased with an average BD-rate gain of 22.2% (up to 31.1%) over the HEVC anchor, at the cost of an acceptable increase in runtime (i.e. less than 3 times the anchor runtime at the decoder side, and less than 1.5 times at the encoder side). Finally, average BD-rate gains are brought up to 28% when the proposed scheme is combined with the state-of-the-art Intra Block Copy method. This complete study results in a codec that offers a realistic coding performance vs runtime tradeoff.

In future work, the coding scheme should also be further evaluated with additional content. More advanced extraction methods using dense disparity maps can also be applied, and specific coding tools dedicated to the encoding of the residual image could also improve the performance.

Integral imaging cannot be used for applications where a large angle of view is required, such as Free Navigation for example, as it only captures the light-field under a narrow angle of view. In Part III, we also study the compression of Super Multi-View content, that provides a sparser sampling of the light-field but with a large baseline. In Chapter 4, we present a subjective quality evaluation of compressed Super Multi-View content on a light-field display system. The goal is to study the impact of compression at the display side in the specific case of light-field content. We provide some initial conclusions on the feasibility of a video service that would require rendering about 80 views. We first show that the bitrates required for encoding and rendering 80 views are realistic and coherent with future networks requirements to support 4K/8K, although some considerations on the tested content characteristics highlight the need for a better codec, in order to further improve the quality and avoid network overload. Preliminary experiments performed during this study lead to recommended coding configurations for Super Multi-View video content, particularly with groups of views (GOVs), that enable a compromise between memory limitations, coding efficiency and parallel processing. Some conclusions are also drawn on the amount of views to skip at the encoder, and to synthesize after the decoding, that is highly sequence-dependent. The ratio between coded and synthesized views depends on the quality of the synthesized views, hence is linked to the quality of the depth maps, the efficiency of the renderer, and the complexity of the scene. Apart from compression, view synthesis can introduce severe distortions, and affects the overall rendering scheme. Our results confirm that improvement of view synthesis and depth estimation algorithms is mandatory. Concerning the evaluation method and metric, results show that the PSNR remains able to reflect an increase or decrease in subjective quality for light-field content. However, depending on the ratio of coded and synthesized views, we have observed that the order of magnitude of the effective quality variation is biased by the PSNR, that is less tolerant to view synthesis artifacts than human viewers.

Experimental results depend on the test conditions and particularly on the tested content. As a consequence, future work should extend the evaluation towards additional content with different depth characteristics (e.g. camera arrangements) and encoding complexities. Subjective evaluations with a denser range of bitrate values could allow refining the boundaries between the ranges of bitrate values associated with each quality level, and a lower range could allow determining the lowest bitrate value possible for an acceptable quality. Using these denser ranges and limit values could allow finding a proper way to

evaluate objectively the quality of compressed and synthesized Super Multi-View content by weighting efficiently the PSNR for synthesized views or by using a more convenient metric. This could allow associating ranges of subjective qualities with ranges of objective values. Preliminary observations have also been initiated on the light-field conversion step in the display system and on the impact of compression on the perception of motion parallax. This should be further studied, as well as other specific aspects of light-field content such as the perception of depth or the angle of view for example.

In Chapter 5 we propose an inter-view reference picture structure adapted to Super Multi-View content with full motion parallax (horizontal and vertical view alignment). Its main features are the minimal distance between the coded and the reference views, and the use of both horizontal and vertical inter-view references. The proposed scheme is more efficient than every other tested state-of-the-art configuration. We also propose to improve the specific 3D-HEVC coding tools NBDV and IVMP in order to exploit both horizontal and vertical directions in a full parallax configuration, providing bitrate savings up to 4.2% over 3D-HEVC. The results of the methods proposed in this study demonstrate that exploiting efficiently both horizontal and vertical dimensions of full parallax Super Multi-View content at the coding structure level and at the coding tools level significantly improves the compression performance. In future work, other specific prediction tools can be improved by taking this aspect into account.

In Chapter 6 we assess how arc camera arrangements impact on the compression efficiency. We show that the direction of the disparity vectors is not the same in linear content (with always the same direction) and arc content (with two opposite directions possible). We compare the efficiency of existing coding technologies on linear and on arc camera arrangements, and results show no significant performance difference in our experimental conditions. Hence we propose perspectives to improve specifically the performance in the arc case (without degrading it for the linear case). We propose to improve the prediction of the DVs by taking into account this possibility of opposite directions. However, there is no significant improvement of the performance with the proposed tools as there are only few cases where the modifications have an impact.

However, the in-depth study of the encoding experiments strongly suggests that there is room for further improvements in other specific coding tools. In future work, other complementary specific aspects of the arc content can be taken into account, such as the rotation of parts of the pictures from one view to another for example.

In Chapter 7, we study the compression of Super Multi-View content targeting Free Navigation applications. We focus on a use case where all the views are encoded and sent to the decoder, and the user can interactively request to the decoder (and renderer) a point of view to be displayed (e.g. on a state-of-the-art 2D display). We first study the coding performances of state-of-the-art methods based on current encoders, regarding the tradeoff between two main criteria: compression efficiency (i.e. lowest bitrate possible) and degree of freedom (i.e. the ability for the user to change the viewpoint, that mainly depends on the decoder capability and the number of pictures to decode in order to display one). Secondly, in an appendix chapter, we propose a Free Navigation coding scheme that performs redundant encodings, thus allowing the user to shift the viewpoint without decoding additional views, in order to target a tradeoff in terms of rate-distortion performance.

In future work, the proposed method should be adapted to a flexible coding structure, e.g. with multiple inter-view and temporal references. Different navigation conditions offering several degrees of freedom to the user should be tested. The tradeoff between the

encoding complexity and the performance should also be studied, in particular considering the different possible iterations on quantization parameters and references orders for example. From a general point of view, our work provides hints and conclusions on several aspects of light-field technologies for future immersive applications. Several capture and display systems are emerging with different characteristics, and with many similarities between the existing formats for the resulting content, as in theory they are all different ways of storing and representing information sampled from the light-field of a scene. In practice, there are however many differences, and converting content from one format to another is not straightforward or trivial. The acquisition device and the format of the content have a very strong impact on the rest of the processing chain and on the performance of the complete light-field system from capture to display. Therefore the choice of capture, representation, and storage formats should not only be driven by compression/coding performances but should also depend on the target application. Coding technologies are proposed and are available to provide a tradeoff between compression efficiency and available features, as for example illustrated for Free Navigation applications with the level of freedom provided to the user. Another example is the display scalability feature in Integral Imaging coding schemes, where it is possible to decode only an extracted 2D view when the integral image cannot be decoded or displayed.

The conclusions drawn from our experimental results emphasize the feasibility of implementing immersive applications based on light-field. Current coding technologies can technically be used with only some structure and configuration improvements to represent light-field content in several ways (e.g. Super Multi-View, Integral Imaging or Point Clouds). Additionally, performances are realistic and our experiments did not show limitations that would completely prevent the use of light-field technologies. However, although the conclusions of the feasibility show a promising future for immersive applications based light-field technologies, some of our results emphasize the fact that there are still some bottlenecks and limitations that should be overcome, typically in cases like Super Multi-View with view synthesis. On this aspect, the modifications and new innovative coding schemes that we propose in this thesis provide significant improvements in compression efficiency. These results show that there is room for improvements for specific kinds of content, that will help for better representation and better coding of the light-field. The future work and contributions of the experts is expected to bring common standard formats that will drive the spread of light-field technologies on the consumer market to make the next generation of immersive video applications a milestone in the evolution of multimedia consumption.

Appendix: Proposed compression scheme for free navigation applications

When taking into account the expectations for future applications in terms of resolution, frame rate and number of views, the estimations based on our results in Chapter 7 report that improvements are required to reach a realistic tradeoff between compression efficiency and freedom of navigation. In this chapter, we propose as a perspective for future work, a coding scheme dedicated to Free Navigation that performs redundant encodings, thus allowing the user to shift the viewpoint without decoding additional pictures, in order to target a tradeoff between the rate-distortion performance and the freedom of navigation. Experimental results are not yet available for this method.

8.1 Proposed coding scheme

8.1.1 Coding structure

An example of the proposed coding structure is illustrated in Figure 8.1. In this scheme, the top pictures ($V_{0,0}$, ..., $V_{0,3}$) are I frames (i.e. intra coded, without inter frame prediction) and the rest of the pictures are called P' frames. P' frames are encoded N times with N different references (e.g. N = 3 in Fig. 8.1). The N encodings of a P' frame must provide a unique reconstructed/decoded frame (i.e. the N reconstructed frames are identical), in order to be able to decode the same picture, using any of the N reference views that is available at the decoder side. In terms of freedom of navigation, this allows the user to choose for the next frame to stay on the current view (i.e. no switching), or to switch to the left view (i.e. with current view index minus 1), or to switch to the right view (i.e. current view index plus 1). A basic way to do this is to encode lossless the difference between the reconstructed pictures. In the next section, an example at PU level is provided for the encoding and decoding of $V_{1,1}$, based on Figure 8.2.

8.1.2 Example with the basic method

In Fig. 8.2, $V_{1,1}$ is first encoded using $V_{0,1}$ as a reference. $V_{1,1}$ is the current frame, $V_{0,1}$ is the same view at previous POC and $V_{0,2}$ is the right view at previous POC. P_{org} is the current coding PU. P_{pred1} and P_{pred2} are predictor PUs respectively located in $V_{0,1}$ and $V_{0,2}$. For each PU in the current view $V_{1,1}$, a motion estimation is first performed in the first reference $V_{0,1}$ (in a given search window W_1), to find a predictor PU P_{pred1}, pointed by a motion vector MV_1. P_{res1} is the residual image corresponding to the difference between

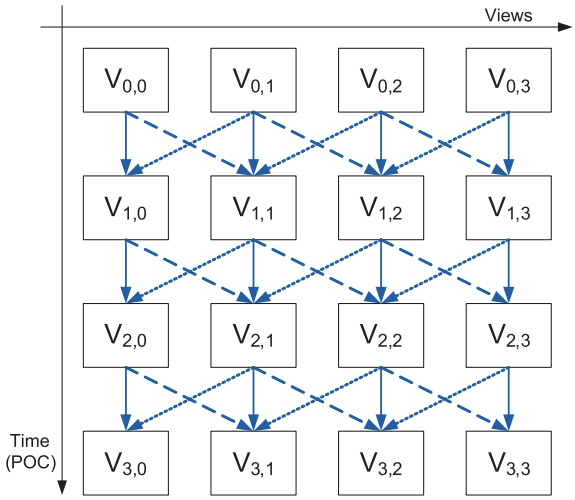


Figure 8.1: Proposed coding structure (example with N = 3)

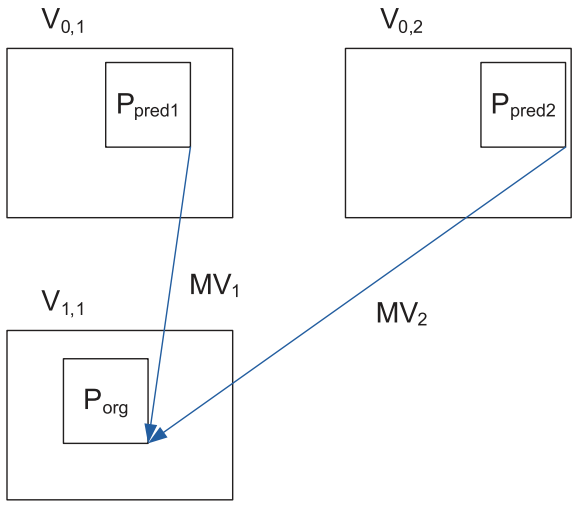


Figure 8.2: Illustration of the basic method with N = 2

Bitstream 1	Bitstream 2	...	Bitsream N
..., P_{res1}, MV1, ...	..., P_{res2}, MV2, ..., P_{diff2}, ...	...	..., PresN, MVN, ..., P_{diffN}, ...

Table 8.1: Syntax elements encoded (i.e. transmitted) in the basic example method

View index	POC	Type	References for N encodings		
			center	left	right
0	0	I	/	/	/
1			/	/	/
2			/	/	/
3			/	/	/
0	1	P'	$V_{0,0}$	/	$V_{0,1}$
1			$V_{0,1}$	$V_{0,0}$	$V_{0,2}$
2			$V_{0,2}$	$V_{0,1}$	$V_{0,0}$
3			$V_{0,3}$	$V_{0,2}$	/
0	2	P'	$V_{1,0}$	/	$V_{1,1}$
1			$V_{1,1}$	$V_{1,0}$	$V_{1,2}$
2			$V_{1,2}$	$V_{1,1}$	$V_{1,0}$
3			$V_{1,3}$	$V_{1,2}$	/
0	3	P'	$V_{2,0}$	/	$V_{2,1}$
1			$V_{2,1}$	$V_{2,0}$	$V_{2,2}$
2			$V_{2,2}$	$V_{2,1}$	$V_{2,0}$
3			$V_{2,3}$	$V_{2,2}$	/

Table 8.2: Coding order (i.e. order in the bitstream) for a Group Of 4×4 Views in the proposed example

the current and the predictor blocks (Eq. (8.1)).

$$P_{res1} = P_{org} - P_{pred1} \tag{8.1}$$

P_{res1} is then transformed, quantized and the resulting coefficients are encoded in the bitstream (see Table 8.1). For the reconstruction, the residual coefficients are first decoded, dequantized and inverse transformed, providing a residual P_{qres1}. This residual block is then summed to the predictor P_{pred1} to obtain the reconstructed PU P_{rec1}. This part corresponds to the state-of-the-art. The same operations are performed with $V_{0,2}$ as a reference, providing the following elements: P_{pred2}, MV_2, P_{res2}, and P_{rec2}. P_{diff2} is a difference block obtained by subtracting the two reconstructed pictures (Eq. (8.2)). P_{diff2} is encoded losslessy. Table 8.2 summarizes the coding order (i.e. order in the bitstream) and references for a Group Of 4×4 Views in our example.

$$P_{diff2} = P_{rec1} - P_{rec2} \tag{8.2}$$

At decoder side, two cases are possible in this basic example to decode $V_{1,1}$:
Either $V_{0,1}$ has been decoded and displayed (i.e. selected by the user at previous time instant) and is therefore available as a reference, or $V_{0,2}$ has been decoded and displayed and is available as a reference.

i) Decoding of $V_{1,1}$ when $V_{0,1}$ is available

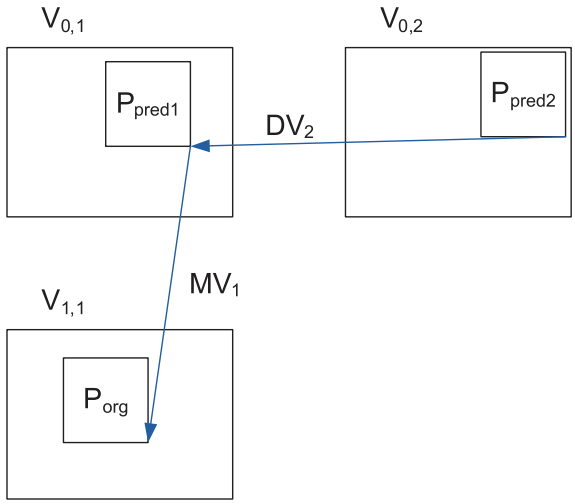


Figure 8.3: Illustration of the proposed method with N = 2

P_{qres1} is first decoded, followed by MV_1. P_{pred1} is found in the available reference view $V_{0,1}$ using MV_1. The decoded PU is obtained by summing the residual PU and the predictor PU.

$$\begin{aligned} P_{dec1} &= P_{qres1} + P_{pred1} \\ &= P_{rec1} \end{aligned} \tag{8.3}$$

ii) Decoding of $V_{1,1}$ when $V_{0,2}$ is available

P_{qres2} is first decoded, followed by MV_2 and P_{diff2}. P_{pred2} is found in the available reference view $V_{0,2}$ using MV_2. The decoded PU is obtained by summing the residual PU, the predictor PU, and the difference P_{diff2}.

$$\begin{aligned} P_{dec2} &= P_{qres2} + P_{pred2} + P_{diff2} \\ &= P_{rec2} + P_{diff2} \\ &= P_{rec1} \\ &= P_{dec1} \end{aligned} \tag{8.4}$$

In both cases, the decoded picture is the same. This method is expected to be inefficient, because the cost of the lossless encoding of the difference (P_{diffN}) between reconstructed views is expected to be very large.

