Paris Doctoral School
of Computer Science,
Telecommunications
and Electronics

Dissertation

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Graduate College of Telecom ParisTech

Speciality: **Image and Signal Processing**

Ismaël DARIBO

Coding and rendering of 3D video sequences; and applications to Three-Dimensional Television (3DTV) and Free Viewpoint Television (FTV).

Approved by:

Philippe Salembier	President
Hideo Saito	Readers
Luce Morin	
Christophe Tillier	Advisors
Béatrice Pesquet-Popescu	

Abstract

In the framework of 3D multiview imaging system, a large number of cameras capture the same scene from different viewpoints. There exists then a fairly strong correlation between the different video sequences, which leads to the necessity to properly take it into account, specially if one wants to transmit such 3D video sequence on a communication channel throughout the different views. Current 3D video standards use conventional predictive patterns, where the prediction is no longer calculated on the time axis but also one camera to another one.

This thesis aims to study the technical processing of a 3D video communication system, including both broadcast model (coding for video 2D + depth) and multiview video model. Direct applications are 3DTV and FVV applications. This area of research gathers various known techniques as video coding 2D+t theory (e.g., predictive coders, wavelet coders, etc.), computer vision background related to multiview imaging (e.g., stereoscopic analysis, DIBR, etc.), 3D display technologies, etc.

Remerciements

Mes remerciements vont en premier lieu Christophe Tillier et Béatrice Pesquet-Popescu pour m'avoir confié ce sujet de recherche, et pour avoir assuré la direction de mes travaux de recherche avec disponibilité et rigueur scientifique.

Je tiens de plus à exprimer mes remerciements aux membres du jury à savoir, Philippe Salembier qui m'a fait l'honneur de présider mon jury de thèse, Hideo Saito et Luce Morin qui ont bien voulu accepter la charge de rapporteur.

Je ne saurais trop remercier mes relecteurs indépendants qui m'ont apporté leur aide dans la relecture de ce manuscrit, en particulier Olivier Leblanc, Cyril Pichard, Vincent Nozick, Jérome Gauthier, Mounir Kaaniche; sans qui une délocalisation en off shore aurait été inéluctable. Mes remerciements vont de nouveau à Olivier Leblanc et Didier Maman pour leurs conseils avisés sur des sujets de pointe et d'actualité tels que la relaxation corporelle, la préparation de tarte et les grands astres célestes, qui en situation de stress m'ont été d'un grand secours.

Les marges de cette page sont bien trop étroites pour remercier tout ceux qui m'ont fait l'honneur d'assister à ma soutenance de thèse depuis les sièges de l'amphithéâtre ou au travers de l'hublot, pour qui maintenant la télévision 3D n'a plus de secret.

Je salue également les membres de l'équipe de recherche SISAR qui ont accompagné mes premiers pas dans le monde de la recherche, notamment dans de nombreuses réunions de travail, composées de deux groupes de deux personnes où les statistiques et la mémoire sont mises à l'honneur. Je tiens à remercier particulièrement Olivier Derpierre, Venceslas Biri, Cyril Pichard, Pascal Chaudeyrac, Patrice Bouvier, Vincent Nozick, François de Sorbier de Pougnadoresse, Pierre Boulenguez et Vincent Sermet.

Toujours dans la même lignée, un grand merci à l'équipe multimédia et audio du département TSI qui m'ont apporté leur bonne humeur durant ces 3 années de dur labeur. En particulier, je remercie mes colocataires de bureau[1] (lieu de concentration intellectuel multipolaire) : Maria Trocan, Thomas Maugey, Mounir Kaaniche, Sara Parrilli, Kundan Kandhway, Manel Abid, Claudio Greco. De plus, je ne saurais trop remercier les doctorants, post-docs, ingénieurs et stagiaires de l'équipe avec qui j'ai passé de très bon moments, que ce soit à la pause déjeuner de 12:00, pause café de 16:00 pétantes ou pot de l'EDITE, à savoir: Maria Trocan, Teodora Petrisor, Lionel Gueguen, Nicloas Tizon, Valentin Emiya qui m'ont chaleureusement intégré à mon début de thèse; Thomas Maugey, Laurent Oudre, Brahim Elloumi, Jérome Gauthier, Wided Souid Miled, Mounira Maazaoui, Jonathan Sillan, Romain Bouqueau, Mounir Kaaniche, Manel Abid, Rafael Galvao de Oliveira, Jean-Louis Durrieu, Romain Hennequin, et tous les autres, pour tous ces moments de fou rire. Et les derniers arrivés que je n'ai que croisé : Valentina Davidoiu, Giovanni Petrazzuoli, Irina Delia Nemoianu; à qui je leur souhaite bien du courage dans

[1] le bureau ne comportant que 4 places, le lecteur comprendra que nous n'avons pas occupé ce bureau en même temps.

cette longue aventure.

S’agissant des permanents du département TSI (et des bars parisiens pour certains), je remercie Fabrice Planche, Christophe Tillier, Marco Cagnazzo, Jean-Claude Dufour, Cyril Concolato, Jean Lefeuvre, Bertrand David et Roland Badeau.

Mais je n’oublie pas les assistantes de direction de l’Institut Charles Cros (paix a son âme) : Sylvie Donard, Marie-Neige Agot, Katia Dautry, Dominique Perronnet; du département TSI : Laurence Zelmar et Patricia Friedrich; et de l’EDITE: Florence Besnard; pour leurs disponibilités et gentillesse de tous les instants. Elles ce sont révélées être une source d’information de référence sur toutes les communications non-codifiées au sein du laboratoire. Également avec grand plaisir je remercie Fabrice Planche pour son soutient logistique et matériel, qui de près comme de loin a contribué à ce travail.

Un grand merci au professeur Hideo Saito pour m’avoir accueilli au sein de son équipe, et m’avoir permis, et à nouveau, de découvrir avec délectation les plaisirs de la vie au pays du soleil levant.

Certaines personnes ne peuvent être oubliées, à savoir ceux qui ont du me supporter durant toutes ces années; par ordre chronologique en commençant par le sud provençale en Avignon : Olivier Leblanc, Pierre-Louis Cayrel, Etienne Couturier, Cherif Maghlout, Mathurin Mellot, Philippe Pascal et les autres membres de la bande Créole Avignonnaise. Un peu plus au centre, pas très loin du monde féerique de Marne-la-Vallée : Virginie Jacquier, Louis-Philippe Lubino, Léonard Sabatier et tout le reste de la promo IMAC’05. Et enfin à Paris : Philippe Gauthier et tous les autres pour toutes ces soirées inoubliables, qui furent telles des feuilles de menthe apportant toutes leurs saveurs à un mojito. Qu’ils trouvent ici l’expression de toute ma gratitude pour leur soutient morale et leur amitié.

Une pensée à : Christian Harnais (mon mentor avec qui tout a commencé), Mme Brenoc (professeur de mathématiques), M. Morin (professeur de physique/chimie), et tous les enseignants qui m’ont transmis leur savoir (volontairement ou pas), et qui m’ont, au moins une fois, mis à la porte de leur cours (ou en n’ont eu l’envie).

Je terminerai par exprimer ma profonde gratitude et ma sympathie envers toute ma famille, ma mère, mon frère, ma soeur; pour leur soutient inconditionnel et incommensurable. Et en particulier ma tendre et bien-aimée grand-mère pour toute son affection depuis ma plus tendre enfance.

Contents

General Introduction

Introduction

The history of stereoscopy, stereoscopic imaging or three-dimensional (3D) imaging can be traced back to 1833 when Sir Charles Wheatstone created a mirror device that provides to the viewer the illusion of depth, in his description of the "Phenomena of Binocular Vision" [132, 133]. The process consists in merging two slightly different views of the same painting or drawing into one stereoscopic image, resulting in a compelling 3D perception of the original picture[2]. With the realization of still photography in 1839, it was only years before the paintings and drawings were replaced by photographs in his stereoscopic viewing device. In 1844, Sir David Brewster further developed the stereoscope by utilizing prismatic lenses to enlarge and fuse the stereo images (see Fig. 1).

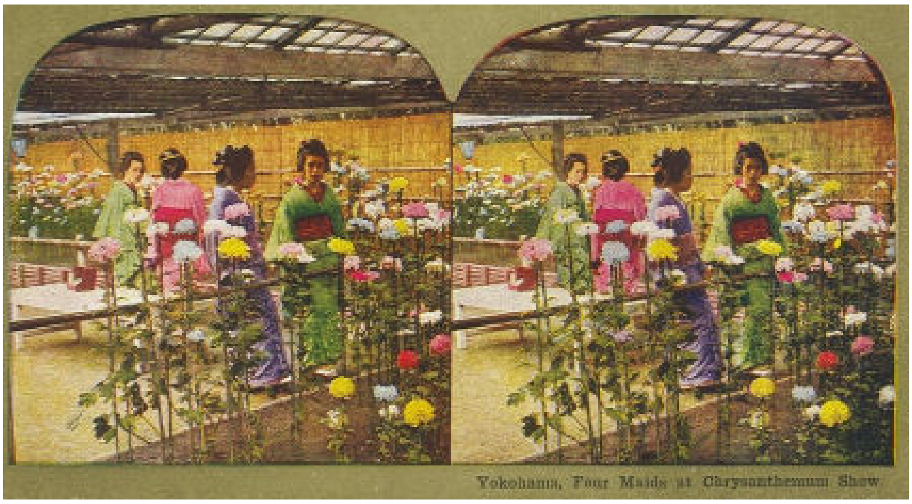


Figure 1: Yokohama, four maids at Chrysanthemum show. A stereo card intended to be viewed in a stereoscope.

Although stereoscopic films date back to 1903 when the Lumière brothers made the first 3D motion picture available to the public, it was not until the 1950s that Hollywood turned to 3D, trying to counteract the dropping box office receipts that occurred as a consequence of the increasing popularity of a competing technology: the television.

Already in 1928, British engineer John Logie Baird, one of the television (TV) pioneers, envisioned the stereoscopic television by demonstrating the principle in front of an audience

[2] Of course, the history of the study of *binocular vision* can be traced back much further, at least to Aristotle (*ca.* 330 B.C.) who considered that both eyes were moved from a single source and also notes the occurrence of double images, known as *diplopia* [130]

of scientists and representatives of the press at the Baird Laboratories in Long Acre. The first “non experimental” stereoscopic broadcast occurred some 55 years later with the Super Bowl halftime show and Coca-Cola commercial on NBC, and the Rose Bowl Parade on Fox.

Three-dimensional television (3DTV) has a long history, and over the years a consensus has been reached that a successful introduction of 3DTV broadcast services can only reach success if the perceived image quality and the viewing comfort is at least comparable to conventional two-dimensional television (2DTV). In addition, 3DTV technology should be compatible with conventional 2DTV to ensure a gradual transition from one system to the other. This is referred to as *backward-compatibility*.

On the other hand, 3D cinema has the ability to generate a compelling sense of physical space, and allows images to emerge from the screen and enter further into the spectator’s space, more than possible with conventional 2D or “flat” cinema. This effect was often exaggerated by throwing or poking objects from the screen at the viewer. Although many good stereoscopic movies were produced in the 1950s, stereoscopic cinema got a bad reputation with the public because of the discomfort experienced when viewing misaligned and overdone stereoscopic movies. Today, stereoscopic cinema is commercially relatively successful, with 3D-IMAX theaters being perhaps the most well-known exponents.

The improvement of 3D technologies raised more interest in 3DTV [37, 86] and in free viewpoint television (FTV) [16, 135, 117]. While 3DTV offers depth perception of program entertainments without wearing special additional glasses, free viewpoint video (FVV) allows the user to freely change his viewpoint position and viewpoint direction around a 3D reconstructed scene. In the meantime, the development of digital televisions and 3D displays has largely improved recently, and thus, reinforce this a wide interest in multiview video (MVV) applications. Sharp, Sony and Sanyo, three Japanese companies, have formed in march 2003 the 3D Consortium [1] in order to help the development of 3D technologies. Japan seems to be again among the first countries in the world to put 3DTV in the market, and develop FVV applications as discussed above. Japan plans to make it a commercial reality by 2020.

In recent events such as CEATEC 2008 (Japan), 3DX festival 2009 (Singapore), CES 2009 (U.S.), IBC 2009 (Amsterdam) and IFA 2009 (Germany), a strong interest has been remarked from various consumer electronics companies to provide 3D video products to the market. There are also a wide range of standardization bodies and consortia considering 3D, including Blu-Ray Disc Association (BDA), ATSC, SMPTE and 3D@Home, to name a few. Currently, the Blu-ray Disc Association (BDA) is in the process of establishing an official standard for 3D HD video. Hollywood is also taking 3D very seriously with some 40 films on the slate for the next three years and many others from majors and independents sure to come, like top executors from Fox, DreamWorks Animation, Disney and others.

Although there is no doubt that high definition television (HDTV) has succeeded in largely increasing the realism of television, it still lacks one very important feature: the representation of natural depth sensation. At present, 3DTV and FTV can be considered to be the logical next step complementing of HDTV to incorporate 3D perception into the viewing experience.

3D video application scenarios

The denomination 3D broadly refers to any visual or auditive system that attempts to maintain or recreate the illusion of 3D sphere of human viewing or hearing, including model representation with a collection of points in 3D space connected by various geometric entities. In the case of video application scenarios, we will refer in this work to the pseudo-3D denomination that refers to the ability of the viewer to perceived an illusion of depth, or the illusion to navigate in a 3D space.

Let us first introduce 3D display technologies that convey the 3D experience to the viewer, and afterwards present some of 3D imaging applications.

3D displays

Any 3D display system (anaglyph glasses, shutter glasses, autostereoscopic, holographic, volumetric, *etc*) which provides different perspective views to the left and right eye, will create a compelling and efficient sensation of depth. Ideally, such 3D displays provide stereopsis (*i.e.* binocular perception of depth), kineopsis (*i.e.* depth perception from motion parallax), and accommodation (*i.e.* depth perception through focusing). 3D displays that provide all of these depth cues are called multiview *autostereoscopic* or *automultiscopic* displays. They allow uninhibited viewing (*i.e.* without glasses) of high-resolution stereoscopic images from arbitrary positions. Modern automultiscopic displays use either holographic, volumetric or parallax technology.

Omni-directional video (ODV)

Conventional video cameras have limited fields of view that make them restrictive in a variety of vision applications. One way to enhance the field of view has been proposed through the omni-directional camera with is hemispherical field of view allows to capture information in all directions. Omni-directional video (ODV) which is recorded using an omni-directional camera has then become widely used because of recent advances in digital video technologies and photographic equipment for immersive video systems, tele-observation and also support surveillance systems.

Three-dimensional television (3DTV)

After the black-and-white prototypes evolved into high-quality color television, 3DTV is believed to be the next major revolution in the history of television by providing to the viewer a feeling of immersion in the movie.

In the sequel of the Advanced Three-dimensional Television System Technologies (ATTEST) project [98], the 3DTV Network of Excellence [4] and the 3D4YOU project [2], 3DTV is entrusted to be designed as a open, flexible and modular system (see Fig. 2), which can be used in a broadcast environment. Essential requirements are the backward-compatibility with existing 2D broadcast system and flexibility to support a wide range of different 2D and 3D displays.

Free viewpoint television (FTV)

Free viewpoint television (FTV) is an innovative visual media that enables the viewer to move his viewpoint freely. This technology enables the viewer to select his preferred view

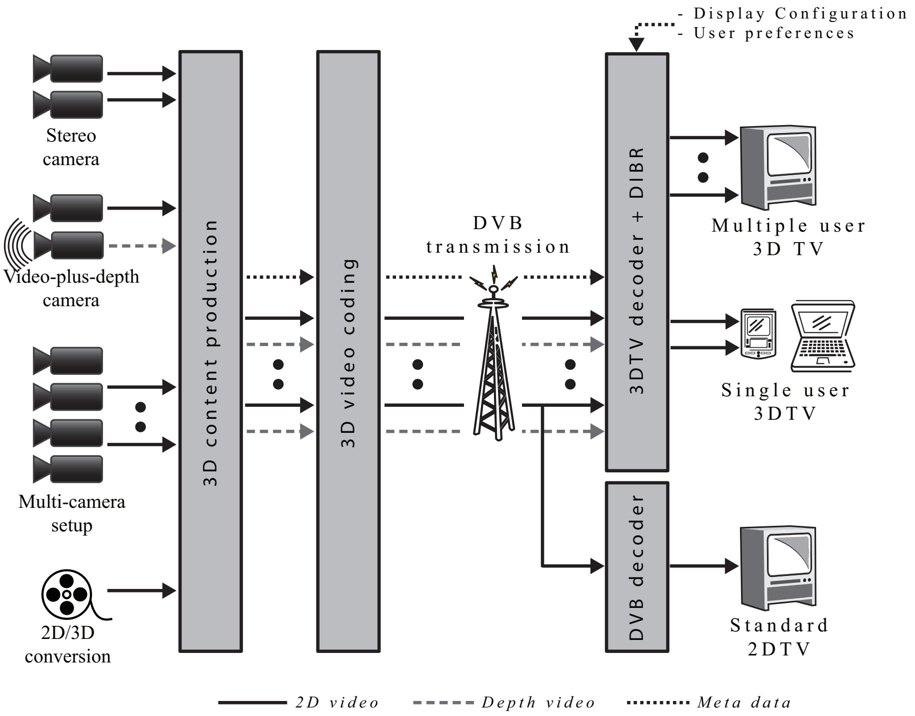


Figure 2: The ATTEST 3DTV end-to-end system.

point with a depth perception of the scene. This could bring an epochal change in the history of visual media since such a function has not been yet achieved by conventional media technology.

In Super Bowl XXXV, CBS employed a FVV system of cameras that allowed 3D visual effect such that the viewpoint revolves around the object event at a temporally frozen moment, also known as *bullet time* effect.

3D video on mobile phone

With the evolution of wireless communication technologies, it is expected that high quality computer graphics let mobile device users access 3D video technologies. In that way, the 3DPHONE project [3] aims to develop technologies and applications enabling a new level of user experience by developing an end-to-end 3D imaging mobile phone.

Scope of the thesis

In sight of the huge amount of data concerned by 3D video communication services, which increases with the number of camera channels, compression efficiency is of paramount importance. To this end, this thesis is devoted to efficiently encode 3D video data.

3D video coders usually fall in two categories: multiview-based and 3D-model-based approaches. Typically, 3D-model-based video coding exploits the *a priori* 3D model knowledge of the scene that is transmitted together with texture and animation parameters [96, 105, 10, 71]. On the other hand, multiview-based coding relies on scenarios where a 3D scene is captured by several cameras (MVV camera system), exploiting therefore the spatial correlations between adjacent cameras [106, 80, 22].

Since 2001, an ad hoc group on 3D audio and visual communication (3DAV) has been established by the Moving Picture Experts Group (MPEG), to respond the needs of a large number of companies for standards that enable MVV applications. Since then, MPEG moved to the joint video team (JVT) in April 2006 and focused on the improvement of compression efficiency for MVV systems. Recently, MPEG has initiated a new exploration experiment specifically targeted towards FVV applications in which more attention is given to depth estimation and 3D video rendering.

The objectives of this thesis consist of the analysis and design of new and efficient multiview-based video coding systems. More exactly, our research interests have been focused on:

- dealing with the disocclusion problem caused by the depth image based rendering (DIBR) technique,
- investigating the compression efficiency of the depth video by adaptive wavelet filter banks and its impact on the view synthesis,
- developing an MPEG-2-based coding scheme of a video-plus-depth sequence through a joint motion estimation and a joint bit allocation strategy,
- constructing and optimizing a dense depth/disparity estimation framework for an H.264-based coding of MVV sequences.

Thesis outline

The layout of this thesis is divided in two parts. The first part gives an overview on the state-of-the-art in 3D video communications by addressing the question of how 3D video may be represented and processed. This part provides the theoretical foundation for the next part. The second part covers the thesis work by presenting the contributions following the above research directions.

Part 1 – 3D video communications: a state-of-the-art

Chapter 1 This chapter outlines the theory relevant for understanding the imaging process of a 3D scene into a camera system. This chapter focuses on the projective geometry, a stereo camera by introducing the epipolar geometry with the *a priori* knowledge of the projective camera quantities, and finally reviews the existing 3D video data representations capable of supporting interactive 3D video services.

Chapter 2 This chapter outlines the general structure of a video codec and MVV extensions, to finally give a classification of existing international standardization activities relevant to 3D video communications.

Part 2 – Implementation of a 3D video codec

Chapter 3 Depth image based rendering (DIBR) technique has been recognized as a promising tool for supporting advanced 3D video services required in MVV systems. However, an inherent problem with DIBR is to fill the newly exposed areas (holes) caused by disocclusions. This chapter addresses the disocclusion problem. To deal with small disocclusions, a hole-filling strategy is designed by pre-processing the depth video with an adaptive Gaussian filter taking into account the distance to the edges. For larger disocclusions, an inpainting approach is proposed to retrieve the missing pixels, by using the depth information.

Chapter 4 In DIBR, the depth video is a key side information for view synthesis. This chapter investigates the compression of the depth video by wavelet filter banks, and his compression effect on the view synthesis. To limit the so-called Gibbs (ringing) artifacts, the proposed approach consists of an adaptive wavelet filter bank that is implemented by the lifting scheme, and in the meantime improving edge preservation of the depth video.

Chapter 5 By construction, a video-plus-depth sequence contains two similar data representations, the texture and the depth informations. However, when encoding a video-plus-depth sequence, the correlation between the texture and depth videos is generally not exploited. This chapter aims at reducing the amount of information to describe the overall video-plus-depth sequence by developing a compression scheme based on a joint motion estimation and joint bit allocation between the texture and depth videos.

Chapter 6 This chapter addresses the problem of MVV compression by taking advantage of the inherent correlation between adjacent cameras through disparity compensated prediction. This chapter proposes a dense motion/disparity estimation algorithm, designed to replace the classical block-based approach.

Appendix A – GPU-based photometric reconstruction from screen light

This appendix presents thesis work on Graphics Processing Unit (GPU)-based 3D reconstruction based on photometric techniques, which is slightly outside the principal focus of this dissertation. This chapter designs a photometric-based 3D reconstruction application based on cheap and accessible devices that allows users to communicate via online applications with a 3D perception of their interlocutor by only means of a computer screen and a web camera. A computer screen is used as a programmable light source capable of providing various lighting conditions, working with a web camera for the photometric reconstruction.

Part I

3D video communication: a state-of-the-art

Chapter 1

3D video data representation: a state-of-the-art

Contents

In this chapter, some background informations on projective geometry, which is necessary to better understand the procedures developed in this work, are discussed. Projective geometry plays an essential role in 3D video communication. Indeed, an estimate of the 3D structure of the scene potentially enables an efficient multiview-based video coding.

The chapter starts with an introduction to projective geometry, the basic concept of camera models and the epipolar geometry. We consider in this work that the camera parameters are given, and then, do not need to be estimated. Thus, we focus on the geometric relation between adjacent cameras, in order to provide the theoretical foundations to map one camera view onto another one.

In the second part, an overview of some existing 3D video data representations susceptible of supporting interactive 3D video services is provided. The problem of the image-based representation of a 3D video, the requirements on complexity and functionality of algorithms are discussed.