8.1.3 Proposed method

We propose a method that allows to encode a given picture with N (e.g. N = 3) different references, providing N identical reconstructed images (as described in Sec 8.1.2). The principle is to encode, at PU level, the $N-1$ differences between the quantized original predictor PU taken from the main (first) reference and the quantized predictor PUs taken from the $N-1$ secondary (e.g. second and third) references. The quantization of the predictor block decreases the quality of the prediction, but it also decreases the cost of the differences between predictor PUs that have to be encoded losslessly.

This method, performed at PU level, is a coding mode that can compete with intra mode, i.e. depending on Rate Distortion Optimization (RDO), a given PU is encoded

either in intra mode, or either in inter mode using the N references jointly. As a description, we provide an example of encoding a view with N = 2 references (like for the basic method example), as illustrated in Figure 8.3. $V_{1,1}$ is the current frame, $V_{0,1}$ is the same view at previous POC and $V_{0,2}$ is the right view at previous POC. $\mathrm{P_{org}}$ is the current coding PU. $\mathrm{P_{pred1}}$ and $\mathrm{P_{pred2}}$ are predictor PUs respectively located in $V_{0,1}$ and $V_{0,2}$. $\mathrm{P_{qpred1}}$ and $\mathrm{P_{qpred2}}$ are the quantized/dequantized versions of $\mathrm{P_{pred1}}$ and $\mathrm{P_{pred2}}$ respectively (Eq (8.5) and Eq (8.6)).

$$\mathrm{P_{qpred1}} = \mathrm{Q^{-1}(Q(P_{pred1}))} \tag{8.5}$$

$$\mathrm{P_{qpred2}} = \mathrm{Q^{-1}(Q(P_{pred2}))} \tag{8.6}$$

For each PU in the current view $V_{1,1}$, a motion estimation is first performed in the first reference $V_{0,1}$ (in a given search window $\mathrm{W_1}$), to find a predictor PU $\mathrm{P_{pred1}}$, pointed by a motion vector $\mathrm{MV_1}$. $\mathrm{P_{pred1}}$ is then quantized/dequantized, providing the block $\mathrm{P_{qpred1}}$. $\mathrm{P_{res1}}$ is the residual image corresponding to the difference between the current and predictor block (Eq (8.7)).

$$\mathrm{P_{res1}} = \mathrm{P_{org}} - \mathrm{P_{qpred1}} \tag{8.7}$$

$\mathrm{P_{res1}}$ is then transformed, quantized and the resulting coefficients are encoded in the bitstream. For the reconstruction, the coefficients are first decoded, dequantized and inverse transformed, providing a residual $\mathrm{P_{qres1}}$ (Eq (8.8)). This residual block is then summed to the quantized predictor $\mathrm{P_{qpred1}}$ to obtain the reconstructed PU $\mathrm{P_{rec1}}$ (Eq (8.9)). D corresponds to the distortion of the reconstructed PU $\mathrm{P_{rec1}}$ against the original $\mathrm{P_{org}}$ (Eq (8.10)). This part corresponds to the state-of-the-art, except for the quantization of the predictor PU.

$$\mathrm{P_{qres1}} = \mathrm{T^{-1}(Q^{-1}(Q(T(P_{res1}))))} \tag{8.8}$$

$$\mathrm{P_{rec1}} = \mathrm{P_{qres1}} + \mathrm{P_{qpred1}} \tag{8.9}$$

$$D = |\mathrm{P_{org}} - \mathrm{P_{rec1}}|^2 \tag{8.10}$$

In the next step, $\mathrm{P_{org}}$ is predicted using the second reference $V_{0,2}$. A motion estimation is performed in a given search window $\mathrm{W_2}$, to find $\mathrm{P_{pred2}}$, pointed by a disparity vector $\mathrm{DV_2}$. $\mathrm{DV_2}$ is the vector that, when summed to $\mathrm{MV_1}$, corresponds to the motion from $\mathrm{P_{pred2}}$ to $\mathrm{P_{org}}$, as shown in Figure 8.3. $\mathrm{P_{pred2}}$ is then quantized/dequantized, providing the PU $\mathrm{P_{qpred2}}$. The difference e_2 between the quantized/dequantized predictor PUs is computed (Eq (8.11)), and encoded losslessly.

$$e_2 = \mathrm{P_{qpred1}} - \mathrm{P_{qpred2}} \tag{8.11}$$

For the reconstruction, the coefficients of $\mathrm{P_{qres1}}$ (already decoded as mentioned above) are summed to the quantized predictor $\mathrm{P_{qpred2}}$ and to the difference results e_2, to obtain the reconstructed PU $\mathrm{P_{rec2}}$, identical to $\mathrm{P_{rec1}}$ as shown in Eq (8.12).

$$\begin{aligned} \mathrm{P_{rec2}} &= \mathrm{P_{qres1}} + \mathrm{P_{qpred2}} + \mathrm{e_2} \\ &= \mathrm{P_{qres1}} + \mathrm{P_{qpred2}} + \mathrm{P_{qpred1}} - \mathrm{P_{qpred2}} \\ &= \mathrm{P_{qres1}} + \mathrm{P_{qpred1}} \\ &= \mathrm{P_{rec1}} \\ &= \mathrm{P_{dec1}} \end{aligned} \tag{8.12}$$

The best vectors $\mathrm{MV_1}$ and $\mathrm{DV_2}$ are selected by RDO. A cost $J = D + \lambda \times R$ is computed. D corresponds to the distortion (as previously mentioned). R corresponds to the rate

Bitstream 1	Bitstream 2	...	Bitsream N
..., P_{qres1}, MV_1, ...	..., DV_2, e_2, ...	...	..., DV_N, ..., e_N, ...

Table 8.3: Syntax elements encoded (i.e. transmitted) in the proposed method

required to encode the following elements: residual coefficients P_{qres1}, motion vector MV_1, disparity vector DV_2, and the difference e_2 (lossless). This cost is used for the competition between the (combinations of) motion/disparity vectors and with the intra mode. Table 8.3 summarizes the elements that are encoded (i.e. transmitted to the decoder).

In a first version of the method, all the combinations are tested at the encoder, i.e. for each tested motion vector MV_1, pointing from the first reference picture, a complete motion estimation is performed to find DV_2 (for the second reference picture). In another (simplified) version, motion estimation is first performed for the first reference, and MV_1 is selected based on the costs related to the first reference only (rate for MV_1 and for P_{qres1}). Then motion estimation for DV_2 is performed only for the selected MV_1 value. The first version is optimal but complex, while the second is expected to perform worse but with a large decrease of complexity (i.e. a decrease in runtime).

At decoder side, two cases are possible in our example to decode $V_{1,1}$:
Either $V_{0,1}$ has been decoded and displayed (i.e. selected by the user at previous time instant) and is therefore available as a reference, or $V_{0,2}$ has been decoded and displayed and is available as a reference. For both cases, P_{qres1} is first decoded, followed by MV_1. The next steps depend on the available reference.

i) Decoding of $V_{1,1}$ when $V_{0,1}$ is available

P_{pred1} is found in the available reference view $V_{0,1}$ using MV_1. P_{pred1} is then quantized/dequantized, providing the block P_{qpred1} (same operation than encoder side). The decoded PU is obtained by summing the residual PU and the dequantized predictor PU (Eq (8.13)).

$$P_{dec1} = P_{qres1} + P_{qpred1} \tag{8.13}$$

ii) Decoding of $V_{1,1}$ when $V_{0,2}$ is available

DV_2 and e_2 are decoded. P_{pred2} is found in the available reference view $V_{0,2}$ using MV_1 and DV_2 ($MV_1 + DV_2$). P_{pred2} is then quantized/dequantized, providing the block P_{qpred2} (same operation than encoder side). The decoded PU is obtained by summing the residual PU, the dequantized predictor PU, and the difference e_2. P_{dec2} is identical to P_{dec1} as shown in Eq (8.14).

$$\begin{aligned} P_{dec2} &= P_{qres1} + P_{qpred2} + e_2 \\ &= P_{qres1} + P_{qpred2} + P_{qpred1} - P_{qpred2} \\ &= P_{qres1} + P_{qpred1} \\ &= P_{rec1} \\ &= P_{dec1} \end{aligned} \tag{8.14}$$

There are two main differences between the proposed and the basic methods mentioned above. The first one is the estimation of a disparity vector (e.g. DV_2) that is summed to the original motion vector (e.g. MV_1) at decoder side, instead of estimating a second motion vector (e.g. MV_2). The second one is the subtraction performed between quantized predictor PUs instead of reconstructed PUs/pictures. In the following we provide

additional comments about the tuning of this method. Several versions of the method can provide a tradeoff between complexity and performance (e.g. exhaustive search or not for the vectors). The Quantization Parameter (QP) used for the predictor PUs does not have to be the same as the QP used for the residual coefficients (although it is possible, but not optimal). Iterations on the QP values can be performed (with an increase expected in both performance and complexity). The computation of e_2, described above as the difference between the quantized predictor PUs, can in practice be more advanced than a basic subtraction (for example including an offset value than could be derived at decoder side to reduce the cost of e_2). Other iterations are possible to make the method optimal (i.e. with more performance but also more complexity here), as for example iterations on the first reference view index (e.g. performing the method with $V_{0,1}$ used first, then with $V_{0,2}$ used first).

8.2 Conclusion and perspectives

In this chapter, we propose a coding scheme that allows decoding the views requested by the user using several possible references, i.e. in several cases of navigation path requested by the user, when the available reference view at the decoder is not known in advance at the encoder.

In future work, the proposed coding method should be tested in different navigation conditions, offering different degrees of freedom to the user. The tradeoff between the encoding complexity and the performance should be studied, in particular considering the different possible iterations on quantization parameters and references orders for example.

List of publications

International conferences

[IC3] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux "Improved integral images compression scheme based on multi-view extraction" in Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, SPIE, 2016, p. 99710L.

[IC2] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Integral images compression scheme based on view extraction", in 23rd European Signal Processing Conference (EUSIPCO), EURASIP, 2015, p. 101-105.

[IC1] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Full parallax Super Multi-View video coding", in International Conference on Image Processing (ICIP), IEEE, 2014, p. 135-139.

National conferences

[NC2] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Schéma de compression d'images intégrales basé sur l'extraction de vues", in CORESA 2016.

[NC1] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Compression de contenu video Super Multi-Vue avec parallaxe horizontale et verticale", in CORESA 2014.

Journal paper and book chapter

[J2] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, P. T. Kovacs, V. Kiran Adhikarla, "Subjective evaluation of Super Multi-View compressed contents on high-end light-field 3D displays", in Signal Processing: Image Communication, Elsevier, 2015, vol. 39, p. 369-385.

[J1] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Full Parallax 3D Video Content Compression", in Novel 3D Media Technologies, Springer New York, 2015, p. 49-70.

MPEG meeting contributions

[M7] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "m39857 [FTV AHG] Integral images compression scheme based on view extraction - follow up of contribution m39036", in ISO/IEC JTC1/SC29/WG11 (Geneva, Switzerland), January 2017.

[M6] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "m39036 [FTV AHG] Integral images compression scheme based on view extraction - Some new results", in ISO/IEC JTC1/SC29/WG11 (Chengdu, China), October 2016.

[M5] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux , "m36377 [FTV AHG] Integral images compression scheme based on view extraction", in ISO/IEC JTC1/SC29/WG11 (Warsaw, Poland), July 2015.

[M4] A. Dricot, J. Jung, "m36686 [FTV AHG] Big Buck Bunny Flowers Anchor Coding Results" in ISO/IEC JTC1/SC29/WG11 (Warsaw, Poland), July 2015.

[M3] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux "m34991 [FTV AHG] EE3 A1 HEVC simulcast results", in ISO/IEC JTC1/SC29/WG11 (Strasbourg, France), October 2014.

[M2] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, P. T. Kovacs, and V. Kiran Adhikarla, "m35023 [FTV AHG] Impact of the view synthesis in a Super Multi-View coding scheme", in ISO/IEC JTC1/SC29/WG11 (Strasbourg, France), October 2014.

[M1] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "m35030 [FTV AHG] Encoding configurations comparison for Super Multi-View content", in ISO/IEC JTC1/SC29/WG11 (Strasbourg, France), October 2014.

Patents

[P4] A. Dricot, J. Jung, "Method of coding and decoding for Free Navigation", October 2016.

[P3] A. Dricot, J. Jung, "Method of coding and decoding for Free Navigation", August 2015.

[P2] A. Dricot, J. Jung, M. Cagnazzo, B. Pesquet, F. Dufaux, "Method of coding and decoding integral images using extrapolation", January 2015.

[P1] A. Dricot, J. Jung, "Method of coding and decoding integral images", May 2014.

Bibliography

[1] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging technologies for 3D video: creation, coding, transmission and rendering*. John Wiley & Sons, 2013.

[2] E. H. Adelson and J. R. Bergen, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[3] A. Lumsdaine and T. Georgiev, "Full resolution lightfield rendering," *Indiana University and Adobe Systems, Tech. Rep*, 2008.

[4] L. Chiariglione, "[n16153] ad hoc groups established at mpeg 115," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11.

[5] M. P. Tehrani, T. Senoh, M. Okui, K. Yamamoto, N. Inoue, and T. Fujii, "[m31095][FTV AHG] use cases and application scenarios for Super Multiview Video and Free-Navigation," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2013.

[6] https://jpeg.org/items/20150320_pleno_summary.html/. Accessed: 2016-08-10.

[7] "[n16352/n72033] Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11 and WG1, 2016.

[8] https://vr.google.com/jump/. Accessed: 2016-08-05.

[9] https://gopro.com/odyssey/. Accessed: 2016-08-05.

[10] http://www.lytro.com/. Accessed: 2016-08-05.

[11] https://www.raytrix.de/. Accessed: 2016-07-29.

[12] www.samsung.com/global/galaxy/gear-vr/. Accessed: 2016-08-05.

[13] https://www3.oculus.com/en-us/rift/. Accessed: 2016-08-05.

[14] www.holografika.com/. Accessed: 2016-08-05.

[15] https://www.youtube.com/channel/UCzuqhhs6NWbgTzMuM09WKDQ/. Accessed: 2016-08-05.

[16] https://www.facebook.com/help/851697264925946/. Accessed: 2016-08-05.

[17] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[18] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *Communications Magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[19] J.-R. Ohm, "Overview of 3D video coding standardization," 2013.

[20] M. Wien, *High Efficiency Video Coding*. Springer, 2015.

[21] www.itu.int/rec/T-REC-H.261/e. Accessed: 2016-08-22.

[22] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, 2012.

[23] J. Jung and G. Laroche, "Competition-based scheme for motion vector selection and coding," *ITU-T SG16/Q6 Doc. VCEG-AC06*, 2006.

[24] G. Laroche, J. Jung, and B. Pesquet-Popescu, "RD optimized coding for motion vector predictor selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1247–1257, 2008.

[25] P. Helle, S. Oudin, B. Bross, D. Marpe, M. O. Bici, K. Ugur, J. Jung, G. Clare, and T. Wiegand, "Block merging for quadtree-based partitioning in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1720–1731, 2012.

[26] T. M. . of 3D-HEVC and MV-HEVC, "Chen, ying and tech, gerhard and wegner, krzysztof and yea, sehoon," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2015.

[27] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Electronic Imaging*, pp. 93–104, International Society for Optics and Photonics, 2004.

[28] E. G. Mora, J. Jung, M. Cagnazzo, and B. Pesquet-Popescu, "Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1554–1565, 2014.

[29] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, "[N16276] Algorithm description of Joint Exploration Test Model 3 (JEM3)," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2016.

[30] F. Pereira and E. A. Da Silva, "Efficient plenoptic imaging representation: Why do we need it?," in *International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2016.

[31] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an efficient representation of visual reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, 2016.

[32] https://home.otoy.com/otoy-demonstrates-first-ever-light-field-capture-for-vr/. Accessed: 2016-09-22.

[33] https://theta360.com/en/. Accessed: 2016-08-08.

[34] http://kodakpixpro.com/Americas/cameras/actioncam/sp360/. Accessed: 2016-08-05.

[35] http://www.360rize.com/. Accessed: 2016-09-25.

[36] http://www.thinktankteam.info/beyond. Accessed: 2016-12-02.

[37] https://www.jauntvr.com/. Accessed: 2016-09-25.

[38] https://hhi.fraunhofer.de/abteilungen/vit/technologien-und-loesungen/capture/panorama-videoproduktion-in-uhd/omnicam-360.html/. Accessed: 2016-09-25.

[39] http://www.video-stitch.com/. Accessed: 2016-09-25.

[40] M. Tanimoto, T. Fujii, T. Senoh, T. Aoki, and Y. Sugihara, "[M12338] Test sequences with different camera arrangements for Call for Proposals on multi-view video coding," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2005.

[41] M. Tanimoto, T. Fujii, and N. Fukushima, "[M15378] 1D Parallel test sequences for MPEG-FTV," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2008.

[42] https://www.youtube.com/watch?vā1oDxu/. Accessed: 2016-09-26.

[43] G. Lippmann, "Epreuves reversibles donnant la sensation du relief," *J. Phys. Theor. Appl.*, vol. 7, no. 1, pp. 821–825, 1908.

[44] T. Georgiev and A. Lumsdaine, "Focused plenoptic camera and rendering," *Journal of Electronic Imaging*, vol. 19, no. 2, p. 021106, 2010.

[45] http://www.tgeorgiev.net/. Accessed: 2016-09-27.

[46] M. Rerabek, T. Bruylants, T. Ebrahimi, F. Pereira, and P. Schelkens, "Call for proposals and evaluation procedure," in *ICME 2016 Grand Challenge: Light-Field Image Compression*.

[47] I. A. Salomie, A. Munteanu, A. Gavrilescu, G. Lafruit, P. Schelkens, R. Deklerck, and J. Cornelis, "MESHGRID - A compact, multiscalable and animation-friendly surface representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 950–966, 2004.

[48] M. P. Tehrani, T. Senoh, M. Okui, K. Yamamoto, N. Inoue, and T. Fujii, "[M31103][FTV AHG] Introduction of Super Multiview Video systems for requirement discussion," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2013.

[49] J. Arai, H. Kawai, and F. Okano, "Microlens arrays for integral imaging system," *Applied optics*, vol. 45, no. 36, pp. 9066–9078, 2006.

[50] R. Martinez-Cuenca, G. Saavedra, M. Martinez-Corral, and B. Javidi, "Progress in 3-D multiperspective display by integral imaging," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1067–1077, 2009.

[51] "http://gl.ict.usc.edu/Research/3DDisplay/." Accessed: 2016-12-27.

[52] "http://www.holymine3d.com/prod_en/prod03e.html." Accessed: 2016-12-27.

[53] T. Yendo, T. Fujii, M. Tanimoto, and M. Panahpour Tehrani, "The Seelinder: Cylindrical 3D display viewable from 360 degrees," *Journal of visual communication and image representation*, vol. 21, no. 5, pp. 586–594, 2010.

[54] J. F. O. Lino, "2D image rendering for 3D holoscopic content using disparity-assisted patch blending," *Thesis to obtain the Master of Science Degree*, 2013.

[55] http://www.xbox.com/fr-FR/xbox-one/accessories/kinect-for-xbox-one#fbid=9SxsePniRdZ. Accessed: 2016-11-19.

[56] M. El Gheche, *Proximal methods for convex minimization of Phi-divergences: application to computer vision.* PhD thesis, Paris Est, 2014.

[57] J. Lu, K. Zhang, G. Lafruit, and F. Catthoor, "Real-time stereo matching: a cross-based local approach," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 733–736, IEEE, 2009.

[58] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.

[59] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International journal of computer vision*, vol. 27, no. 2, pp. 161–195, 1998.

[60] O. Stankiewicz, K. Wegner, M. Tanimoto, and M. Domanski, "[M31518] Enhanced Depth Estimation Reference Software (DERS)," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2013.

[61] K. Wegner, O. Stankiewicz, T. Senoh, G. Lafruit, and M. Tanimoto, "[N16522] FTV software summary," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2016.

[62] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole-filling method using depth based in-painting for view synthesis in free viewpoint television (FTV) and 3D video," in *Picture Coding Symposium*, 2009.

[63] A. I. Purica, E. G. Mora, B. Pesquet-Popescu, M. Cagnazzo, and B. Ionescu, "Multiview plus depth video coding with temporal prediction view synthesis," *Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 2, pp. 360–374, 2016.

[64] O. Stankiewicz, K. Wegner, M. Tanimoto, and M. Domanski, "[M31520] Enhanced view synthesis reference software (VSRS) for Free-viewpoint Television," 2013.

[65] G. Lafruit, M. Domanski, K. Wegner, T. Grajek, T. Senoh, J. Jung, P. Tamás Kovács, P. Goorts, L. Jorissen, A. Munteanu, *et al.*, "New visual coding exploration in MPEG: Super Multi-View and Free Navigation in Free viewpoint TV," *IS&T Electronic Imaging: Stereoscopic Displays and Applications XXVII Proceedings*, 2016.

[66] M. Forman, A. Aggoun, and M. McCormick, "Compression of integral 3D TV pictures," in *Fifth International Conference on Image Processing and its Applications*, pp. 584–588, IET, 1995.

[67] A. Aggoun, "A 3D DCT compression algorithm for omnidirectional integral images," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, pp. II–II, IEEE, 2006.

[68] M. C. Forman, A. Aggoun, and M. McCormick, "A novel coding scheme for full parallax 3D-TV pictures," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1997*, vol. 4, pp. 2945–2947, IEEE, 1997.

[69] M. Forman and A. Aggoun, "Quantisation strategies for 3D-DCT-based compression of full parallax 3D images," in *Sixth International Conference on Image Processing and Its Applications*, pp. 32–35, IET, 1997.

[70] E. Elharar, A. Stern, O. Hadar, and B. Javidi, "A hybrid compression method for integral images using discrete wavelet transform and discrete cosine transform," *Journal of display technology*, vol. 3, no. 3, pp. 321–325, 2007.

[71] A. Aggoun, "Compression of 3D integral images using 3D wavelet transform," *Journal of Display Technology*, vol. 7, no. 11, pp. 586–592, 2011.

[72] S. Kishk, H. E. M. Ahmed, and H. Helmy, "Integral images compression using discrete wavelets and PCA," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 4, no. 2, pp. 65–78, 2011.

[73] H. H. Zayed, S. E. Kishk, and H. M. Ahmed, "3D wavelets with SPIHT coding for integral imaging compression," *International Journal of Computer Science and Network Security*, vol. 12, no. 1, p. 126, 2012.

[74] R. Olsson, M. Sjostrom, and Y. Xu, "A combined pre-processing and H. 264 compression scheme for 3D integral images," in *International Conference on Image Processing (ICIP)*, pp. 513–516, IEEE, 2006.

[75] S. Yeom, A. Stern, and B. Javidi, "Compression of 3D color integral images," *Optics express*, vol. 12, no. 8, pp. 1632–1642, 2004.

[76] P. Yan and Y. Xianyuan, "Integral image compression based on optical characteristic," *IET Computer Vision*, vol. 5, no. 3, pp. 164–168, 2011.

[77] H.-H. Kang, D.-H. Shin, and E.-S. Kim, "Compression scheme of sub-images using Karhunen-Loeve transform in three-dimensional integral imaging," *Optics Communications*, vol. 281, no. 14, pp. 3640–3647, 2008.

[78] J. Dick, H. Almeida, L. D. Soares, and P. Nunes, "3D holoscopic video coding using MVC," in *International Conference on Computer as a Tool (EUROCON)*, pp. 1–4, IEEE, 2011.

[79] S. Shi, P. Gioia, and G. Madec, "Efficient compression method for integral images using multi-view video coding," in *International Conference on Image Processing (ICIP)*, pp. 137–140, IEEE, September 2011.

[80] S. Adedoyin, W. A. C. Fernando, and A. Aggoun, "A joint motion & disparity motion estimation technique for 3D integral video compression using evolutionary strategy," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 732–739, 2007.

[81] S. Adedoyin, W. A. C. Fernando, A. Aggoun, and K. Kondoz, "Motion and disparity estimation with self adapted evolutionary strategy in 3D video coding," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1768–1775, 2007.

[82] C. Conti, J. Lino, P. Nunes, L. D. Soares, and P. L. Correia, "Improved spatial prediction for 3D holoscopic image and video coding," in *European Signal Processing Conference (EUSIPCO)*, EURASIP, 2011.

[83] C. Conti, J. Lino, P. Nunes, L. D. Soares, and P. Lobato Correia, "Spatial prediction based on self-similarity compensation for 3D holoscopic image and video coding," in *International Conference on Image Processing (ICIP)*, pp. 961–964, IEEE, 2011.

[84] C. Conti, P. Nunes, and L. D. Soares, "New HEVC prediction modes for 3D holoscopic video coding," in *International Conference on Image Processing (ICIP)*, pp. 1325–1328, IEEE, 2012.

[85] C. Conti, J. Lino, P. Nunes, and L. D. Soares, "Spatial and temporal prediction scheme for 3D holoscopic video coding based on H. 264/AVC," in *19th International Packet Video Workshop (PV)*, pp. 143–148, IEEE, 2012.

[86] L. F. Lucas, C. Conti, P. Nunes, L. D. Soares, N. M. Rodrigues, C. L. Pagliari, E. A. da Silva, and S. M. de Faria, "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC," in *European Signal Processing Conference (EUSIPCO)*, pp. 11–15, EURASIP, 2014.

[87] C. Conti, P. Nunes, and L. D. Soares, "Using self-similarity compensation for improving inter-layer prediction in scalable 3D holoscopic video coding," in *SPIE Optical Engineering+ Applications*, pp. 88561K–88561K, International Society for Optics and Photonics, 2013.

[88] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–4, IEEE, 2016.

[89] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–4, IEEE, 2016.

[90] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. da Silva, and L. Soares, "Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–4, IEEE, 2016.

[91] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–4, IEEE, 2016.

[92] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–4, IEEE, 2016.

[93] C. Conti, P. Nunes, and L. D. Soares, "Inter-layer prediction scheme for scalable 3-D holoscopic video coding," *Signal Processing Letters*, vol. 20, no. 8, pp. 819–822, 2013.

[94] T. Georgiev, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala, "Spatio-angular resolution tradeoffs in integral photography," *Rendering Techniques*, vol. 2006, pp. 263–272, 2006.

[95] F. Bossen, "[JCT-VC L1100] Test conditions and software reference configurations," 2013.

[96] http://www.ffmpeg.org/. Accessed: 2016-11-24.

[97] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, 2001.

[98] G. Alves, F. Pereira, and E. A. Da Silva, "Light field imaging coding: Performance assessment methodology and standards benchmarking," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–6, IEEE, 2016.

[99] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," in *European Signal Processing Conference (EUSIPCO)*, pp. 898–902, EURASIP, 2016.

[100] N. Wiener, *Extrapolation, interpolation and smoothing of stationary-time-series*. New York, NY: Wiley Eds, 1949.

[101] J. Boyce, J. Chen, Y. Chen, D. Flynn, M. Hannuksela, M. Naccari, C. Rosewarne, K. Sharman, J. Sole, G. Sullivan, *et al.*, "Edition 2 Draft Text of High Efficiency Video Coding (HEVC), Including Format Range (REext), Scalability (SHVC), and Multi-View (MV-HEVC) Extensions," *document JCTVC-R1013*, 2014.