1.1 Introduction to projective geometry

1.1.1 Projective geometry

Projective geometry is an efficient mathematical framework used in computer vision which nicely models perspective projection. It is used in scene reconstruction, robotics for the positioning of a robot in space, object recognition, mosaicing, image synthesis, analysis of shadows, *etc*.

Intuitive definition

Euclidean geometry presents some limitations, as shown in the Fig. 1.1. We can see that the road boarders are not projected along parallel lines, but along lines appearing in the projected image to converge toward a single point, called *vanishing point*. This has the effect that distant objects appear smaller than nearer objects. Photographic lenses and the human eye work in the same way. Unlike the Euclidean geometry (finite), the projective geometry can models the concept of infinity. Thus, a 3D point located at the infinity will be able to be projected as a point onto the projected image. Projective geometry becomes therefore an attractive framework, against the mere Euclidean geometry.

Homogeneous coordinates

The relationship between Cartesian coordinates and Euclidean geometry is well known. Homogeneous coordinates and projective geometry bear exactly the same relationship. Homogeneous coordinates makes calculations possible in projective space just as Cartesian coordinates do in Euclidean space. As a result, affine transformations can be easily represented by a matrix thereby creating an augmented vector by adding another coordinate as the last element. For example, a 2D point usually written with two coordinates $(x, y)^\top$ in Euclidean space, becomes $(x_1, x_2, x_3)^\top$ in projective space, such that:

$$x = x_1/x_3, \quad y = x_2/x_3$$

where $x_3 \neq 0$. The case $x_3 = 0$ corresponds to points at infinity. Usually the coordinates $(x, y)^\top$ and $(x_1, x_2, x_3)^\top$ are called respectively *inhomogeneous coordinates* and *homogeneous coordinates*. In the same way, a 3D point represented by a 3-element vector $(x, y, z)^\top$ becomes $(x_1, x_2, x_3, x_4)^\top$.

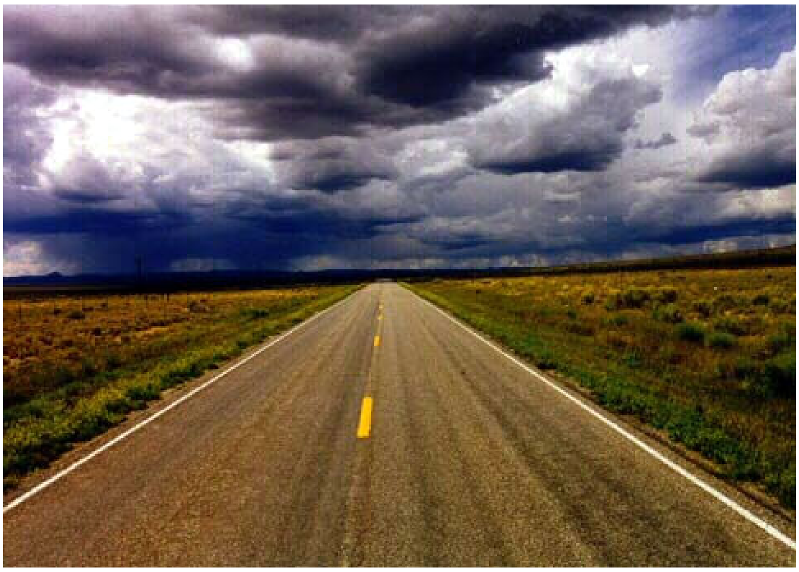


Figure 1.1: Two parallel lines get narrower and meet at the vanishing point.

Projective space

As a generalization, the projective space $\mathbb{P}^n$ is defined by the relation of equivalence:

$$\begin{aligned} &(x_1, ..., x_{n+1}) \approx (x'_1, ..., x'_{n+1}) \\ \iff &\exists\lambda \neq 0 \ \mid \ (x_1, ..., x_{n+1}) = \lambda(x'_1, ..., x'_{n+1}) \end{aligned}$$

where $\lambda \neq 0$ corresponds to a free scaling parameter, usually called *homogeneous scaling factor*. Homogeneous coordinates provide therefore a scale invariant.

Projective transformations

The relationship between a point in a homogeneous projective space $\mathbb{P}^n$ and its projection in an other homogeneous projective space $\mathbb{P}^m$ can be described with a matrix. It is a projective projection (also known as *linear mapping*):

$$\begin{aligned} \mathbb{P}^n &\longmapsto \mathbb{P}^m \\ \mathbf{x} &\longmapsto \mathbf{y} = \boldsymbol{W}\mathbf{x} \end{aligned}$$

where $\boldsymbol{W}$ is a non-singular $(m+1)\times(n+1)$ matrix. This homogeneous equation is scale invariant, so, the matrix $\boldsymbol{W}$ is a homogeneous matrix, and has at most $(m+1)\times(n+1)$-1 degrees of freedom.

1.1.2 Pinhole camera model

Here we describe the image acquisition process known as the *pinhole camera model* [48], which is regularly employed as a basis in this thesis (see Fig. 1.2). It broadly approximates all current models. It is described by its *optical center* $\mathbf{C}$ (also known as the *camera projection center*) and the *image plane*. The plane parallel to the image plane containing the optical center is called the *focal plane* of the camera. The distance of the image plane from **C** is the *focal length* f. The line passing through the optical center, and perpendicular to the image plane is called the *optical axis* or *principal ray* of the camera.

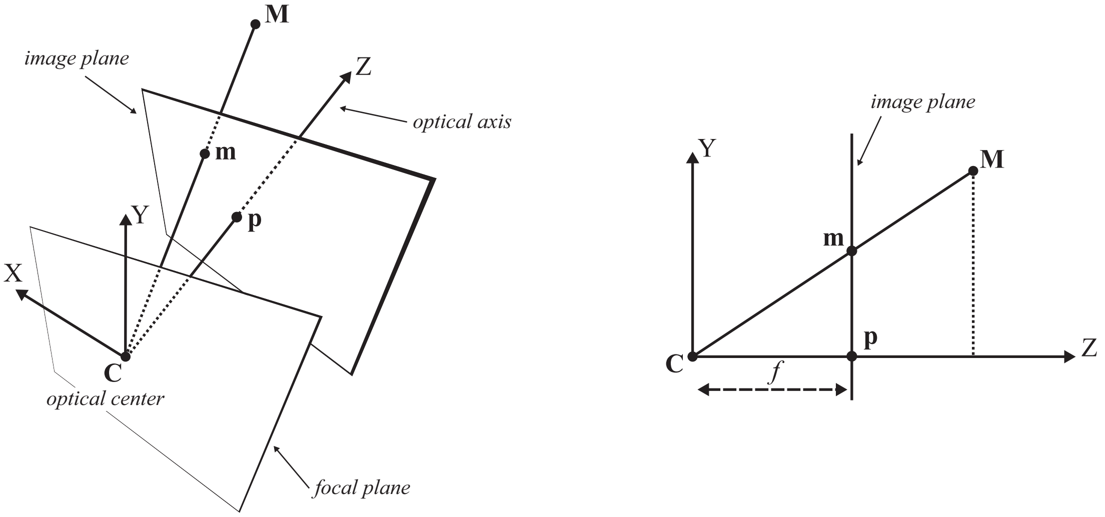


Figure 1.2: Pinhole camera geometry

The pinhole camera model denotes a simple transformation from the 3D-world coordinates system $\mathcal{R}_0$ onto a 2D-image plane in the camera coordinates system $\mathcal{R}_C$. This transformation can be decomposed in five steps:

- Changing of the 3D coordinate system from the 3D-world coordinate system $\mathcal{R}_0$ to the 3D-camera coordinate system $\mathcal{R}_C$.
- Projection 3D/2D : the 3D point expressed in $\mathcal{R}_C$ is projected onto the image plane.
- Shifting of the origin of the image.
- Transformation of the distance metric coordinate units to pixel units.
- Lens distortion correction.

The parameters in the camera model are organized in extrinsic (or external) parameters and intrinsic (or internal) parameters [48, 97]. The extrinsic parameters describe the position and orientation of the cameras with respect to the world coordinate system $\mathcal{R}_0$, as illustrated in Fig. 1.3. The intrinsic parameters describe the properties of the lenses and the charged-coupled device (CCD) chips within the camera, such as focal length, pixel size, center of projection.

The parameters of this model can be estimated through a process called camera calibration [48, 126], which is based on the analysis of image features (lines, points, corners, ...). Knowing the camera calibration enables to easily move from an Euclidean space into a projective space.

In a more complex model, errors resulting from many properties and artifacts of cameras (misaligned lenses, mispositioning of the CCD chip, *etc*) are taken into account.

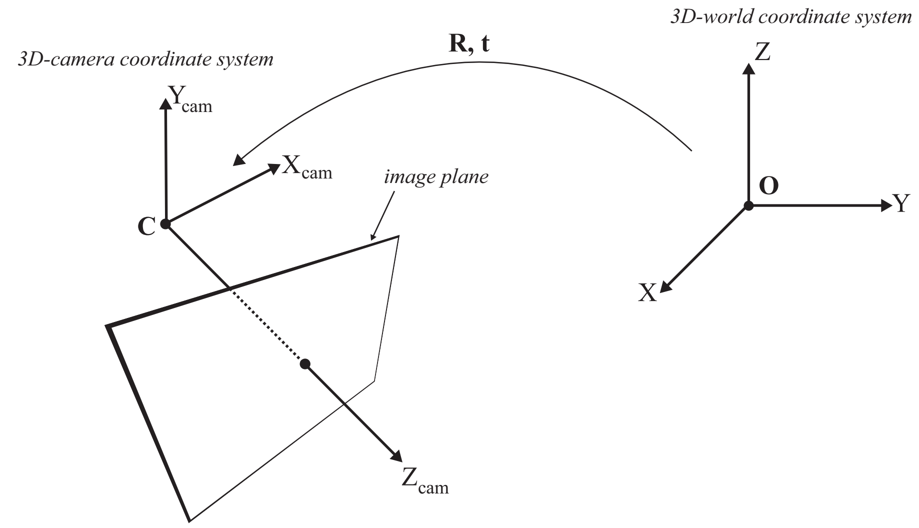


Figure 1.3: World to camera coordinate system

Extrinsic camera parameters

The extrinsic parameters describe the position and orientation of the camera with respect to the 3D-world coordinate system $\mathcal{R}_0$. It is equivalent to know the geometric transformation between a 3D point expressed in the 3D-world coordinate system $\mathcal{R}_0$ and in the 3D-camera coordinate system $\mathcal{R}_C$. This transformation encapsulates a 3×1 position vector $\mathbf{C}$ expressed in $\mathcal{R}_0$, and an orthogonal 3×3 rotation matrix $\boldsymbol{R}$ (see Fig. 1.3). The relation between a 3D-world homogeneous point $\mathbf{M} = (x, y, z, w)^\top$ and a 3D-camera homogeneous point $\mathbf{M}' = (x', y', z', w')^\top$ can be written as:

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} & \mathbf{0}_3^\top \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_3 & -\mathbf{C} \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \tag{1.1}$$

Notice that the 3×1 position vector $\mathbf{C}$ is expressed in non-homogeneous coordinates. The all zero element is denoted by the 1×3 vector $\mathbf{0}_3$, and the 3×3 identity matrix by $\boldsymbol{I}_3$. Alternatively, when combining matrices, Eq. (1.1) can be reformulated as:

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\mathbf{C} \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} \\ \mathbf{0}_3 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} - \begin{bmatrix} \boldsymbol{R}\mathbf{C} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} \\ \mathbf{0}_3 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} + \begin{bmatrix} \mathbf{t} \\ 1 \end{bmatrix} \tag{1.2}$$

where $\mathbf{t} = -\boldsymbol{R}\mathbf{C}$ is a 3×1 translation vector.

In the remainder of the thesis, we will regularly use the following notation:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = [\boldsymbol{R}|\mathbf{t}] \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \tag{1.3}$$

Intrinsic camera parameters

By opposition to the extrinsic parameters that describe the external position and orientation of the camera, the intrinsic parameters indicate the internal camera parameters.

3D/2D perspective projection using homogeneous coordinates A simple perspective projection of a 3D-camera point $\mathbf{M}' = (x', y', z', w')^\top$ to a 2D-image point $\mathbf{m} = (u, v, \eta)^\top$ on an image plane can be described using only information about the focal length f. Then the perspective projection can be simply expressed as:

$$\begin{pmatrix} u \\ v \\ \eta \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} \tag{1.4}$$

where $\eta = z'$ is the homogeneous scaling factor.

Principal point offset The pinhole camera model assumes that the origin of the image coordinate system, corresponding to the principal point offset $(o_x, o_y)^\top$, is located at the center of the image. In this case $(o_x, o_y)^\top = (0, 0)$. However, in general, most of the current imaging systems define the origin at the top-left pixel of the image. A conversion of coordinate system is thus necessary. Using the homogeneous coordinates, the principal point offset can be readily integrated into the projection matrix as follows:

$$\begin{pmatrix} u \\ v \\ \eta \end{pmatrix} = \begin{bmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} \tag{1.5}$$

Image-sensor characteristics: transformation from distance units to pixel units The derived camera projection matrix described in Eq. (1.5) ignores the fact that the CCD image sensor of the camera might not be squared and be skewed. Indeed, some imaging systems, especially those which must maintain compatibility with standard-definition television motion pictures, define an image as a grid of rectangular pixels in which the width of the pixel is slightly different from that of its height, *i.e.*, an aspect ratio different of 1:1. For example, in still camera photography three common aspect ratios are 4:3, 3:2 and, commonly seen in professional cameras 16:9 aspect ratio. Furthermore, due to the fact that pixels can be skewed, the image plane corresponds thus to a parallelogram as shown in Fig. 1.4. When the pixels are not manufactured to have a 90 degree angle, the skew is non zero, especially for the older cameras. In modern cameras, it is a reasonable approximation to suppose a zero skew.

Both aforementioned imperfections of the imaging system can be taken into account in the camera model, using the pixel size s_x, s_y in distance unit(mm) along the CCD axes, and the skew s of the pixels (see Fig. 1.4). The projection matrix can be updated as follows:

$$\begin{pmatrix} u \\ v \\ \eta \end{pmatrix} = \begin{bmatrix} f/s_x & s & o_x & 0 \\ 0 & f/s_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = [\boldsymbol{K}|\mathbf{0}_3] \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} \tag{1.6}$$

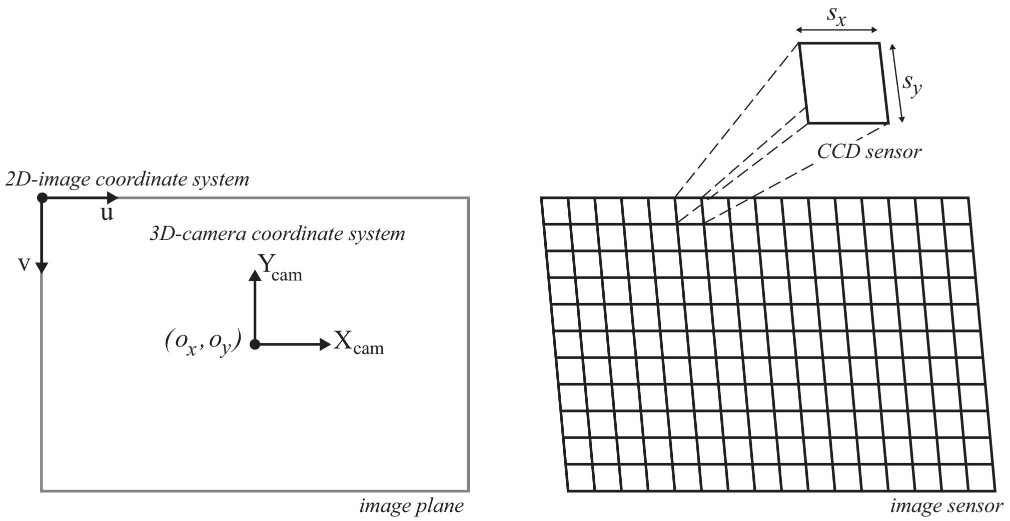


Figure 1.4: Non square skewed pixels

K is a 3×3 camera calibration matrix. It depends on the so-called intrinsic parameters discussed above. The all zero element is denoted by $\mathbf{0}_3$.

Radial lens correction So far, the imaging operation has been assumed to be perfectly linear. However, real camera lenses suffer from a non-linear lens distortion. The degree of lens distortion increases as the focal length decreases. In practice, radial lens distortion causes straight lines to appear curved. As seen in Fig. 1.5, the radial lens distortion appears more visible at the image edges, where the radial distance is high. A standard technique to model the radial lens can be described as follow.

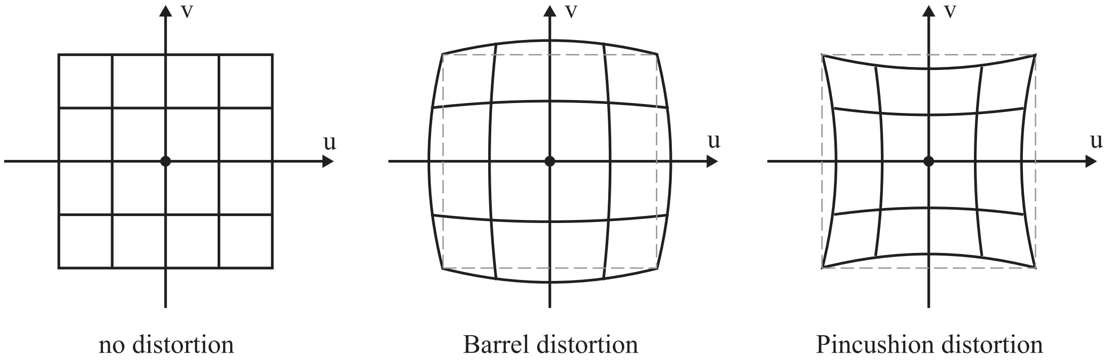


Figure 1.5: Radial lens distortion

Let $(u_c, v_c)^\top$ and $(u_d, v_d)^\top$ be the corrected and distorted pixel positions respectively in non-homogeneous coordinates. The relation between a corrected and distorted pixel can be modeled with a polynomial function and can be written as a function of the *distorted pixel position* as follows:

$$\begin{pmatrix} u_c - o_x \\ v_c - o_y \end{pmatrix} = L(r_d) \begin{pmatrix} u_d - o_x \\ v_d - o_y \end{pmatrix}, \tag{1.7}$$

where the distortion polynomial function L is given by:

$$L(r_d) = 1 + \sum_{i=1}^{n} k_i \cdot r_d^{2i} \quad \text{and} \quad r_d^2 = (u_d - o_x)^2 + (v_d - o_y)^2. \tag{1.8}$$

where $2n$ is the order of the distortion polynomial L. For most of the practical tasks, second order or fourth order polynomial is sufficient. In the following we will consider the second order polynomial (*i.e.* $n = 1$). Eq. (1.8) can be updated to:

$$L(r_d) = 1 + k_1 \cdot r_d^2 + k_2 \quad \text{and} \quad r_d^2 = (u_d - o_x)^2 + (v_d - o_y)^2. \tag{1.9}$$

In the case $k_1 = 0$, it can be noted that $u_c = u_d$ and $v_c = v_d$, which corresponds to the absence of radial lens distortion.

To generate an undistorted image, we have to apply the function $L(r)$ on the *distorted pixel position*. This technique is usually known as the *inverse mapping* method. The inverse mapping technique consists of scanning each pixel in the output image, re-sampling and interpolating the correct pixel from the input image. To perform an inverse mapping, the inversion of the radial lens distortion model is necessary and can be described as follows.

First, similarly to the second part of Eq. (1.9), we define:

$$r_c^2 = (u_c - o_x)^2 + (v_c - o_y)^2. \tag{1.10}$$

Then, taking the squared norm of Eq. (1.7), it can be derived that:

$$(u_c - o_x)^2 + (v_c - o_y)^2 = L^2(r_d) \cdot \big((u_d - o_x)^2 + (v_d - o_y)^2\big), \tag{1.11}$$

which is equivalent to

$$r_c = L(r_d) \cdot r_d. \tag{1.12}$$

When taking into account Eq. (1.9), this equation can be rewritten as a cubic polynomial:

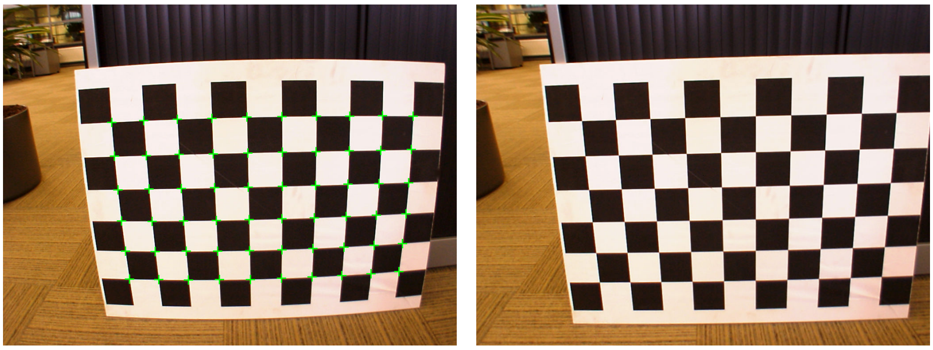
$$r_d^3 + \frac{1}{k_1} r_d - \frac{r_c}{k_1} = 0. \tag{1.13}$$

The inverted lens distortion function can be derived by substituting Eq. (1.12) into Eq. (1.7) and developing it:

$$\begin{pmatrix} u_d - o_x \\ v_d - o_y \end{pmatrix} = \frac{r_d}{r_c} \begin{pmatrix} u_c - o_x \\ v_c - o_y \end{pmatrix}, \tag{1.14}$$

where r_d can be calculated by solving the cubic polynomial function of Eq. (1.13).

Estimation of distortion parameters The discussed lens-distortion correction method requires knowledge of the lens parameters, *i.e.*, k_1 and $(o_x, o_y)^\top$. The estimation of the distortion parameters can be performed by minimizing a cost function that measures the curvature of lines in the distorted image. To measure this curvature, a practical solution is to detect feature points belonging to the same line on a calibration rig, *e.g.*, a checkerboard calibration pattern (see Fig. 1.1.2). Each point belonging to the same line in the distorted image forms a bended line instead of a straight line [33, 122]. By comparing the deviation of the bended line from the theoretical straight line model, the distortion parameters can be calculated. In the following, we will consider corrected imaging system.



(a) Distorted image

(b) Corrected image

Figure 1.6: Distorted image and its corresponding image using the inverted mapping method.