[102] M. Rerabek, E. Upenik, and T. Ebrahimi, "JPEG backward compatible coding of omnidirectional images," in *SPIE Optical Engineering+ Applications*, pp. 99711O–99711O, International Society for Optics and Photonics, 2016.

[103] J. Theytaz, L. Yuan, D. McNally, and T. Ebrahimi, "Towards an animated JPEG," in *SPIE Optical Engineering+ Applications*, pp. 99711X–99711X, International Society for Optics and Photonics, 2016.

[104] E. Bosc, P. Le Callet, L. Morin, and M. Pressigout, "Visual quality assessment of synthesized views in the context of 3D-TV," in *3D-TV System with Depth-Image-Based Rendering*, pp. 439–473, Springer, 2013.

[105] D. Howard, M. Green, R. Palaniappan, and N. Jayant, "Visibility of digital video artifacts in stereoscopic 3d and comparison to 2d," in *SMPTE Conferences*, vol. 2010, pp. 1–15, Society of Motion Picture and Television Engineers, 2010.

[106] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-D synthesized view assessment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1332–1343, 2011.

[107] P. T. Kovács, A. Nagy, Z.and Barsi, V. K. Adhikarla, and R. Bregovic, "Overview of the applicability of H.264/MVC for real-time light-field applications," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2014.

[108] https://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/. Accessed: 2016-09-26.

[109] P. T. Kovacs, A. Fekete, K. Lackner, V. K. Adhikarla, A. Zare, and T. Balogh, "[M35721][FTV AHG] Big Buck Bunny light-field test sequences," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2015.

[110] "http://www.blender.org/." Accessed: 2016-12-27.

[111] "http://www.autodesk.fr/products/3ds-max/overview." Accessed: 2016-12-27.

[112] K. Wegner, O. Stankiewicz, K. Klimaszewski, and M. Domański, "[M33243] FTV EE3: Compression of FTV video with circular camera arrangement," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2014.

[113] G. Tech, K.Wegner, Y. Chen, M. Hannuksela, and J. Boyce, "[M32661] MV-HEVC Draft Text 7," in *International Organisation For Standardisation*, ISO/IEC JTC 1/SC 29/WG 11, 2014.

[114] P. Hanhart and T. Ebrahimi, "Calculation of average coding efficiency based on subjective quality scores," *Journal of Visual Communication and Image Representation*, vol. 25, no. 3, pp. 555–564, 2014.

[115] Recommendation ITU-R BT.500-13, *Methodology for the subjective assessment of the quality of television pictures*, 2012.

[116] W. Chen, J. Fournier, M. Barkowsky, and P. Le Callet, "New requirements of subjective video quality assessment methodologies for 3DTV," in *Video Processing and Quality Metrics (VPQM)*, 2010.

[117] W. A. IJsselsteijn, H. De Ridder, and J. Vliegen, "Subjective evaluation of stereoscopic images: effects of camera parameters and display duration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 225–233, 2000.

[118] F. D. Simone, *Selected contributions on multimedia quality evaluation.* PhD thesis, 2012.

[119] M. G. Kendall, *Rank correlation methods.* London : Griffin, 1970.

[120] M. Tanimoto, K. Wegner, and G. Lafruit, "[M34604][AHG Report] AHG on FTV (Free-viewpoint Television)," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2015.

[121] "http://www.3d-contournet.eu/stsms/." Accessed: 2016-12-27.

[122] "http://www.bigbuckbunny.org/." Copyright 2008, Blender Foundation, Accessed: 2016-12-27.

[123] E. J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock, "Motion parallax as a determinant of perceived depth," *Journal of experimental psychology*, vol. 58, no. 1, p. 40, 1959.

[124] D. Rusanovskyy, F.-C. Chen, L. Zhang, and T. Suzuki, "[F1003] 3D-AVC Test Model 8," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2013.

[125] G. Sullivan, J. Boyce, Y. Chen, J.-R. Ohm, A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE Journal of selected topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, 2013.

[126] L. Zhang, Y. Chen, and M.Karczewicz, "[M24937] 3D-CE5.h related: Disparity vector derivation for multiview video and 3DV," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2012.

[127] G. Tech, K.Wegner, Y. Chen, and S. Yea, "[JCT3V-B1005] 3D-HEVC test model 2," in *International Organisation For Standardisation*, ITU-T SG 16 WP 3 & ISO/IEC JTC1/SC29/WG11, 2012.

[128] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.

[129] P. Merkle, A. Smolic, K. Muller, and 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, 2007.

[130] T. Chung, K. Song, and C.-S. Kim, "Compression of 2-D wide multi-view video sequences using view interpolation," in *International Conference on Image Processing (ICIP)*, pp. 2440–2443, IEEE, 2008.

[131] T.-Y. Chung, I.-L. Jung, K. Song, and C.-S. Kim, "Virtual view interpolation and prediction structure for full parallax multi-view video," in *Advances in Multimedia Information Processing - PCM*, vol. 5879, pp. 543–550, Springer, 2009.

[132] T.-Y. Chung, I.-L. Jung, K. Song, and C.-S. Kim, "Multi-view video coding with view interpolation prediction for 2D camera arrays," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5, pp. 474–486, 2010.

[133] A. Avci, J. De Cock, P. Lambert, R. Beernaert, J. De Smet, L. Bogaert, Y. Meuret, H. Thienpont, and H. De Smet, "Efficient disparity vector prediction schemes with modified P frame for 2D camera arrays," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 287–292, 2012.

[134] D. Rusanovsky, K. Muller, and A. Vetro, "[JCT3V-B11000] Common Test Conditions of 3DV Core Experiments," in *International Organisation For Standardisation*, ITU-T SG 16 WP 3 & ISO/IEC JTC1/SC29/WG11, 2012.

[135] J. Stankowski, Ł. Kowalski, J. Samelak, M. Domański, T. Grajek, and K. Wegner, "3D-HEVC extension for circular camera arrangements," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, IEEE, 2015.

[136] M. Karczewicz and R. Kurceren, "The SP-and SI-frames design for H. 264/AVC," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 637–644, 2003.

[137] N. Farber and B. Girod, "Robust H. 263 compatible video transmission for mobile access to video servers," in *International Conference on Image Processing (ICIP)*, vol. 2, pp. 73–76, IEEE, 1997.

[138] B. Motz, G. Cheung, and N.-M. Cheung, "Designing coding structures with merge frames for interactive multiview video streaming," in *International Conference on Multimedia & Expo Workshops (ICME)*, pp. 1–6, IEEE, 2016.

[139] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 744–761, 2011.

[140] T. Maugey and P. Frossard, "Interactive multiview video system with low decoding complexity," in *International Conference on Image Processing (ICIP)*, pp. 589–592, IEEE, 2011.

[141] T. Maugey and P. Frossard, "Interactive multiview video system with non-complex navigation at the decoder," *IEEE Transactions on Multimedia*, vol. 15, 2013.

[142] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain representation for interactive multiview imaging," *Transactions on Image Processing*, vol. 22, no. 9, pp. 3459–3472, 2013.

[143] A. De Abreu, L. Toni, T. Maugey, N. Thomos, P. Frossard, and F. Pereira, "Multiview video representations for quality-scalable navigation," in *Visual Communications and Image Processing Conference*, pp. 295–298, IEEE, 2014.

[144] M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu, "A differential motion estimation method for image interpolation in distributed video coding," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1861–1864, IEEE, 2009.

[145] T. Maugey and B. Pesquet-Popescu, "Side information estimation and new symmetric schemes for multi-view distributed video coding," *Journal of Visual Communication and Image Representation*, vol. 19, no. 8, pp. 589–599, 2008.

[146] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp, "Distributed video coding: selecting the most promising application scenarios," *Signal Processing: Image Communication*, vol. 23, no. 5, pp. 339–352, 2008.

[147] T. Maugey, J. Gauthier, M. Cagnazzo, and B. Pesquet-Popescu, "Evaluation of side information effectiveness in distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 12, pp. 2116–2126, 2013.

[148] C. Brites, J. Ascenso, and F. Pereira, "Side information creation for efficient wyner-ziv video coding: classifying and reviewing," *Signal Processing: Image Communication*, vol. 28, no. 7, pp. 689–726, 2013.

[149] T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion schemes for multiview distributed video coding," in *European Signal Processing Conference (EUSIPCO)*, pp. 559–563, EURASIP, 2009.

[150] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *Conference on Speech and Image Processing, Multimedia Communications and Services*, pp. 1–6, EURASIP, 2005.

[151] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," in *Conference on Advanced Video and Signal Based Surveillance*, pp. 593–598, IEEE, 2005.

[152] G. Petrazzuoli, T. Maugey, M. Cagnazzo, and B. Pesquet-Popescu, "A distributed video coding system for multi view video plus depth," in *Asilomar Conference on Signals, Systems and Computers*, pp. 699–703, IEEE, 2013.

[153] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Using distributed source coding and depth image based rendering to improve interactive multiview video access," in *International Conference on Image Processing (ICIP)*, pp. 597–600, IEEE, 2011.

[154] G. Petrazzuoli, *Temporal and inter-view interpolation for the improvement of the side information in distributed video coding*. PhD thesis, 2013.

[155] F. Pereira, C. Brites, J. Ascenso, and M. Tagliasacchi, "Wyner-Ziv video coding: a review of the early architectures and further developments," in *International Conference on Multimedia and Expo (ICME)*, pp. 625–628, IEEE, 2008.

[156] J. Ascenso and F. Pereira, "Advanced side information creation techniques and framework for Wyner-Ziv video coding," *Journal of Visual Communication and Image Representation*, vol. 19, no. 8, pp. 600–613, 2008.

[157] T. Clerckx, A. Munteanu, J. Cornelis, and P. Schelkens, "Distributed video coding with shared encoder/decoder complexity," in *International Conference on Image Processing (ICIP)*, vol. 6, pp. VI–417, IEEE, 2007.

[158] "[N15733] Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation - update," in *International Organisation For Standardisation*, ISO/IEC JTC1/SC29/WG11, 2015.

Résumé technique de la thèse en français

Résumé

L'évolution des technologies vidéo permet de créer des expériences de plus en plus immersives. Cependant, les technologies 3D actuelles sont encore très limitées et offrent aux utilisateurs des situations de visualisation qui ne sont ni confortables ni naturelles. La prochaine génération de technologies vidéo immersives apparait donc comme un défi technique majeur, en particulier avec la prometteuse approche *light-field* (LF). Le light-field représente tous les rayons lumineux (c'est-à-dire dans toutes les directions) dans une scène. De nouveaux dispositifs d'acquisition permettant d'échantillonner une partie du light-field apparaissent, tels que des ensembles de caméras (e.g. Google Jump/GoPro Odyssey) ou des appareils photo plénoptiques basés sur des ensembles de micro-lentilles (e.g. Lytro Illum). Plusieurs sortes de systèmes d'affichage ciblent des applications immersives, comme les *Head Mounted Displays* (e.g. Samsung Gear VR, Oculus Rift) ou les écrans light-field basés sur la projection (e.g. Holovizio d'Holografika), et des applications cibles prometteuses existent déjà (e.g. la vidéo 360° est une première étape avant la réalité virtuelle). Depuis plusieurs années, le light-field a stimulé l'intérêt de plusieurs entreprises et institutions, par exemple dans des groupes comme MPEG et JPEG. Les contenus light-field ont des structures spécifiques et utilisent une quantité massive de données, ce qui représente un défi pour implémenter les futurs services. L'un des buts principaux de notre travail est d'abord de déterminer quelles technologies sont réalistes ou prometteuses. Cette étude est faite sous l'angle de la compression image et vidéo, car l'efficacité de la compression est un facteur clé pour mettre en place ces services light-field sur le marché. Dans un deuxième temps, on propose des améliorations et des nouveaux schémas de codage pour augmenter les performances de compression et permettre une transmission efficace des contenus light-field sur les futurs réseaux.

Introduction

L'évolution des technologies vidéo offre des expériences de plus en plus immersives aux utilisateurs. L'*Ultra High Definition* (UHD), avec les résolutions 4K et 8K, le *High Frame Rate* (HFR), le *High Dynamic Range* (HDR) et aussi le *Wide Color Gamut* (WCG) amènent progressivement la vidéo 2D aux limites de la perception du système visuel humain. Cependant, les technologies vidéo 3D actuellement disponibles sur le marché sont mal acceptées par les utilisateurs car elles sont encore très limitées et ne peuvent offrir des expériences suffisamment confortables.

La stéréoscopie se base sur l'utilisation de seulement deux vues (une pour chaque oeil) et ne permet donc pas la parallaxe de mouvement, c'est à dire qu'il n'est pas possible pour le spectateur de changer de point de vue (par exemple en bougeant devant l'écran pour obtenir plus d'informations sur la scène visualisée). Cet indice qui contribue à la perception du relief est pourtant un élément clé pour les applications immersives. De plus, l'utilisation de lunettes est source d'inconfort, et le conflit entre la distance d'accommodation (les yeux se focalisent sur l'écran) et la distance de convergence (les yeux convergent sur l'image de l'objet potentiellement devant ou derrière l'écran) donne une situation de visualisation qui n'est pas naturelle et qui peut causer des migraines et des fatigues oculaires (parfois nommées *cybersickness*). Les systèmes d'affichage auto-stéréoscopiques utilisent plus de deux vues (par exemple entre 8 et 30) mais sont limités par le manque de parallaxe de mouvement fluide. Les positions de visualisation qui permettent à l'utilisateur d'observer la scène convenablement (c'est à dire avec une perception correcte de la profondeur et sans artefact) sont restreintes à certaines zones appelées *sweet spots*. Ces stimuli de perception non-naturels sont des limitations sévères qui altèrent la qualité de l'expérience de visualisation et la rendent irréaliste.

La prochaine génération de technologies vidéo immersives apparait donc comme un défi technique majeur, en particulier avec la prometteuse approche dite *light-field*. Un light-field, ou champ de lumière, représente tous les rayons lumineux dans une scène, c'est à dire pour chaque point dans l'espace et dans toutes les directions. Il est donc fonction de deux angles (i.e. la direction du rayon) et trois coordonnées spatiales. Cette fonction à 5 dimensions est appelée la fonction plénoptique. D'un point de vue conceptuel, comme la vidéo 2D fournit un échantillonnage basique du light-field en offrant une vue d'une scène selon un angle donné, les périphériques d'acquisition light-field offrent un échantillonnage plus large et plus dense avec plusieurs vues de la scène (i.e. en capturant les rayons selon plusieurs directions).

Depuis plusieurs années maintenant, la représentation light-field est au centre de l'attention de beaucoup d'experts dans différentes entreprises et institutions du domaine des technologies vidéo. Des efforts sont faits pour comprendre et déterminer le potentiel des périphériques et des formats émergeants, par exemple dans des groupes comme MPEG, avec en particulier les groupes *Free Viewpoint Television* (FTV) et *Virtual Reality* (VR), comme JPEG avec *JPEG Pleno*, et plus récemment avec un groupe conjoint: *Joint ad hoc group for digital representations of light/sound fields for immersive media applications*. De nouvelles technologies émergent rapidement sur le marché. Des dispositifs de capture sont maintenant disponibles, avec des ensembles de caméras (e.g. Google Jump/GoPro Odyssey, Lytro Immerge) ou des caméras plénoptiques basées sur des ensembles de microlentilles (e.g. Lytro Illum, Raytrix). Plusieurs systèmes d'affichage ou de rendu ciblent les applications immersives, comme les visiocasques (ou *Head Mounted Display*, e.g. Samsung Gear VR, Oculus Rift), et les écrans light-field basés sur la projection (e.g. Holografika's Holovizio). De plus, des applications prometteuses existent déjà (e.g. la vidéo 360°, déjà implémentée par Youtube et Facebook, qui est une première étape avant la réalité virtuelle) ou sont en développement (e.g. vidéo 360° stéréoscopique, téléprésence immersive, *Free Navigation*, etc.). Les contenus light-field, images et vidéos, nécessaires pour créer ces expériences immersives ont des formats et des structures spécifiques, et requièrent une quantité massive de données, ce qui représente un défi pour les futures transmissions sur nos réseaux et pour implémenter les futurs services.

Le but principal de nos travaux est d'étudier la faisabilité de l'implémentation de nouveaux services light-field immersifs. Cette étude est faite sous l'angle de la compression

image et vidéo, car l'efficacité de compression est un facteur clé pour la mise en place de ces services. On cherche premièrement à déterminer quelles technologies et quels formats sont réalistes, et lesquels sont prometteurs pour la capture, le rendu, et le codage en considérant différentes applications cibles. On propose ensuite des améliorations des technologies de compression de l'état de l'art et des nouveaux schémas de codage dans le but d'améliorer les performances de compression et de permettre une transmission efficace des contenus light-field sur les futurs réseaux. La structure du manuscrit est organisée comme suit.

- Dans le Chapitre 1, on décrit certains principes de base de la compression d'image et de vidéo qui sont implémentés dans les standards actuels et qui sont utiles pour comprendre les contributions techniques décrites dans cette thèse.

- Le Chapitre 2 retranscrit le contexte de nos travaux en donnant une vue d'ensemble de certaines technologies light-field de l'état de l'art, de la capture au rendu, incluant plusieurs traitements intermédiaires. Ce chapitre met principalement l'accent sur l'imagerie intégrale (ou plénoptique) et le Super Multi-Vues (SMV), qui sont respectivement basés sur des ensembles de micro-lentilles ou de caméras, et sur lesquelles nos contributions techniques sont principalement ciblées.

- Dans le Chaptitre 3, on propose un schéma de compression d'images intégrales basé sur l'extraction de vues. On tire avantage du processus d'extraction de vue pour reconstruire un prédicteur fiable et créer une image intégrale résiduelle qui est encodée. On propose d'abord plusieurs méthodes itératives pour sélectionner le paramétrage le plus efficace, en utilisant un processus d'optimisation débit-distortion, pour éviter la recherche exhaustive. Des gains en temps d'exécution sont également obtenus en étudiant les interactions entre les différents paramètres. Dans un second temps, on détermine l'impact de la position et de la taille des patches utilisés pour l'extraction de vue sur la performance de compression. On propose d'améliorer la méthode avec des techniques de filtrage avancées. Des méthodes basées sur le filtrage de Wiener sont utilisées pour améliorer l'étape de reconstruction. La performance du schéma avec plusieurs vues extraites est étudiée. Finalement, le comportement de cette méthode mise en compétition ou en collaboration avec des méthodes de l'état de l'art est étudié.

- Dans le Chapitre 4, on présente une évaluation de qualité subjective de contenu vidéo SMV compressé sur des écrans light-field. En effet, alors que la compréhension profonde des interactions entre compression et rendu présente un intérêt fondamental, évaluer la qualité de rendu des contenus light-field représente encore aujourd'hui un défi technique. Le but principal de cette étude est de déterminer l'impact de la compression sur la qualité perçue pour les contenus et les écrans light-field. A notre connaissance, les travaux présentés dans ce chapitre sont les premiers à montrer de telles expérimentations subjectives et à rapporter ce type de résultats.

- Le Chapitre 5 est dédié à la compression de contenu SMV avec parallaxe horizontale et verticale (*full parallax*), par exemple filmé avec des rigs de caméras 2D (où les caméras sont alignées horizontalement et verticalement). Les extensions multi-vues des encodeurs actuels sont adéquates pour encoder du contenu avec parallaxe horizontale uniquement, et doivent donc être modifiées et adaptées pour le full parallax. On propose d'abord un schéma de prédiction inter-vues qui exploite les dimensions

horizontales et verticales au niveau de la structure de codage. On propose ensuite des améliorations au niveau des outils de codage, basées sur la prédiction inter-vues.

- Le Chapitre 6 rapporte les résultats d'une étude centrée sur l'impact de l'utilisation d'ensemble de caméras alignées en arc (i.e. à la place des alignements linéaires habituels) sur la performance de compression. Les performances des technologies de codage actuelles sur l'arc et le linéaire sont d'abord comparées. On propose ensuite des améliorations spécifiques pour le cas en arc.

- Dans le Chapitre 7, on étudie la compression de contenu SMV ciblant des applications de *Free Navigation* (FN). On se concentre sur des applications où toutes les vues sont encodées et transmises au décodeur, et l'utilisateur sélectionne interactivement une vue à afficher (par exemple sur un écran 2D classique). On compare d'abord les performances des méthodes de codage de l'état de l'art. L'évaluation de la performance est basée sur le compromis entre l'efficacité de la compression (i.e. débit le plus bas possible) et le degré de liberté (i.e. la capacité pour l'utilisateur de changer librement de point de vue, qui dépend principalement des capacités du décodeur et du nombre d'images à décoder pour en afficher une). On propose finalement un schéma de compression avec des encodages redondants, permettant à l'utilisateur de changer de point de vue sans décoder de vues additionnelles.

- Les conclusions et les perspectives sont finalement dressées dans le Chapitre 8, suivi d'une liste des publications résultant des travaux présentés dans le manuscrit.

Dans ce chapitre, chaque section est associée à un chapitre listé ci-dessus, dont elle donne un résumé technique en français.

1 Quelques principes de la compression d'image et de vidéo

Dans le Chapitre 1, on revient sur certains principes de base de la compression d'image et de vidéo multi-vues qui sont implémentés dans les standards actuels et qui sont utiles pour comprendre les contributions techniques décrites dans le manuscrit. *High Efficiency Video Coding* (HEVC) a été normalisé en 2013. C'est un standard de compression (développé en partie au sein du groupe MPEG) qui succède au très répandu H.264/AVC. Ces deux standards sont basés sur des schémas de codage hybride (Figure 1). Parmi les caractéristiques clés de ce type de schéma, on peut citer premièrement le découpage de l'image en blocs. HEVC offre un partionnement hiérarchique (*quad-tree partitioning*) à plusieurs niveaux: codage (*Coding Units* ou CU), prédiction (*Prediction Units* ou PU) et transformée (*Transform Units* ou TU). Deuxièmement, ces blocs peuvent être codés soit avec un mode de prédiction intra, soit avec un mode de prédiction inter. En mode intra, un bloc est prédit en dérivant les valeurs des pixels voisins dans l'image courante, qui ont déjà été codés et décodés. En mode inter, un bloc prédicteur est trouvé dans une autre image de référence (prise à un autre instant). Un vecteur de mouvement (*Motion Vector*, MV) qui représente le déplacement entre le bloc courant et le bloc prédicteur est alors transmis au décodeur.

Le format *Multi-View plus Depth* (MVD) est un format composé de plusieurs séquences représentant la même scène capturée selon différents angles de vues. Ce sont les textures. Il est également composé des cartes de profondeur (*Depth Maps* en anglais) qui sont des images en niveau de gris représentant la profondeur, c'est à dire la distance des objets par rapport à la caméra. À partir de ces vues (textures et cartes de profondeur), il est

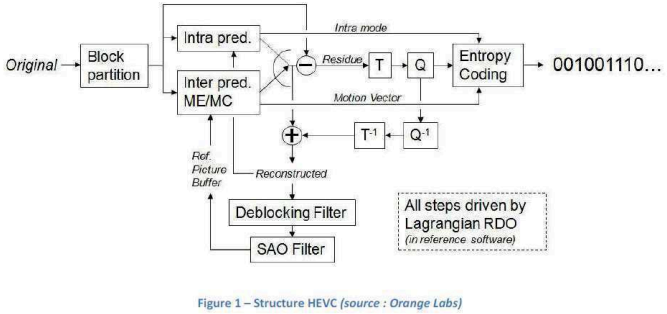


Figure 1 – Structure HEVC *(source : Orange Labs)*

Figure 1: Diagramme en bloc du codage vidéo hybride

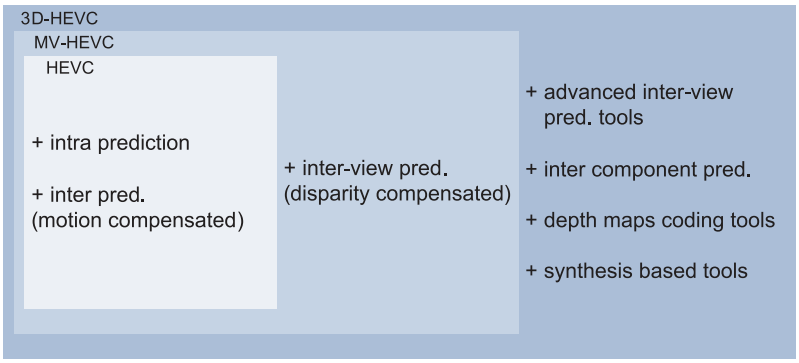


Figure 2: HEVC, MV-HEVC et 3D-HEVC

possible de synthétiser des vues intermédiaires qui n'ont pas été capturées par une caméra par exemple.

MV-HEVC et 3D-HEVC sont des extensions du codeur HEVC. MV-HEVC intègre des éléments de syntaxes additionnels qui permettent d'activer la prédiction inter-vues. Le principe est similaire à la prédiction temporelle, avec une image de référence prise au même instant dans une autre vue, plutôt qu'à un instant différent dans la même vue pour le cas temporel. Le vecteur utilisé pour la prédiction représente ici, non plus un mouvement, mais la disparité entre les deux images (*Disparity Vector*, DV). 3D-HEVC est une extension conçue pour le format MVD, qui intègre des outils de prédiction inter-vues avancés, des outils de prédiction inter-composantes (e.g. pour prédire le partitionnement de la carte de profondeur à partir de celui de la texture), ainsi que des outils spécifiques au codage des cartes de profondeurs et des outils basés sur la synthèse de vues.

Comme le montre la Figure 2, HEVC et ces deux extensions sont imbriqués. Cette construction se réflète dans l'analyse des performances. De manière générale, HEVC offre 50% de réduction de débit face à H.264 pour des séquences 2D. MV-HEVC apporte environ 30% de gains supplémentaires face à HEVC, dans un cas commun avec 2 textures à encoder. Le gain s'élève à 70% lorqu'on le mesure uniquement sur la vue additionnelle. Finalement, 3D-HEVC amène 20% de gains environ sur MV-HEVC dans un cas commun avec 3 textures, 3 cartes de profondeur associées, et 6 vues synthétisées.

De par sa performance élevée, sa reconnaissance, et son utilisation répandue au sein de la communauté scientifique et industrielle, l'encodage basé sur HEVC et ses extensions est de facto la référence pour les travaux présentés dans ce manuscrit.

2 Vue d'ensemble des technologies light-field

2.1 Qu'est-ce que le *light-field*?

Le *light-field* (LF) est constitué de tous les rayons de lumière qui traverse une scène donnée, c'est à dire à tous les points de l'espace et dans toutes les directions. Il peut donc être représenté par une fonction à 5 dimensions (trois coordonnées spatiales et deux angles). Cette fonction s'appelle la fonction plénoptique.

S'il n'est pas possible en pratique de capturer l'infinité de rayons représentant la totalité du *light-field*, de nombreuses technologies d'acquisition et d'affichage émergent et permettent de l'échantillonner et de le représenter. Ces technologies se basent principalement sur la capture d'images d'une scène selon un grand nombre d'angles de vue. Différents formats existent de la capture à l'affichage correspondant à différentes applications cibles.

2.2 Acquisition et formats

Les différentes technologies d'acquisition ligth-field peuvent être répertoriées en plusieurs sous-catégories. On parle d'abord d'acquisition divergente, lorsqu'une ou plusieurs caméras sont disposées de manière à capturer plusieurs vues de la scène de façon omnidirectionnelle, c'est à dire autour du dispositif (e.g. Ricoh theta, Gopro Odyssey, Lytro Immerge). Le contenu résultant peut alors être projeté sur une large image, visant des applications de vidéo 360° et de réalité virtuelle (VR).

Le pendant de cette technologie est le Super Multi-Vue (SMV) convergent. Un ensemble de caméras est ici disposé autour ou en face de la scène. Ces caméras peuvent être alignées sur une ou deux dimensions (offrant une parallaxe de mouvement respectivement horizontale ou totale), suivant un arrangement linéaire, circulaire (en arc) ou même non structuré. Le contenu résultant est un ensemble de vues de la scène (e.g. format multi-vues ou MVD), pouvant être visionné par des systèmes d'affichage light-field (*SMV displays*).

Les caméras plénoptiques permettent également de capturer le light-field. Des dispositifs sont déjà disponibles sur le marché, en général sous le nom de photographie plénoptique (e.g. Lytro Illum, Raytrix). Une caméra/un capteur est utilisé avec un ensemble de micro-lentilles. Elle permet de capturer une image plénoptique (ou image intégrale), composée de Micro-Images (MIs), chacune produite par l'une des micro-lentilles. Chaque MI contient de l'information visuelle capturée selon plusieurs angles de vues. À l'heure actuelle, ces dispositifs sont principalement utilisés pour la photographie (i.e. image fixe) mais de nouvelles applications émergent par exemple avec l'apparition des systèmes *Lytro Cinema*.