Conclusion

To sum up, a 3D-world point $\mathbf{M} = (x, y, z, w)^\top$ is projected onto a 2D-image point $\mathbf{m} = (u, v, \eta)^\top$

$$\begin{pmatrix} u \\ v \\ \eta \end{pmatrix} = \boldsymbol{P} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \tag{1.15}$$

where $\boldsymbol{P}$ is a 3×4 camera projection matrix and $\mathbf{m} = (u, v, \eta)^\top$ is a homogeneous point on the image plane. In general, the projection matrix $\boldsymbol{P}$ is a 3×4 full-rank matrix, invariant by scale factor. It can be decomposed as:

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\mathbf{t}] \tag{1.16}$$

$\boldsymbol{P}$ has 11 degrees of freedom: 5 from the calibration matrix $\boldsymbol{K}$, 3 from $\boldsymbol{R}$ and 3 from $\mathbf{t}$. $\boldsymbol{R}$ is a 3×3 rotation matrix, $\mathbf{t}$ is a 3×1 translation vector, and $\boldsymbol{K}$ is a 3×3 camera calibration matrix defined as:

$$\boldsymbol{K} = \begin{bmatrix} f/s_x & s & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.17}$$

The camera projection center $\mathbf{C}$ is the only 3D point for which the projection is not defined, *i.e.*:

$$\boldsymbol{P} \begin{pmatrix} \mathbf{C} \\ 1 \end{pmatrix} = 0 \tag{1.18}$$

thus $\mathbf{C}$ is the right null-space of $\boldsymbol{P}$. After solving, we obtain:

$$\mathbf{C} = -\boldsymbol{P}_{3\times3}^{-1} \cdot \boldsymbol{P}_{.,4} = -\boldsymbol{R}^{-1}\mathbf{t} \tag{1.19}$$

where $\boldsymbol{P}_{3\times3}$ is the matrix composed of the first three rows and first three columns of $\boldsymbol{P}$, and $\boldsymbol{P}_{.,4}$ the fourth column of $\boldsymbol{P}$. Notice that the 3×1 vector $\mathbf{C}$ is expressed in non-homogeneous coordinates.

1.1.3 Two-views geometry: epipolar geometry by knowing the projection camera matrices

In this section, we will give much attention to the case of two camera views capturing the same scene from different viewpoints. This approach is denoted in the literature, as *stereoscopic* camera system. The principle of acquiring a 3D information with a stereo camera system is shown in Fig. 1.7, where the two perspective views are acquired simultaneously. Most of the 3D scene points must be visible in both views. However, this may not always be valid in the case of occlusions, *i.e.*, as we can see in Fig. 1.7, the point **M** located between the two spheres, is visible by only one viewpoint.

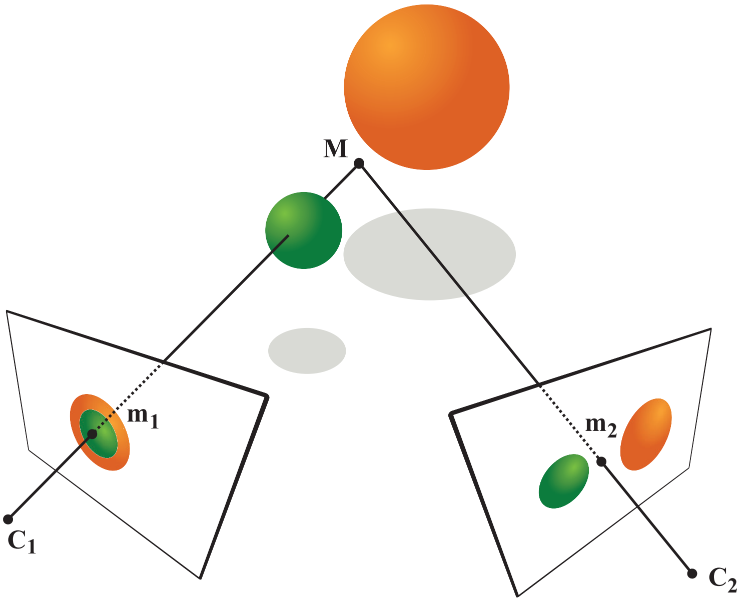


Figure 1.7: 3D Acquisition from a Stereo Camera

From this point, we will consider only finite points in the remainder of this work.

Any unoccluded 3D point $\mathbf{M} = (x, y, z, 1)^\top$ is projected, through the camera centers $\mathbf{C}_1$ and $\mathbf{C}_2$, onto the first and the second view as $\mathbf{m}_1 = (u_1, v_1, 1)^\top$ and $\mathbf{m}_2 = (u_2, v_2, 1)^\top$, respectively. Algebraically, each view has an associated 3×4 projection camera matrix $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. The 3D-world mapping onto each 2D-image can be expressed by the following equations:

$$\lambda_1 \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \boldsymbol{P}_1 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1.20a}$$

$$\lambda_2 \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \boldsymbol{P}_2 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1.20b}$$

Intuitively, we can notice an existing geometric relation between the two 2D-image point $\mathbf{m}_1$ and $\mathbf{m}_2$, since they correspond to the same 3D point $\mathbf{M}$. In stereo system, this relation can be formulated by the *epipolar geometry* [48]. Thus, the position of an object in one view can be deduced by knowing the position of this same object in the second view. Fig. 1.8 illustrates the epipolar relation between the two views.

Some terminology related to the epipolar geometry are introduced here (Fig. 1.8).

- The 3D points $\mathbf{M}$, $\mathbf{C}_1$, $\mathbf{C}_2$ define a plan Π called *epipolar plane*. By construction, the projected points $\mathbf{m}_1$ and $\mathbf{m}_2$ belong then to Π.
- By definition, the 3D point $\mathbf{M}$ lies in the line $(\mathbf{C}_1\mathbf{m}_1)$. The corresponding 2D-image point $\mathbf{m}_2$ belongs thus to the mapping of the line $(\mathbf{C}_1\mathbf{m}_1)$ in the second image plane I_2. That projected line is called *epipolar line* of I_2 associated to $\mathbf{m}_1$.
- The line going through the two camera projection centers ($\mathbf{C}_1$ and $\mathbf{C}_2$) is called the *baseline*.
- The *epipoles* e_1 and e_2 are the 2D-image points determined by the intersection of the image plane with the baseline. Moreover, all the epipolar lines of the image I_2 intersect at the same point, the epipole e_2, which is the projection of $\mathbf{C}_1$ onto I_2. In the same way, $\mathbf{C}_2$ is projected onto I_1 on the epipole e_1.

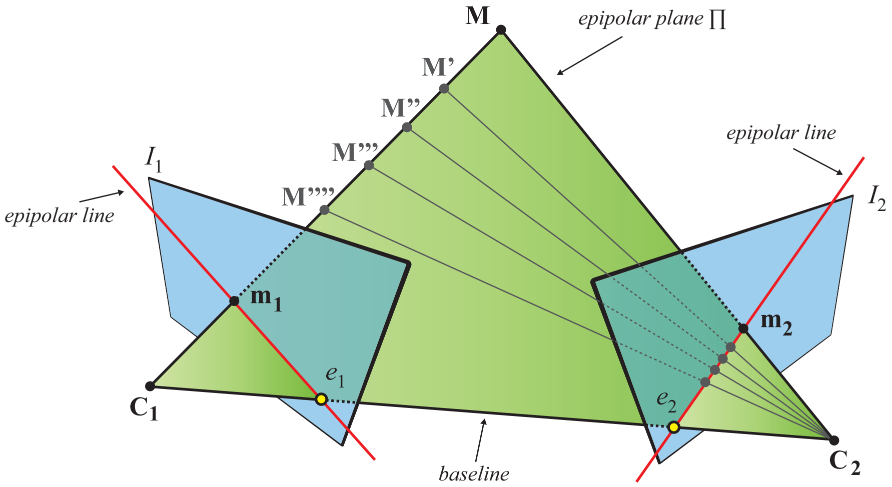


Figure 1.8: Epipolar geometry

The epipolar geometry can be described analytically in several ways, depending on the amount of *a priori* knowledge. Three general cases can be identified:

- If both intrinsic and extrinsic camera parameters are known, the epipolar geometry can be described in terms of the projection matrix (Eq. (1.20)).
- If only the intrinsic parameters are known, we work in normalized coordinates and the epipolar geometry is described by the essential matrix. The essential matrix can be considered as the calibrated form of the fundamental matrix.
- If neither intrinsic nor extrinsic parameters are known, the epipolar geometry is described by the fundamental matrix.

The essential and fundamental matrices are defined in the following.

The epipolar constraint

The perspective projection of a 3D-world point $\mathbf{M} = (x, y, z, 1)^\top$ onto an image plane I_1 in a 2D-image point $\mathbf{m}_1 = (u_1, v_1, 1)^\top$ can be geometrically modeled by an optical ray, passing through the optical camera center $\mathbf{C}_1 = (C_{1_x}, C_{1_y}, C_{1_z})^\top$ and the point in space $\mathbf{M}$. The projective 2D-image point $\mathbf{m}_1$ is therefore the intersection of the optical ray with the image plane I_1.

As discussed above, the corresponding 2D-point $\mathbf{m}_2 = (u_2, v_2, 1)\top$ must lie on a line known as *epipolar line* in I_2. This line is a projection of the optical ray onto I_2. Let us first express the equation of the optical ray, in order to derive the inverse projection process.

The inverse projection, known as *back-projection*, consists in back-projecting the 2D-image point $\mathbf{m}_1$ to the 3D space and deriving the corresponding coordinates. The optical ray can be modeled by the set of 3D-world points $\mathbf{M}$ resulting from the back-projection of $\mathbf{m}_1$ as a parametric line as follows:

$$\mathbf{M} = \begin{pmatrix} \mathbf{C}_1 \\ 1 \end{pmatrix} + \lambda_1 \begin{pmatrix} \boldsymbol{R}_1^{-1} \boldsymbol{K}_1^{-1} \mathbf{m}_1 \\ 0 \end{pmatrix} \tag{1.21}$$

where λ is a positive scaling factor defining the position of the 3D-world point $\mathbf{M}$ on the optical ray. If the depth value z of the 3D-world point $\mathbf{M}$ is known, it is possible to compute the two other coordinates x and y by calculating λ with the relation:

$$\lambda_1 = \frac{z - C_{1_z}}{c} \quad \text{where} \quad \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \boldsymbol{R}_1^{-1} \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{1.22}$$

The equation of the epipolar line can be derived from the equations (1.15) and (1.21). As we mentioned earlier, the epipolar line of I_2 can be considered as the projection of the optical ray:

$$\lambda_2 \mathbf{m}_2 = \boldsymbol{P}_2 \mathbf{M} = \boldsymbol{P}_2 \begin{pmatrix} \mathbf{C}_1 \\ 1 \end{pmatrix} + \lambda_1 \boldsymbol{P}_2 \begin{pmatrix} \boldsymbol{R}_1^{-1} \boldsymbol{K}_1^{-1} \mathbf{m}_1 \\ 0 \end{pmatrix} \tag{1.23}$$

The essential matrix

If the intrinsic camera parameters $\boldsymbol{K}$ are known, then we may switch to *normalized coordinates* as follows:

$$\text{if} \quad \hat{\mathbf{m}} = \boldsymbol{K}^{-1} \mathbf{m} \tag{1.24}$$

$$\text{then} \quad \hat{\mathbf{m}} = \boldsymbol{K}^{-1} \boldsymbol{P} \mathbf{M} = [\boldsymbol{R} | \mathbf{t}]\, \boldsymbol{M} \tag{1.25}$$

where $\hat{\mathbf{m}}$ is the expression of the projected point $\mathbf{m}$ in normalized coordinates. $\hat{\boldsymbol{P}} = \boldsymbol{K}^{-1} \boldsymbol{P} = [\boldsymbol{R} | \mathbf{t}]$ is called a *normalized projection camera matrix*, in which the known calibration $\boldsymbol{K}$ has been removed.

Now, consider a pair of normalized projection camera matrices $\hat{\boldsymbol{P}}_1$ and $\hat{\boldsymbol{P}}_2$ with an associated camera coordinate system $\mathcal{R}_{C_1}$ and $\mathcal{R}_{C_2}$, respectively, where the world coordinate system is fixed on the first camera (*i.e.* $\mathcal{R}_{C_1} = \mathcal{R}_0$). Consider also the matrices $\boldsymbol{R}$ and $\mathbf{t}$ characterizing the rotation and the translation from the first camera coordinate system $\mathcal{R}_{C_1}$ toward the second one $\mathcal{R}_{C_2}$. Now, the two normalized projection matrices can be expressed as:

$$\hat{\boldsymbol{P}}_1 = [\boldsymbol{I}_3 | \mathbf{0}_3] \quad \text{and} \quad \hat{\boldsymbol{P}}_2 = [\boldsymbol{R} | \mathbf{t}] \tag{1.26}$$

where the 3×3 identity matrix is denoted by $\boldsymbol{I}_3$ and the all zero element by $\mathbf{0}_3$.

The essential matrix corresponding to the pair of normalized camera matrices has the form:

$$\boldsymbol{E} = [\mathbf{t}]_\times \boldsymbol{R} = \boldsymbol{R}\left[\boldsymbol{R}^\top \mathbf{t}\right]_\times \tag{1.27}$$

where $[\mathbf{t}]_\times$ is the skew-symmetric matrix representation of the cross product with the 3×1 vector $\mathbf{t}$.

The relationship between two corresponding image points $\hat{\mathbf{m}}_1$ and $\hat{\mathbf{m}}_2$ in normalized coordinates system is expressed by:

$$\hat{\mathbf{m}}_1^\top \boldsymbol{E} \hat{\mathbf{m}}_2 = 0 \tag{1.28}$$

$\boldsymbol{E}$ is a 3×3 matrix which encodes only information on the extrinsic camera parameters between two camera views. Its rank is two and has only five degrees of freedom: a 3D rotation and a 3D translation direction.

The fundamental matrix

The fundamental matrix $\boldsymbol{F}$ may be thought of as the generalization of the essential matrix in which the camera parameters are not known. $\boldsymbol{F}$ can be computed from correspondences between image points alone. No knowledge of neither camera parameters nor relative pose is required. For any pair of corresponding image points, we have:

$$\mathbf{m}_1^\top \boldsymbol{F} \mathbf{m}_2 = 0 \tag{1.29}$$

The fundamental matrix $\boldsymbol{F}$ is a 3×3 matrix, rank two homogeneous matrix that defines the epipolar geometry between two images from two uncalibrated cameras [48].

1.2 3D video data representation

This section introduces different 3D video data representations linked to the 3D video applications (3DTV, FTV, 3D phone, ...) discussed in the general introduction. These different 3D video data representations have different advantages and drawbacks with regard to the complexity, efficiency and functionality according to the following general requirements:

- utilize existing delivery infrastructure and media as much as possible,
- require minimal change to device components,
- backwards compatibility - unacceptable for 3D services to break existing devices,
- support wide range of display devices and allow for future extension,
- high quality.

1.2.1 Conventional stereo video

A stereoscopic system is the most well-known and the most simple type of acquisition technique for 3D video data representation. A stereoscopic video can provide 3D impression by using a left video and a right video as a pair, thereby a stereo camera system, while a monoscopic 2D video cannot provide it. As a result, a pair of 2D videos is acquired, one for the left eye, and the other for the right eye, as illustrated in Fig 1.9.

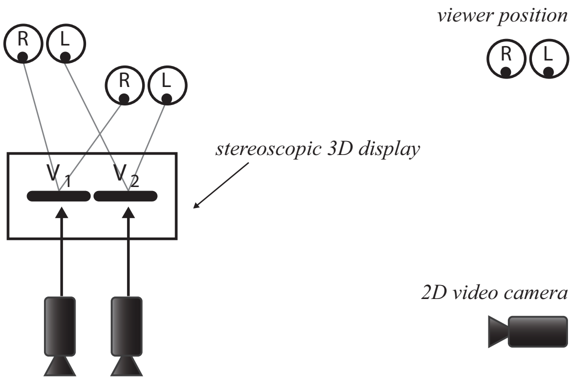


Figure 1.9: Efficient support of stereoscopic displays based on stereo video content.

Stereo multiplexing A simple way to represent the stereo video data is to apply one of the multiplexing approaches shown in Fig. 1.10, which include the time and spatial multiplexing.

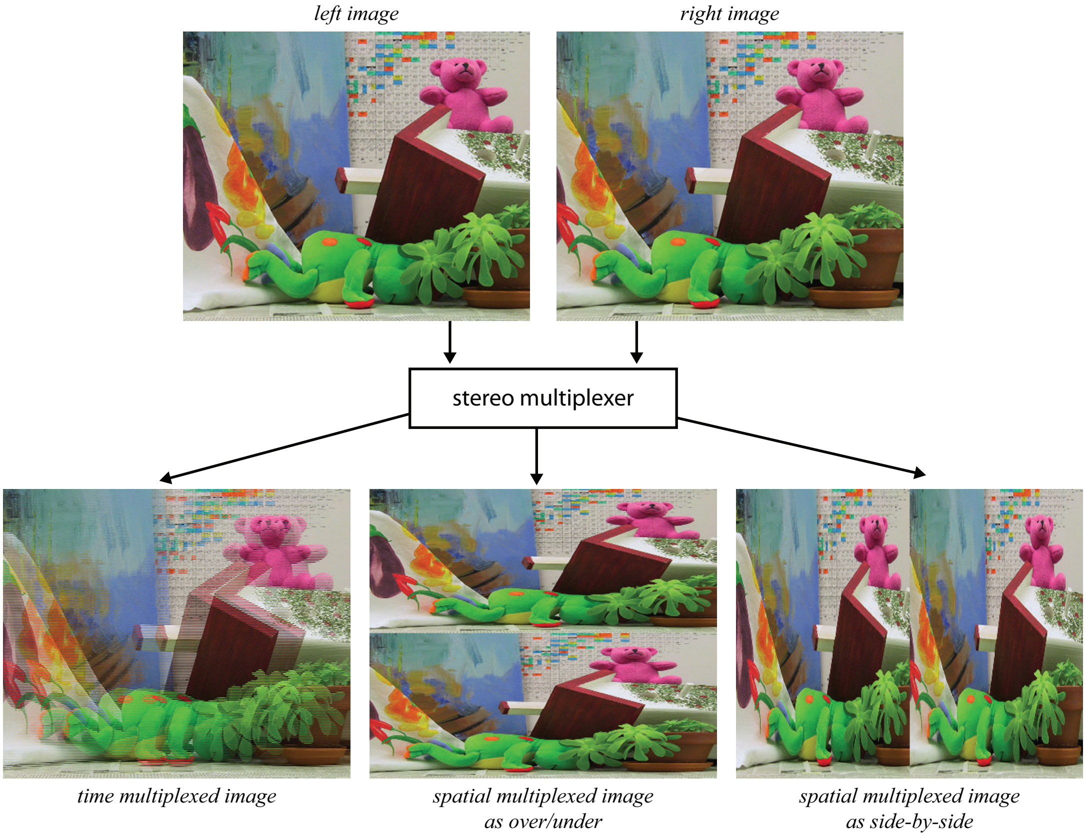


Figure 1.10: Stereoscopic multiplexing pictures.

Within the time multiplexed format, the left and right pictures are interleaved as alternating frames or fields. With spatial multiplexing, the left and right pictures would appear in either a side-by-side or over/under format. As is often the case with spatial multiplexing, the respective views are "squeezed" in the horizontal dimension or vertical dimension to fit within the size of the original picture, at the cost of loosing the spatial resolution.

Binocular suppression theory A new stereo representation has been derived from the so called binocular suppression theory [108, 109]. Subjective experiments have shown that to some extent, if one of the videos of a stereo pair is low-filtered, the perceived overall quality of the stereo video will be dominated by the higher quality image. As a result, other mixed stereo video representations may be derived, *e.g.*, in which one of the video may have a lower resolution, a lower quality, *etc*.

A general drawback of the stereo video representation is that it can be optimized for only one receiver configuration (size, number of views of the display, display type, *etc*), *i.e.*, 3D impression can not be modified at the receiver side. Moreover, the baseline is fixed from capturing. Also, head motion parallax, occlusion and disocclusion can not be supported when moving the viewpoint.

1.2.2 Video-plus-depth

Another well-known representation format is the video-plus-depth data representation. Initially studied in the computer vision field, the video-plus-depth format provides a regular 2D video enriched with its associated depth video (see Fig. 1.11).

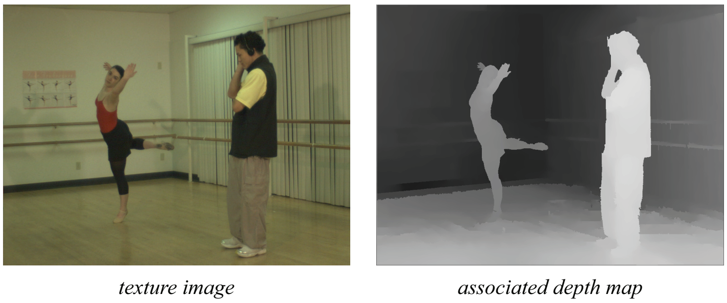


Figure 1.11: Texture picture and its associated depth map.

The 2D video provides the texture information, the color intensity, the structure of the scene, whereas the depth video represents the Z-distance per-pixel between the optical center of the camera and a 3D point in the visual scene. In the following the 2D video may be denoted as texture video in opposition to the depth video.

The depth video can be regarded as a monochromatic texture-less video signal. Generally, the depth data is quantized with 8 bits, *i.e.*, the closest point is associated with the value 255 and the most distant point is associated the value 0. With that, the depth video is specified as a smoothed gray level representation. The per-pixel depth data range is restricted to a range in between two extremes Z_{near} and Z_{far} indicating the minimum and maximum Z-distance.

The benefits of this representation is to still be able to respond the stereoscopic vision needs at the receiver side as illustrated in Fig. 1.12. After decoding, the second color video corresponding to the second view is reconstructed from the transmitted video-plus-depth data by means of DIBR techniques [77, 74, 94]. The ability to generate a stereoscopic video from a video-plus-depth data at the receiver side is an extended functionality, compared to

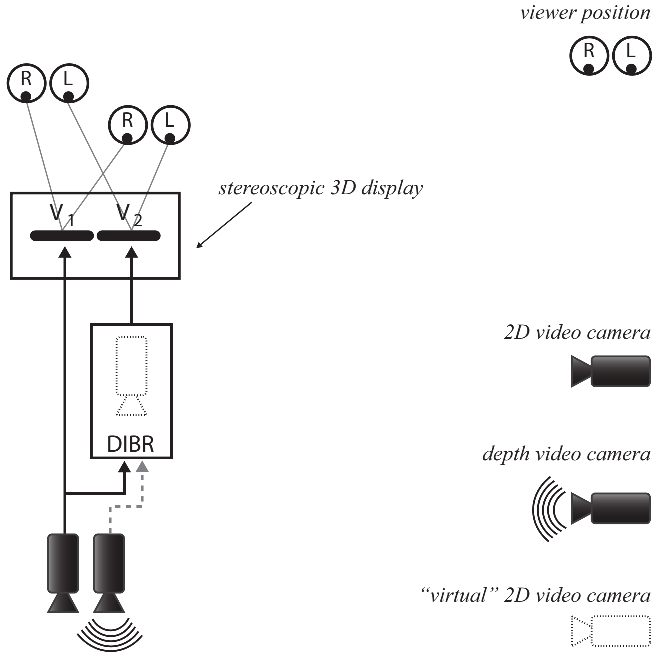


Figure 1.12: Efficient support of stereo autostereoscopic displays based on video-plus-depth content.

the conventional stereo video data representation. The 3D impression can thus be adjusted and customized after transmission.