D'autres formats et représentations existent pour le light-field, par exemple les nuages de points (*Point clouds*), et *meshes 3D*. Ces formats sont basés sur la géométrie, et on les appelle parfois également des formats objets. Ils sont maintenant complétement intégrés dans l'étude des représentations light-field.

2.3 Rendu et affichage

Plusieurs systèmes d'affichages peuvent être la cible d'applications liées au light-field. Premièrement, les displays 2D classiques sont à considérer, pour des applications comme la vidéo 360 (déjà implémentée par Youtube ou Facebook par exemple), ou comme la Free Navigation (permettant à l'utilisateur de se déplacer dans la scène de manière interactive).

Les écrans SMV, dits aussi *light-field displays systems*, sont des dispositifs basés en général sur la projection, utilisant des écrans directifs, qui permettent d'afficher un contenu

en 3D à partir d'un grand nombre de vues, et de le visualiser sans lunette. On peut citer comme principal exemple les écrans Holovizio implémentés par Holografika.

Les écrans lenticulaires, ou systèmes d'affichage plénoptiques, sont basés sur le même principe que l'acquisition. C'est à dire qu'un ensemble de micro-lentilles est couplé à un écran. Des dispositifs ont été implémentés, par NHK ou par Canon par exemple, mais ils restent des prototypes expérimentaux ou de démonstrations.

Les *Head Mounted Displays* (visiocasques en français) sont également des systèmes cibles pour les applications de réalité virtuelle et de vidéo 360.

D'autres types de systèmes de rendu ou d'affichage, plus marginaux, ont déjà été mis en démonstration, comme par exemple les systèmes dit *Tabletop* ou *All-around*, permettant à l'utilisateur de visualiser le contenu en relief en tournant autour du dispositif. Ces systèmes restent pour la plupart expérimentaux.

2.4 Codage

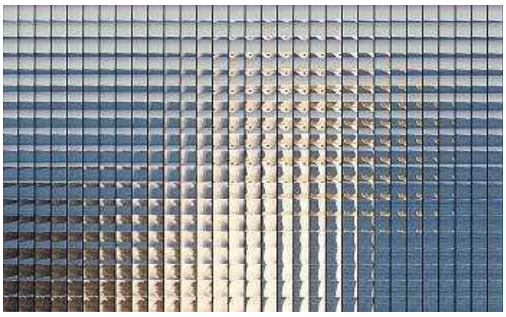
Il est possible d'utiliser les technologies de codage actuelles pour compresser des contenus light-field. Les images plénoptiques sont par exemple représentées sous la forme d'une large image composées de micro-images, et peuvent donc être encodées avec des codeurs 2D comme JPEG ou HEVC Intra. D'autres méthodes plus avancées ont été également proposées dans la littérature (voir Section 3).

Les différentes vues et cartes de profondeur d'un contenu au format MVD peuvent être encodées avec HEVC (en simulcast, c'est à dire que chaque vue est encodée indépendamment comme une séquence 2D), avec MV-HEVC, ou avec 3D-HEVC. Pour réduire encore le débit, il est également possible de n'encoder qu'un sous-ensemble de toutes les vues, et de synthétiser les vues manquantes après le décodage.

Bien que techniquement, les encodeurs actuels puissent être utilisés pour compresser des contenus light-field (parfois au prix d'une légère modification de la syntaxe, comme pour le nombre de vues en SMV), on s'attend à ce que la performance soit insuffisante, ou du moins sous-optimale pour plusieurs cas, comme par exemple un nombre de vues très grand, des caméras suivant un alignement non-liénaire (en arc typiquement), certains scénarios d'application type Free Navigation, ou tout simplement la structure en micro-images pour l'imagerie plénoptique. Des améliorations sont donc possibles et nécessaires pour le codage des light-fields.

2.5 Différentes représentations d'un même light-field

Les formats discutés dans cette section sont similaires car ils permettent un échantillonnage et une représentation du light-field. Il est possible en théorie de faire la conversion d'une représentation à l'autre, par exemple Point Clouds vers cartes de profondeurs, ou image intégrale vers vues, etc. Dans la pratique des limitations existent, car si une correspondance est clairement suggérée, les différentes façons d'échantillonner impliquent des compromis. L'imagerie plénoptique et le SMV capturent tous les deux une scène selon plusieurs angles, mais dans le premier cas on a une résolution limitée avec un capteur pour toutes les vues, alors qu'un ensemble de caméras offre la résolution totale d'une caméra pour chaque vue. Les ensembles de caméras permettent également un angle de captation plus large, tandis qu'une caméra plénoptique offre un échantillonnage plus dense avec un angle plus réduit. Les améliorations proposées pour les technologies de codage doivent prendre en compte ces aspects en plus de la performance de compression.



(a) Micro-Images

(b) Vue de la scène originale

Figure 3: Une image intégrale - *Seagull*

3 Schéma de codage d'image d'intégrale basé sur l'extraction de vues

3.1 Motivations

Les images plénoptiques, ou images intégrales ont une résolution large pour pouvoir fournir un nombre élevé de vues ayant une résolution suffisante. De plus, la structure en Micro-Images (MIs) induit un artefact en forme de grille qui complique l'encodage (Figure 3). Dans le Chapitre 3, on propose un schéma de codage original et efficace qui permet d'améliorer la performance de compression en prenant en compte ces caractéristiques.

3.2 État de l'art

3.2.1 Méthodes d'extraction

Le schéma de codage proposé dans nos travaux est basé sur l'extraction de vues. On décrit ici la méthode de l'état de l'art ayant servi à son implémentation. L'extraction d'une vue se fait par extraction d'un patch (i.e. un groupe de pixel) dans chaque micro-image. Ce patch est copié dans la vue. La position du patch dans la MI détermine l'angle de la vue extraite, et la taille du patch détermine la profondeur du plan de netteté dans cette vue.

Le moyennage (pondéré) des pixels qui entourent les patchs permet de réduire les effets de blocs et le crénelage. Dans ce cas, les pixels qui sont hors du plan de netteté sont floutés comme sur une photographie 2D classique. L'estimation de disparité au niveau micro-image permet de choisir une taille de patch variable en fonction de la profondeur des objets dans chaque MI.

3.2.2 Méthodes de compression

Les méthodes de compression d'image intégrale proposées dans la littérature sont basées sur des approches différentes et variées. Une première sous-catégorie se distingue, basée sur la transformée. En général, une Transformée en Cosinus Discrète (DCT), une 3D-DCT, ou une transformée en ondelettes, est appliquée à un ensemble de micro-images. Les

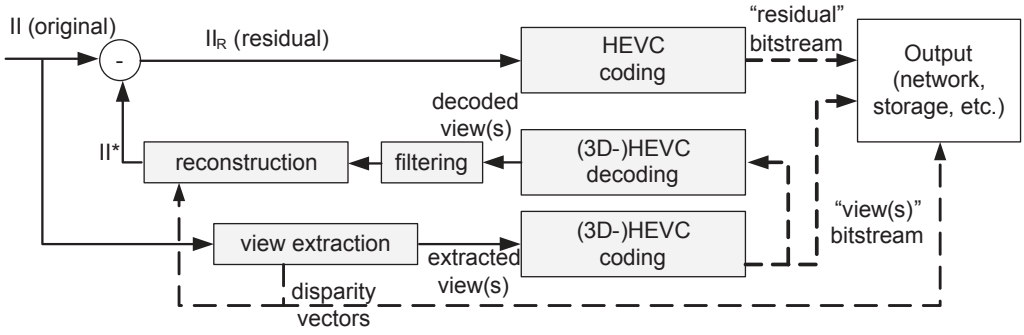


Figure 4: Schéma proposé - *encodeur*

résultats rapportés pour ces méthodes sont cependant en général limités en comparaison aux performances d'encodeurs tels que HEVC.

Une autre approche, dite *Pseudo Video Sequence* (PVS) ou *Multi-Vues*, se base sur l'encodage de séquences de micro-images ou bien de vues extraites, en les considérant comme des séquences 2D ou multi-vues classiques. L'aspect non naturel de ces images reste difficile à appréhender pour les encodeurs classiques.

L'approche *Self-Similarities* (SS) cherche à exploiter les redondances spatiales non-locales. Basée sur le principe du mode Intra Block Copy d'une extension d'HEVC (Range extension), elle consiste à effectuer une prédiction du bloc courant à partir d'un bloc de référence, situé non pas dans une autre image comme en prédiction temporelle, mais dans la partie de l'image courante déjà codée (zone causale). Cette méthode se révèle très efficace sur les images fixes mais atteint des limites de performance pour les séquences quand la prédiction temporelle est activée.

On peut finalement citer un schéma de codage scalable, dont le but est de créer un flux qui peut être décodé et lu par différents types de systèmes de display. La première couche est une vue centrale extraite de l'image intégrale (pouvant donc être affichée sur un écran 2D classique). La seconde couche est un ensemble multi-vue. La troisième couche correspond à l'image intégrale. Cette scalabilité à un coût en termes de débit, qui peut être réduit grâce à des méthodes de prédictions inter-couches.

3.3 Schéma de codage proposé

On propose un schéma de codage d'image intégrale illustré en Figure 4. Il est basé sur l'encodage d'une image résiduelle $\mathrm{II_R}$. Cette image correspond à la différence entre l'image originale II et une image prédite et reconstruite II*. Pour obtenir II*, on procède d'abord à l'extraction de vues, qui sont également codées, puis à partir de ces vues décodées on effectue l'opération inverse pour reconstruire une image proche de l'originale. Au décodeur, II* est obtenue avec la même opération, puis sommée à l'image décodée $\mathrm{II_R}$, pour obtenir la version décodée de l'image originale. L'avantage des vues est leur résolution réduite et leur aspect naturel facile à encoder en comparaison à II. Et l'information perdue lors des processus d'extraction et de reconstruction est transmise dans l'image $\mathrm{II_R}$. De plus, des résultats expérimentaux ont montré qu'appliquer un filtre moyenneur aux vues avant la reconstruction permet d'atténuer les erreurs dans II*. La performance de ce schéma dépend principalement de la qualité de l'image reconstruite II* et donc du compromis entre le débit attribué aux vues et celui attribué à l'image résiduelle $\mathrm{II_R}$.

3.4 Recherche exhaustive de la meilleure configuration

Trois paramètres principaux influent sur la performance: $\mathrm{QP_V}$ et $\mathrm{QP_R}$, qui correspondent aux paramètres de quantification utilisés respectivement pour les vues et pour $\mathrm{II_R}$, et M qui correspond à la taille (en pixels) du filtre appliqué aux vues avant l'étape de reconstruction.

On teste dans un premier temps de manière exhaustive un grand nombre de valeurs pour chacun de ces paramètres. Parmi les 1804 combinaisons de valeurs testées, on détermine celle qui donne la meilleure performance afin d'analyser ce résultat. On utilise pour cette expérience 7 images intégrales fixes, et on compare la performance du schéma proposé (utilisant une vue extraite), à une performance d'ancrage donnée par l'encodage de l'image originale II avec HEVC. La métrique BD-rate est utilisée.

Un gain moyen de 15.7% (jusqu'à 31.3% pour l'une des images) est obtenu. On observe qu'en moyenne, environ 97% du débit total est dédié à l'image résiduelle $\mathrm{II_R}$. La valeur de ce paramètre doit donc servir à définir le débit ou la qualité cible. On observe également que les valeurs de $\mathrm{QP_V}$ augmentent proportionnellement à celles de $\mathrm{QP_R}$. En revanche, M ne varie pas de manière significative en fonction de $\mathrm{QP_R}$, mais dépend du contenu de l'image testée.

Dans la suite, on cherche à déterminer un critère permettant de choisir automatiquement les valeurs optimales de $\mathrm{QP_V}$ et M pour un $\mathrm{QP_R}$ donné.

3.5 Détermination d'un critère d'optimisation du rapport débit-distortion

Notre but ici est, pour un $\mathrm{QP_R}$ donné, de trouver itérativement la configuration qui minimise un coût $D + \lambda R$, où D est la distortion, R le débit, et λ un multiplicateur de Lagrange à déterminer. À partir des résultats expérimentaux obtenus lors de la recherche exhaustive (cf. Section précédente), on peut obtenir la valeur de λ en fonction de $\mathrm{QP_R}$: $\lambda = f(\mathrm{QP_R})$. Par régression linéaire, en utilisant la méthode des moindres carrés, on obtient la fonction décrite par l'Équation 1, avec $a = 0.34$ et $b = -15.8$.

$$\lambda = 2^{a\mathrm{QP_R}+\mathrm{b}} \tag{1}$$

3.6 Méthodes itératives avec une vue extraite

On propose ici plusieurs méthodes itératives pour choisir automatiquement les valeurs optimales de $\mathrm{QP_V}$ et M pour un $\mathrm{QP_R}$ donné. Trois critères sont testés: le critère *RDO*, où l'on cherche à minimiser le coût $D + \lambda R$, avec λ tel que défini précédemment; le critère *MSE*, où l'on cherche à minimiser l'erreur quadratique moyenne entre II^* et II; et le critère *Fixe*, où l'on détermine les valeurs de manière empirique avant l'expérience. Le Tableau 1 résume l'utilisation de ces critères par les méthodes proposées. Les résultats expérimentaux sont donnés dans le Tableau 2.

Les gains proches des gains optimaux obtenus par recherche exhaustive montrent la robustesse du processus d'optimisation débit-distorsion proposé. On observe que le critère *MSE* est un bon indicateur de la performance également. Les résultats donnés par le critère *Fixe* montrent qu'on peut assigner empiriquement une valeur de $\mathrm{QP_V}$ à un $\mathrm{QP_R}$ donné. En revanche, des itérations sur M sont nécessaires. On obtient finalement avec la *method_3.1* un codec efficace avec une performance réaliste en termes de complexité, puisque le nombre réduit d'itérations permet d'avoir un temps d'exécution seulement de 1.3 (136%) fois le temps de l'ancrage.