However, the rendering process does not allow to create perfect "virtual" views in general, it is still prone to errors in particular for disoccluded points, but has other advantages. The adaptivity required at the user side for the different kinds of receivers is automatically obtained by the possibility to render all the desired "virtual" views. Also, the concept of video-plus-depth is highly interesting due to the backward compatibility, the compression efficiency of the depth video and the rendering extended functionality.

On the other hand, the advantages of video-plus-depth data representation over the conventional stereo video data representation are paid by increasing the complexity at the sender side and the receiver side. First of all, the depth video data has to be generated. This is usually done by depth/disparity estimation from a captured stereo video content [101]. Such algorithms are highly complex and still prone to errors. Also, a drawback at the decoder side is that the format is only capable of rendering a limited depth range since it does not directly handle occlusions.

So, video-plus-depth content is more suitable for applications with playback functionality, where the depth estimation and the view synthesis can be performed offline, *e.g.*, in a production studio or home 3D editing suite.

1.2.3 Multiview video

MVV can be considered as an extension of the stereo video data representation to a higher number of views.

As illustrated in Fig. 1.13, multiview autostereoscopic displays project multiple views at the same time into the viewing zone, such that consecutive views act like stereo pairs. As a result, head motion parallax viewing can be supported within practical limits, but

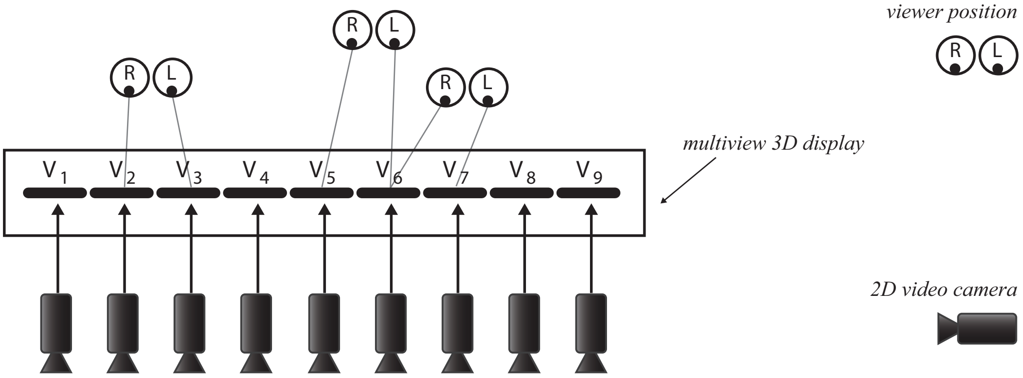


Figure 1.13: Efficient support of multiview autostereoscopic displays based on MVV content.

the amount of data to be processed and transmitted increases significantly compared to the conventional stereo data or classical 2D video.

1.2.4 Multiview video-plus-depth

The development of a wide range multiview autostereoscopic displays and FVV applications, increase more and more the number of output video needs. The user can therefore choose his own viewpoint (e.g. super bowl XXXV, bullet time effect, *etc*). Such advanced 3D video applications require a 3D video format that allows rendering a continuum of output views or a very large number of different output views at the decoder side.

The previously presented 3D multiview video format are not still enough sufficient to support such requirements without intensively increasing the number of input views, and so on, the bandwidth.

As we discussed before, video-plus-depth supports only a very limited continuum around the available original view, since view synthesis artifacts increase dramatically with the distance of the "virtual" viewpoint. To overcome this issue, MPEG started an activity to develop a new 3D video standard that would support these requirements [107]. It is based on a multiview video-plus-depth (MVD) format as illustrated in Fig. 1.14. Video-plus-depth data is combined with MVV data to form the MVD format, consisting of multiple 2D videos with for each 2D video has an associated depth video.

This new representation clearly targets high-quality and high-resolution in exchange of high bitrate and high complexity. MVD data processing at the sender side and at the receiver side involves a number of highly computationally intensive processing steps. Depth video data has to be estimated for N views. N 2D videos and N depth videos have to be transmitted. Finally, multiple virtual views have to be rendered from the received data.

For low complexity applications, a layered, scalable representation of the MVD can be considered, where a base layer is accessible for low complexity devices without having to cope with the whole signal.

In April 2007, the MVD format has been presented for advanced future video systems, such 3DTV and FTV [107].

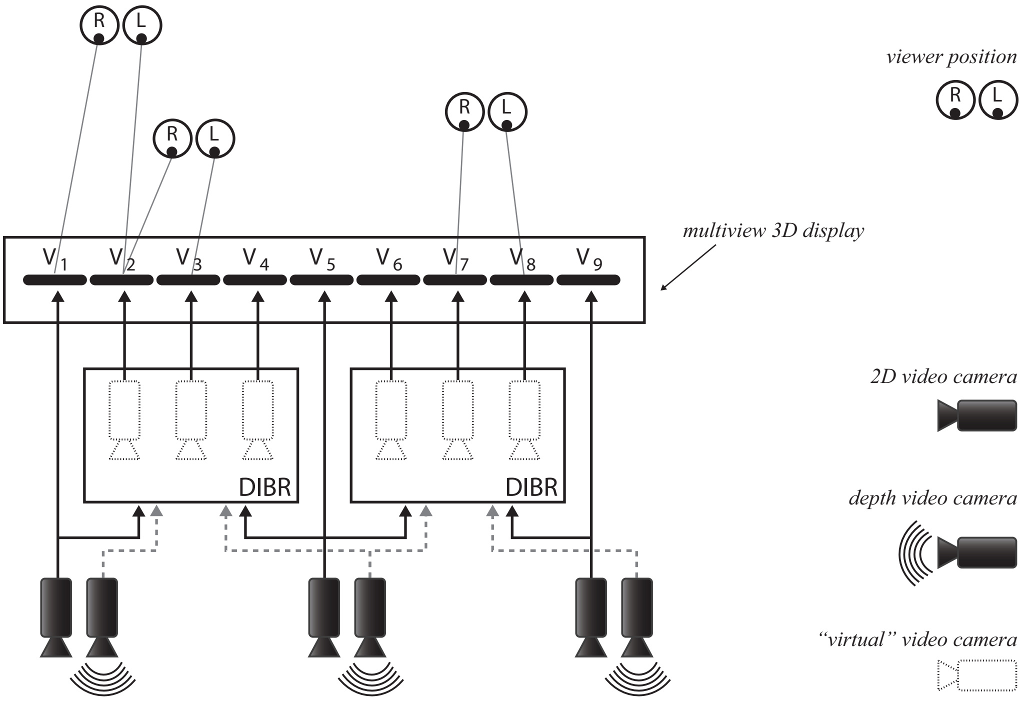


Figure 1.14: Efficient support of multiview autostereoscopic displays based on MVD content.

1.2.5 Layered depth video

layered depth video (LDV) is a derivative and an alternative to MVD representation (see Fig. 1.17). LDV is the temporally extension of the so-called layered depth image (LDI).

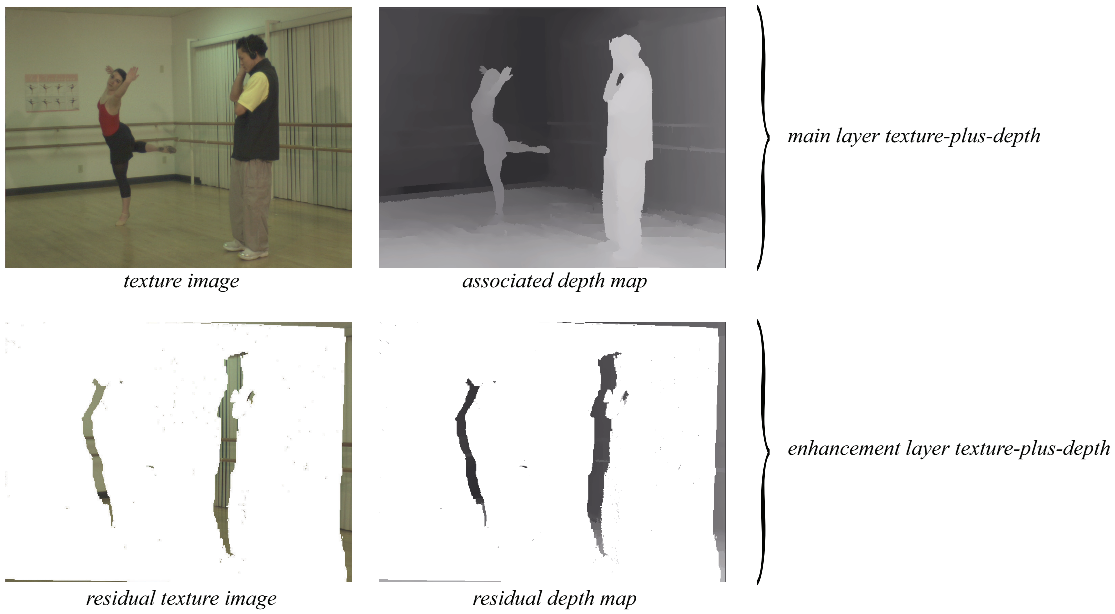


Figure 1.15: Example of LDI.

LDI uses one texture image with an associated depth map as main layer, and a enhancement layer including a residual texture image and a residual depth map as shown in Fig. 1.15. LDV can be generated from MVD by warping the main layer image onto other contributing input images (*e.g.* an additional left and right view) as illustrated in Fig. 1.16.

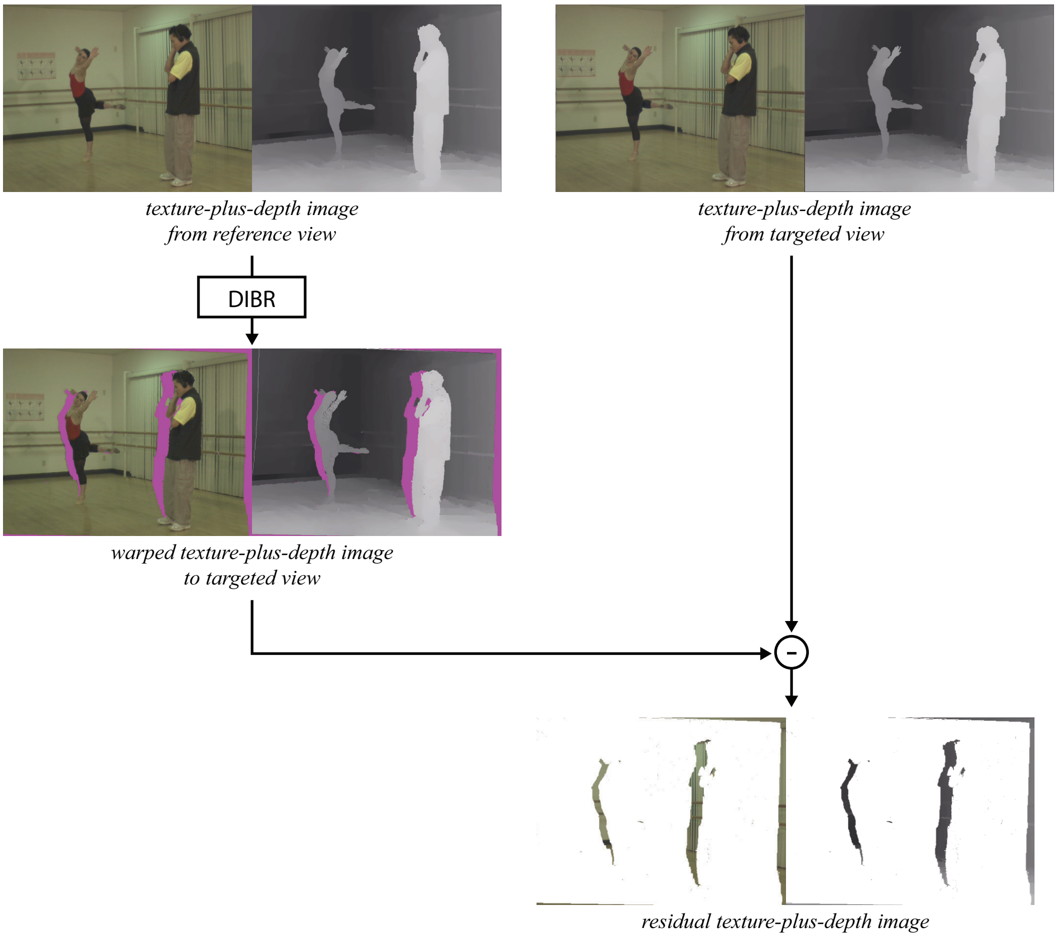


Figure 1.16: Generation of LDI data

By subtraction, it is then determined which parts of the other contributing input images are covered in the main layer image by DIBR. These are then assigned as residual images and transmitted while the rest is omitted.