Nom	Critère QP_V	Critère M	Itérations sur M
method_1	RDO		Toutes
method_2.1	RDO		Première itération sur QP_V
method_2.2	RDO	MSE	Première itération sur QP_V
method_3.1	fixed	MSE	Unique itération sur QP_V
method_3.2	fixe		Aucune

Table 1: Méthodes itératives proposées pour choisir QP_V et M

Méthode	BD-Rate (%)	Temps d'encodage (%)
method_1	-15.7	48367
method_2.1	-15.7	5526
method_2.2	-15.3	4443
method_3.1	-15.5	136
method_3.2	-8.5	120

Table 2: Gains BD-Rate moyens et variations du temps d'encodage

3.7 Amélioration de l'étape de filtrage

On propose d'améliorer l'étape de filtrage des vues, juste avant la reconstruction de II^*, en remplaçant le moyenneur par un filtrage de Wiener. Le filtrage de Wiener produit une estimation d'un signal cible, en filtrant un signal connu, de manière à minimiser l'erreur quadratique moyenne entre le signal cible et son estimation. Dans notre cas, l'image originale II est la cible, dont II^* est l'estimation, obtenu par reconstruction à partir des vues filtrées. Le fait d'insérer l'étape de reconstruction après le filtrage rend notre cas particulier, puisque les données filtrées et estimées sont présentées différemment. Pour contourner le problème, on peut effectuer le calcul des coefficients du filtre de Wiener au niveau des MIs. Les coefficients, calculés à l'encodeur, sont transmis au décodeur.

On propose deux méthodes. La première est basique, c'est à dire qu'on calcule un ensemble de coefficients (i.e. un filtre) pour l'ensemble des MIs. Pour la seconde, on adapte le filtre à la disparité de chaque MI.

Le filtrage de Wiener permet de faire passer les gains moyens du schéma proposé de 15.6% à 17.4% (avec la méthode adaptive). La méthode adaptive apporte une légère amélioration de seulement 0.2% sur la méthode basique. On observe également des légères pertes pour certaines des images testées, plus particulièrement pour les cas où le schéma est déjà efficace même avec le filtre moyenneur. Le filtre de Wiener minimise l'erreur quadratique moyenne, dont on a montré qu'elle est un bon indicateur de la performance du schéma proposé, cependant cette performance dépend également d'autres éléments (tels que l'aspect, lisse ou non, de l'image résiduelle par exemple).

3.8 Méthodes proposées avec plusieurs vues extraites

On propose de tester le schéma avec plusieurs vues extraites. Différentes combinaisons, avec un nombre de vues allant de 3 jusqu'à 9, sont testées. Les résultats montrent de larges améliorations par rapport au cas avec une seule vue extraite. Le meilleur résultat est donné par l'extraction de trois vues alignées horizontalement avec un gain moyen de 22.2%. On observe des gains très larges pour les images sur lesquelles le schéma avec

une seule vue extraite était le moins efficace. On observe également de légères pertes sur d'autres images, pour les cas où l'amélioration amenée par les vues supplémentaires ne permet pas de compenser l'augmentation du débit pour ces vues.

3.9 Combinaison et comparaison avec l'état de l'art

Finalement, on compare les performances du schéma de codage proposé aux performances de la méthode *Self-Similarity* (décrite précédemment). Cette méthode de l'état de l'art apporte également un large gain sur HEVC, de 19.1% en moyenne dans nos conditions de test. Le schéma proposé offre un gain moyen de 11.7% sur cette méthode.

Cependant, les deux méthodes sont compatibles. En effet, l'image résiduelle dans notre schéma peut bénéficier de l'activation de la méthode *Self-Similarity* pour son encodage. Lorsqu'elles sont combinées, ces deux méthodes offre un gain très large de 28% en moyenne sur HEVC.

3.10 Conclusions et perspectives

Dans ce chapitre, un schéma de codage robuste et efficace est proposé pour les images intégrales. Des gains moyens de 28% (et allant jusqu'à 36.3%) sont rapportés sur HEVC, avec des résultats réguliers sur l'ensemble des images testées. L'augmentation des temps de codage et de décodage est légère, et donc réaliste. De plus, de par sa structure basée sur HEVC, ce schéma est compatible avec la compression de séquences, et offre même une scalabilité au niveau display (avec un sous-ensemble de vues pouvant être décodé et affiché séparément).

Parmi les perspectives pour des travaux futurs, on peut citer entre autres l'adaptation du codage HEVC à l'aspect spécifique de l'image intégrale résiduelle, ou encore l'utilisation de cartes de disparité dense pour l'extraction de vue.

4 Évaluation subjective de contenu Super Multi-Vues compressé sur des écrans light-field

4.1 Motivations et contexte

Les travaux décrits ici sont le résultat d'une mission (*Short Term Scientific Mission*, STSM) financée par l'organisme COST Action 3D-ConTourNet, et réalisée en collaboration avec l'entreprise Holografika à Budapest (Hongrie). L'objectif principal est de déterminer l'impact de la compression sur la qualité perçue dans le cas des contenus et displays light-field Super Multi-Vues. On s'intéresse aux débits requis pour transmettre ce types de contenu, aux configurations recommandées pour un codage efficace, à la proportion de vues qu'il est possible de synthétiser, à l'impact de la synthèse sur la qualité, et finalement à la fiabilité de l'utilisation de la métrique PSNR pour le SMV. À notre connaissance, cette étude est la première à avoir montré de telles expériences et rapporté des résultats de ce type.

4.2 Conditions expérimentales

L'évaluation subjective est réalisée sur l'écran Holovizio C80 d'Holografika (Figure 5), dit Holovizio Cinema (dont les dimensions sont de 3×1.8 mètres). Deux séquences naturelles

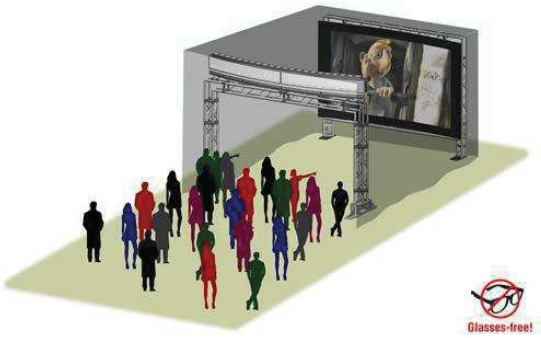


Figure 5: Holovizio C80 cinema system

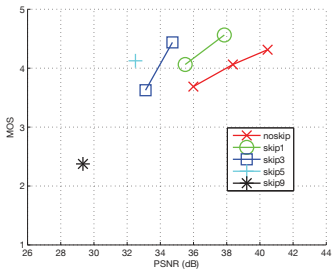
(*Dog* et *Champagne*), et une séquence synthétique (*Bunny*) sont évaluées. Chacune contient 80 vues, capturées ou générées avec un système de caméras alignées horizontalement en linéaire. Toutes les cartes de profondeur utilisées sont générées avec l'outil DERS (*Depth Estimation Reference Software*). Les séquences sont encodées à différents niveaux de qualité (i.e. différents débits) avec MV-HEVC. On compare également plusieurs configurations, notamment en faisant varier le nombre de vues synthétisées, avec un ratio allant de zéro à neuf vues synthétisées entre deux vues codées. 16 sujets prennent part à l'expérience, et évaluent les séquences avec la méthode DSIS (Double Stimulus Impairment Scale). Il s'agit de comparer la séquence compressée à l'originale, en notant la dégradation (i.e. due aux artefacts de compression) sur une échelle allant de 1 (pour très dérangeant) à 5 (pour imperceptible).

4.3 Résultats expérimentaux et conclusions

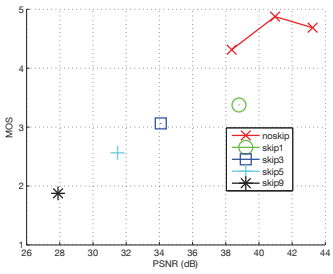
Les débits associés à une bonne qualité perçue sont de 5 Mb/s pour la séquence Bunny (avec 5 vues synthétisées entre deux codées), 11 Mb/s pour la séquence Dog (avec 1 vues synthétisée entre deux codées), et 10 Mb/s pour la séquence Champagne (sans synthèse). Ces débits sont réalistes par rapport aux futures attentes pour la 4K (de 10 à 15 Mb/s) ou la 8K (environ 3 fois plus) par exemple. Il faut cependant prendre en compte le fait que l'on travaille ici avec un ensemble de séquences restreint, avec des résolutions peu élevées (1280×960) et un contenu qu'on peut considérer comme simple à coder (avec un fond fixe par exemple). Le fait que ces débits soient réalistes n'implique donc pas que des améliorations des codecs ne sont pas nécessaires pour ce type de contenu à l'avenir.

Plusieurs configurations et structures de prédiciton sont comparées. On observe notamment l'intérêt des groupes de vues (*Groups Of Views*, GOV), qui consistent à introduire des images intra régulièrement pour obtenir des groupes de vues indépendants les uns des autres en termes de prédiction. Ces groupes offrent un compromis entre l'efficacité de codage et les limitations mémoires. Ils ouvrent également la voie au (dé-)codage parallèle.

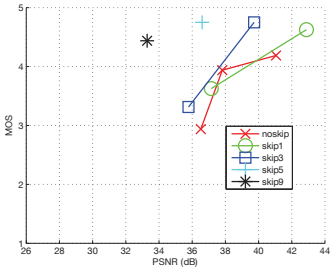
Concernant le ratio de vues codées et synthétisées, les résultats varient de manière très significative d'une séquence à l'autre. Pour Bunny on obtient une bonne qualité subjective même en synthétisant jusqu'à 9 vues entre 2 vues codées (le maximum testé dans notre expérimentation), alors que pour Champagne, la synthèse ne permet pas d'obtenir une qualité suffisante même en ne synthétisant qu'une seule vue entre 2 vues codées. La qualité de la synthèse dépend beaucoup du contenu et également de la qualité des cartes de profondeur. Des améliorations sont nécessaires pour les algorithmes de synthèse, afin



(a) Dog



(b) Champagne



(c) Bunny

Figure 6: MOS vs. PSNR

de rendre cette technique fiable pour le codage à l'avenir.

Pour illustrer l'impact de la synthèse sur la qualité des vues, on synthétise des vues à partir des vues originales, c'est à dire sans que la compression n'impacte le résultat. Les résultats objectifs obtenus vont de 25 dB à 44 dB (en PSNR). Ces résultats confirment la dégradation sévère de la qualité due à la synthèse pour certaines séquences.

Les résultats objectifs et subjectifs mis en correspondance donnent des courbes ascendantes (Figure 6). C'est à dire que le PSNR est capable de refléter une croissance ou une décroissance de la qualité subjective. En revanche, d'une configuration à l'autre, en fonction du nombre de vues synthétisées, les ordres de grandeurs varient de manière très significative. Par exemple pour la séquence Dog, une bonne qualité subjective peut être associée à des PSNR allant de 33 dB (pour 5 vues synthétisées) à 39 dB (sans synthèse). Le PSNR est bien plus sensible aux artefacts de synthèse que le système visuel humain. Cette métrique peut donc être utilisée pour évaluer objectivement des contenus SMV, mais seulement avec des configurations stables par rapport au ratio entre les nombres de vues codées et synthétisées.

Parmi les nombreuses perspectives pour le futur de ces travaux, il convient de mentionner principalement: l'utilisation d'un nombre plus large de séquences pour confirmer les résultats, la comparaison avec des contenus capturés avec d'autres arrangements de caméra (typiquement en arc), la comparaison d'intervalles de débits plus larges et plus denses, et finalement l'étude d'aspects plus spécifiques au light-field, comme la perception de la parallaxe de mouvement.

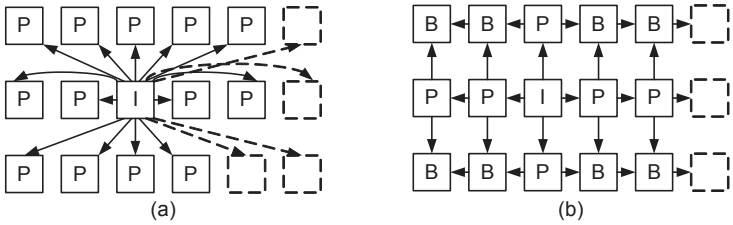


Figure 7: (a) Ancrage basique, (b) *Central2D*

5 Compression de contenu vidéo Super Multi-Vue avec parallaxe horizontale et verticale

5.1 Motivations

La parallaxe de mouvement est définie comme le changement optique de champs visuel qui résulte d'un changement de position de visualisation. Elle est considérée comme un facteur clé dans la perception du relief et de la profondeur. Certains dispositifs de capture tels que les caméras plénoptiques et les ensembles de caméras en deux dimensions permettent d'obtenir du contenu avec parallaxe horizontale et verticale. Les technologies de codage actuelles sont cependant limitées à la parallaxe horizontale uniquement, et nécessitent une adaptation et une amélioration pour être utilisées efficacement sur les contenus dit *full parallax*.

Dans la littérature, des améliorations sont proposées d'abord au niveau de la structure de codage, avec des schémas de prédiction inter-vues en deux dimensions. Cependant, l'une des limites des travaux existants est l'utilisation restreinte de la dimension verticale pour la prédiction. Des travaux proposent également des outils de codage qui s'appliquent directement au niveau des blocs (*Coding Units*, CU), utilisant les relations géométriques entre les vues, par exemple pour dériver des vecteurs de disparité (DV) afin de réduire la complexité (i.e. le temps d'éxécution dans ce cas) du codage. Dans le Chapitre 5, on propose des améliorations à ces deux niveaux, structures et outils, dans le but d'augmenter l'efficacité du codage.

5.2 Structure de codage proposée: *Central2D*

Le but dans cette section est d'améliorer l'efficacité du codage avec des modifications non-normatives, c'est à dire en utilisant le codec (et donc le standard) tel quel, seulement configuré différamment. On propose une structure de prédiction inter-vues en deux dimensions, *Central2D*, qui exploite efficacement l'alignement horizontal et vertical des vues, comme le montre la Figure 7 (b). La vue centrale est codée indépendamment (sans référence inter-vues). Pour une configuration avec $N \times M$ vues, les $N-1$ (respectivement $M-1$) qui sont sur le même axe horizontal (respectivement vertical) que la vue centrale sont codées avec une seule vue de référence (la plus proche dans la direction du centre). Toutes les autres vues sont codées en utilisant une référence horizontale et une référence verticale, ce qui permet d'exploiter les deux dimensions pour un grand nombre de vues (seulement $M+N-1$ vues n'utilisent pas les deux dimensions). De plus, avec cette méthode, la distance entre la vue codée et la vue de référence est minimale (i.e. vues adjacentes) et on n'utilise pas de référence en diagonale.

5.3 Améliorations proposées pour les outils de codage

Neighboring Block Disparity Vector (NBDV) et Inter-View Motion Prediction (IVMP) sont des outils de prédiction inter-vues implémentés dans 3D-HEVC. Le principe de NBDV est le suivant. Une recherche est effectuée dans les blocs voisins du bloc courant (c'est à dire le bloc à coder), qui sont déjà codés/décodés, dans le but de trouver un bloc qui a été prédit avec un vecteur de disparité (DV). Lorsqu'un tel bloc est trouvé, la recherche s'arrête et le DV est inséré dans la liste des candidats du mode *merge*, pour potentiellement servir à coder le bloc courant.