Regarding complexity and usability, the same of the MVD data representation applies on LDV data.

1.3 Conclusion

In this chapter we have reviewed the 3D video data representations that may become the standard for interactive 3D video services in the next years, and thus, most material will be produced in these formats. We propose in the following reviewing 3D video coding concepts by presenting the generic predictive DPCM/DCT design, and some recent 3D video coding standards.

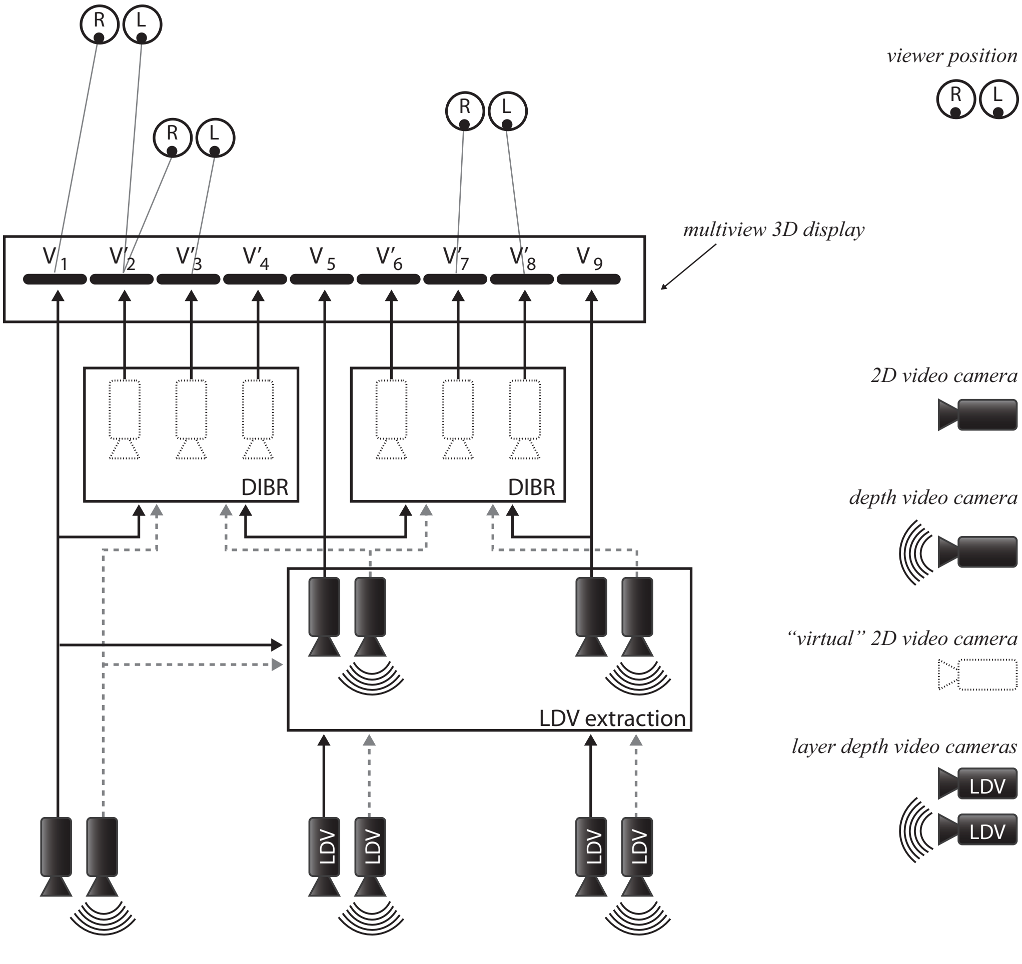


Figure 1.17: Efficient support of multiview autostereoscopic displays based on LDV content.

Chapter 2

3D video coding : a state-of-the-art

Contents

This chapter introduces and describes coding standards for encoding the 3D video data representation introduced in the previous chapter.

In the first part, we start by presenting the principles of the so-called block-based *predictive* video coding configuration, employed by most of all recent video codecs as for example the ever-popular DivX (Microsoft's MPEG-4 plus mp3) codec.

The next two parts provide an overview of the MPEG-2 and H.264/MPEG-4 AVC standards, respectively, in the second part and the third part of this chapter. The descriptions we provide here are only a brief review of the standards, which are used in the remainder of this work. For more details the reader is referred to the normative standard documents of MPEG-2 [6] and H.264/MPEG-4 AVC [57].

In the fourth part, we introduce the existing advanced 3D video coding standard, which respond to the problem of efficiency encode the huge amount of data required by 3D video communication services and their associated 3D video data.

2.1 Principles of a predictive video codec

In this section we describe the background information of a predictive video codec. Most of all the recent video codecs, as the well known MPEG-2 and DivX video codecs, employ a predictive coding configuration, that incorporates a temporal prediction (also called as DPCM), a spatial transform stage and an entropy coder. The model is often described as a hybrid DPCM/DCT codec, where the ever-popular discrete cosinus transform (DCT) is used as spatial transformation operator.

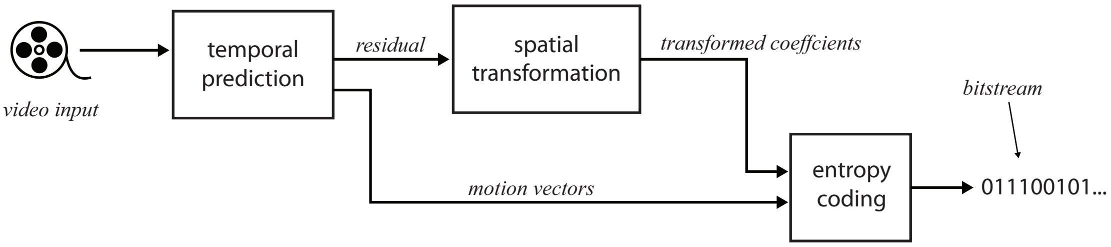


Figure 2.1: Video encoder block diagram.

Most of the visual coding standards (MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263 and H.264) since 1990's have been based on the same generic predictive DPCM/DCT design. The basic source-coding algorithm is a hybrid of *inter-picture prediction*, to exploit the temporal statistical dependencies, and *transform coding of the prediction residual* to exploit the spatial statistical dependencies.

High compression efficiency is achieved by exploiting both spatial and temporal redundancies. Temporally adjacent frames are often highly correlated. In the spatial domain neighboring pixels are very similar especially in homogeneous areas.

Fig. 2.1 shows the encoding process. A video encoder carries out three main operations: the temporal prediction, the spatial transformation and the entropy coding to produce a compressed binary stream (the bitstream).

An encoder configuration using the closed-loop motion prediction, *i.e.*, using as reference images the reconstructed images followed by spatial transformation of the residual prediction, is denoted as a predictive video encoder.

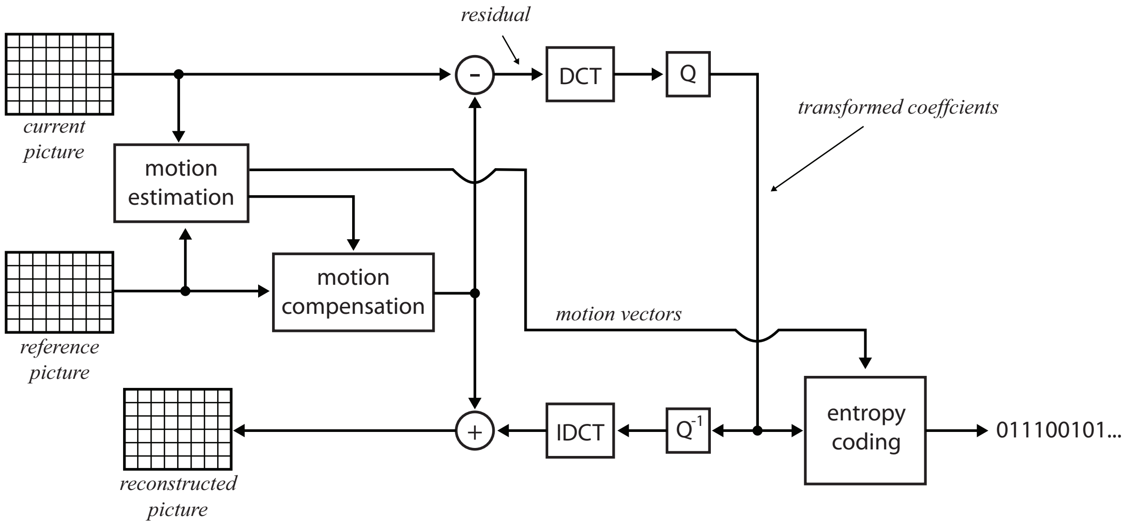


Figure 2.2: DPCM/DCT video encoder.

2.1.1 Temporal prediction

The goal of the temporal unit is to reduce the temporal correlation between adjacent frames by forming a *predicted frame*, and subtracting it from the current frame. The output of this process is the *residual frame*, that is the difference between the predicted frame and the current frame. The accuracy of the prediction can be improved by estimating the motion between adjacent frames, and then generating a motion vector field, which minimizes the residual. The predicted frame is then called the *motion compensated frame*, and the residual frame is denoted as *motion compensated error*.

The motion estimation and compensation flow is applied, as follows:

1. The current frame I_{curr} is partitioned in blocks of pixels (*e.g.* macroblocks of 16×16 pixels in MPEG).

2. The current frame I_{curr} is compared with a reference frame I_{ref}, for example the previous encoded frame. For each block, a motion estimation function finds a block of pixels in I_{ref} that matches the best the current macroblock (MB) in I_{curr}. The offset between the current MB position and the chosen reference MB is a motion vector.

3. Based on the derived motion vector field, a motion compensated predicted frame I_{pred} is generated.

4. Finally, I_{pred} is subtracted from I_{curr} to produce a residual.

2.1.2 Spatial transformation

The purpose of the spatial transformation is to spatially decorrelate the residual frame into a transform domain to reduce spatial redundancies. The choice of the transform depends on the fact that the transform should be reversible, have a low memory requirement, have a low number of arithmetic operations, etc. The transformed coefficients are quantized and provide then a more compact representation of the residual data into a small number of values. The quantization step truncates some the least significant transformed coefficients, making some coefficients null.

Many transforms have been proposed, like the Karhunen-Loeve Transform (KLT), Singular Value Decomposition (SVD), the Discrete Wavelet Transform (DWT), and the ever-popular DCT. The DCT has been retained, which is well fitted with the block-based structure.

Discrete Cosinus Transform (DCT)

Let the DCT operate on a matrix $\boldsymbol{X}$, representing a block of N×N samples and create a output matrix $\boldsymbol{Y}$, an N×N block of transformed coefficients (usually $N = 8$ or $N = 4$). The DCT (and its inverse the inverse discrete cosinus transform (IDCT)) can be described in terms