Ce DV trouvé par NBDV est ensuite utilisé par l'outil IVMP. Si le bloc pointé par le DV (se trouvant la vue de référence) est prédit par un vecteur de mouvement (MV), alors ce vecteur est également inséré dans la liste du mode *merge*.

NBDV et IVMP sont implémentés pour fonctionner dans les *Common Test Conditions* (CTC) de 3D-HEVC, c'est à dire avec une seule référence inter-vue, alignée horizontalement, et étant la vue centrale (avec l'index 0, la *base view*). On propose d'abord une adaptation du codeur en autorisant plusieurs références inter-vues, pouvant être horizontales ou verticales, et donc pouvant avoir un index différent de 0. Dans un second temps, on propose d'améliorer ses outils pour le cas *full parallax* en modifiant NBDV de manière à trouver non pas un, mais deux vecteurs de disparité: un horizontal et un vertical. Ce second DV permet de modifier le candidat *merge* inséré par NBDV pour qu'il utilise les deux vecteurs pour faire une bi-prédiction. Il permet également à IVMP d'insérer un second candidat, basé sur le second DV. Finalement, dans le but d'augmenter les chances de trouver ce second DV avec NBDV, on propose une méthode de dérivation inter-vue inspirée de l'outil IVMP. Si le bloc pointé par le premier DV horizontal trouvé est prédit par un DV vertical, alors ce deuxième DV est utilisé.

5.4 Résultats expérimentaux et conclusions

Deux séquences sont utilisées pour tester les méthodes proposées: *CoastalGuard* (50 frames, *Computer Generated*, résolution 768×384) et *Akko&Kayo* (290 frames, naturelle, résolution 640×480). Les expériences sont effectuées avec des configurations de 3×3 vues et 11×5 vues. La structure de prédiction proposée *Central2D* est comparée à un ancrage basique pour lequel seule la vue centrale sert de référence inter-vues pour toutes les autres vues. Elle est également comparée aux autres structures de l'état de l'art mentionnées dans la section dédiée du manuscrit. L'encodage de ces séquences est réalisé avec MV-HEVC. Pour les propositions d'améliorations des outils de codage NBDV et IVMP, l'encodage est réalisé avec 3D-HEVC (c'est à dire qu'on compare notre version améliorée de l'encodeur à l'encodeur 3D-HEVC sans modification). La métrique Bjøntegaard Delta (BD) rate est utilisée pour mesurer les gains en compression.

La structure de prédiction inter-vues proposée *Central2D* fournit des gains allant jusqu'à 8% dans la configuration avec 3×3 vues et jusqu'à 29% dans la configuration avec 11×5 vues, par rapport à la structure d'ancrage basique. Elle fournit ainsi la performance la plus élévée en comparaison à toutes les autres structures de l'état de l'art testées dans cette expérience.

De plus, les adaptations et améliorations proposées pour les outils de codage de 3D-HEVC fournissent des gains allant jusqu'à 3% dans la configuration avec 3×3 vues et jusqu'à 4% dans la configuration avec 11×5 vues, par rapport au résultat d'ancrage utilisant 3D-HEVC sans modification.

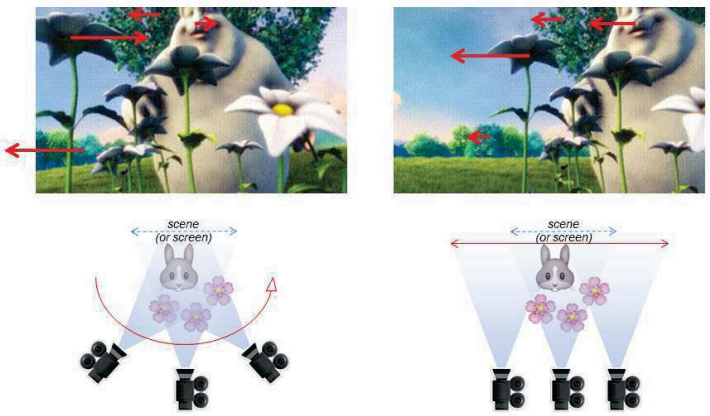


Figure 8: Comparaison de la disparité en arc (gauche) et en linéaire (droite)

Ces résultats montrent que la prise en compte de la bidimensionnalité du contenu à l'encodage pour la prédiction inter-vues permet d'améliorer de manière très significative l'efficacité de la compression. Ces travaux offrent donc des perspectives intéressantes d'amélioration, notamment en prenant en compte cet aspect pour améliorer d'autres outils parmi les nombreux outils de prédiction spécifiques à 3D-HEVC.

6 Impact de l'alignement en arc sur les outils de prédiction de la disparité

Le Chapitre 6 est dédié à l'étude de l'impact de l'arrangement des caméras sur la performance de compression des contenus SMV. Plus particulièrement, on compare l'efficacité des extensions d'HEVC sur les contenus linéaires et sur ceux en arc.

Dans la littérature, il est rapporté que les contenus SMV en arc sont plus adéquats que les contenus linéaires pour les écrans du type Holovizio d'Holografika, car ils permettent de couvrir un angle de vue plus large. En revanche, au niveau du codage, des résultats rapportent que les contenus en arc sont plus compliqués à compresser que les contenus linéaires.

La principale différence en termes de codage est la disparité (Figure 8). Si dans les séquences linéaires, la disparité est unidirectionnelle, dans les séquences en arc, les vecteurs de disparité peuvent pointer dans les deux directions horizontales (i.e. vers la gauche ou vers la droite). On étudie dans un premier temps l'impact de cette différence en effectuant des encodages de séquences générées avec les deux types d'arrangements. Contrairement à ce qui est reporté dans la littérature, la performance de codage est similaire dans nos résultats, et la variation des gains de 3D-HEVC sur HEVC ou MV-HEVC ne varie pas de manière significative d'un arrangement à l'autre.

Dans un second temps, on propose d'améliorer le codage des contenus en arc en exploitant cette spécificité. On propose des modifications d'outils de 3D-HEVC, similaires

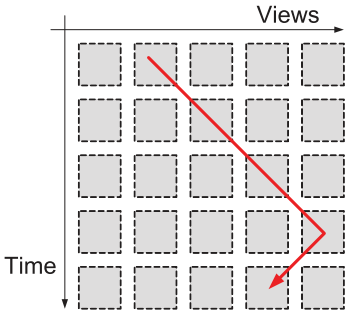


Figure 9: Simulation de navigation d'un utilisateur

à celles proposées pour les contenus *full parallax*, où le but est de dériver des vecteurs de disparité pointant dans les deux directions. Les méthodes proposées n'apportent pas de gain significatif car le nombre de cas où elles s'appliquent est limité, cependant l'analyse détaillée des résultats montre que des gains potentiels sont possibles en exploitant ces caractéristiques de la disparité dans la suite de ces travaux.

7 Schéma de compression pour les applications de *Free Navigation*

Les travaux décrits dans le Chapitre 7 du manuscrit se situent dans le contexte des applications *Free Navigation*. On se focalise sur un cas pratique réaliste, pour lequel toutes les vues d'une séquence SMV sont encodées et transmises au décodeur. L'utilisateur peut choisir une vue de manière interactive pendant la lecture de la séquence pour se déplacer de gauche à droite (ou de droite à gauche) dans la scène. L'encodage est réalisé hors-ligne et il n'y a pas de communication possible du décodeur vers l'encodeur. L'objectif pour le système de codage est de fournir un compromis intéressant entre la performance de compression (i.e. l'efficacité) et la flexibilité offerte à l'utilisateur en termes de changement de vue possible.

La Free Navigation amène de nouvelles contraintes pour le codage, car l'utilisateur peut changer de vue à chaque instant, et on peut donc se trouver dans un cas où les images précédentes dans la vue à décoder ne sont pas disponibles, car elles n'ont pas été décodées auparavant, lorsque l'utilisateur naviguait dans une autre vue. La première étape de l'étude consiste à comparer la performance des techniques de codage de l'état de l'art dans ces conditions. On compare des méthodes basées sur un encodage avec 3D-HEVC, où la configuration de prédiction varie, avec différentes structures de prédiction et différentes tailles de groupe de vues (GOV). Les deux principaux critères pour mesurer la performance des méthodes testées sont: l'efficacité de la compression (i.e. le rapport

	Débit	Temps de décodage et nombre d'images à décoder
All Intra (AI)	Max	Min
FTV (CfE)	Min	Max
GOV 5×8	Large	OK

Table 3: Résumé de nos conclusions sur la comparaison des méthodes d'ancrage

débit-distorsion) et le degré de liberté offert pour les changements de vues, qui dépend de la capacité du décodeur, et que l'on mesure donc ici en nombre d'images à décoder pour en afficher une.

Les conditions de test sont les suivantes. Trois séquences sont utilisées, comprenant 20 vues et 40 images. On compare deux configurations limites: *All Intra* (AI), où toutes les vues sont encodées indépendamment; et *FTV (CfE)*, où on exploite un maximum de corrélations temporelles et inter-vues avec 3D-HEVC. Finalement, on teste également des configurations intermédiaires avec des groupes de vues de tailles différentes pour trouver un compromis entre ces deux limites. La performance est mesurée en simulant un cas où l'utilisateur change de vue à chaque instant dans la même direction de manière à avoir une contrainte assez exigeante (Figure 9).

En termes de liberté de changement de vue, la configuration *All Intra* offre la possibilité de choisir n'importe quelle vue sans décodage additionnel, mais sa performance est limitée par un débit élevé. Pour la configuration *FTV (CfE)*, bien que l'efficacité soit bien plus élevée (avec un débit approximativement 20 fois moins large en moyenne), le nombre d'images à décoder est trop élevé pour être réaliste, car on doit en général décoder toutes les vues pour afficher une seule image. Les configurations intermédiaires offrent des compromis intéressants, par exemple avec les groupes de vues de taille 5×8 qui permettent de ne décoder que 10 images en moyenne pour en afficher une, avec un débit plus raisonnable (Tableau 3). Ce débit reste cependant élevé par rapport aux contraintes estimées pour ces expérimentations. Des améliorations de l'efficacité de codage semblent donc nécessaire pour ce type d'application.

Une méthode de codage basée sur des encodages redondants est proposée dans un chapitre annexe du manuscrit. Le principe est d'encoder l'image courante plusieurs fois de suite, en utilisant différentes vues comme référence à chaque fois, afin d'activer le changement de vue à chaque instant pour l'utilisateur. L'avantage de la méthode est d'utiliser les paramètres du codage fait avec la première référence pour les codages suivants, dans le but de gagner en efficacité.

8 Conclusion

Nos travaux fournissent des informations et des conclusions sur plusieurs aspects des technologies light-field pour les futures applications immersives. Plusieurs systèmes de capture et d'affichage (ou de rendu) émergent avec différentes caractéristiques, et avec beaucoup de similarités entre les formats existants pour le contenu qui en résulte, car en théorie ces formats correspondent tous à des façons différentes de stocker et représenter l'information échantillonnée à partir d'un light-field. En pratique, il y a cependant beaucoup de différences, et la conversion d'un format à un autre n'est pas triviale. Le dispositif de capture et le format du contenu ont un impact fort sur le reste de la chaine de traitement et sur la performance du système complet de l'acquisition au rendu. En conséquence, le choix du format de capture, de représentation, ou de stockage ne doit pas être uniquement fait en fonction de la performance de codage/compression mais doit aussi dépendre de l'application cible. Des méthodes de codage sont proposées et sont disponibles pour fournir un compromis entre l'efficacité de la compression et les fonctionnalités possibles, comme par exemple pour les applications de *Free Navigation* avec le degré de liberté donné à l'utilisateur. La scalabilité pour l'affichage est un autre exemple, notamment pour les images plénoptiques, pour lesquelles il est possible de décoder séparément une seule vue 2D de la scène.

Les conclusions tirées de nos résultats expérimentaux nous donnent des indications sur la faisabilité de l'implémentation de services et d'applications immersives basés sur les light-fields. Il est possible d'utiliser les technologies de codage actuelles, avec simplement quelques modifications au niveau des structures et des configurations, pour représenter du contenu light-field de différentes façons (e.g. Super Multi-Vue, imagerie plénoptique ou Nuages de points). De plus, les performances sont réalistes et nos expérimentations ne montrent aucune limitation qui empêcherait complètement ou définitivement l'utilisation de ces technologies light-field. Cependant, bien que les conclusions sur la faisabilité mettent en avant un futur prometteur pour les applications immersives basées sur le light-field, certains de nos résultats montrent le fait que des facteurs limitants doivent être étudiés et corrigés, typiquement dans le cas du Super Multi-Vue avec synthèse par exemple. Sur ces aspects, les modifications et les schémas de codage innovants proposés dans nos travaux fournissent des améliorations significatives de l'efficacité de compression. Ces résultats montrent que des améliorations sont possibles pour les différents types de contenu, qui permettront une meilleure représentation et un meilleur codage du light-field. Il faut s'attendre à ce que les futurs travaux et les futures contributions des experts du domaine conduisent à des formats communs, et potentiellement standards, qui vont dynamiser le développement des industries liées aux technologies light-field, et faire de la prochaine génération d'applications vidéo immersives une étape clé dans l'évolution de la consommation de contenus multimedias.

LIGHT-FIELD IMAGE AND VIDEO COMPRESSION FOR FUTURE IMMERSIVE APPLICATIONS

Antoine DRICOT

ABSTRACT : Evolutions in video technologies tend to offer increasingly immersive experiences. However, currently available 3D technologies are still very limited and only provide uncomfortable and unnatural viewing situations to the users. The next generation of immersive video technologies appears therefore as a major technical challenge, particularly with the promising light-field (LF) approach.
The light-field represents all the light rays (i.e. in all directions) in a scene. New devices for sampling/capturing the light-field of a scene are emerging fast such as camera arrays or plenoptic cameras based on lenticular arrays. Several kinds of display systems target immersive applications like Head Mounted Display and projection-based light-field display systems, and promising target applications already exist. For several years now this light-field representation has been drawing a lot of interest from many companies and institutions, for example in MPEG and JPEG groups.
Light-field contents have specific structures, and use massive amounts of data, that represent a challenge to set up future services. One of the main goals of this work is first to assess which technologies and formats are realistic or promising. The study is done through the scope of image/video compression, as compression efficiency is a key factor for enabling these services on the consumer markets. Secondly, improvements and new coding schemes are proposed to increase compression performance in order to enable efficient light-field content transmission on future networks.

KEY-WORDS : light-field, compression, image and video coding

TELECOM ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY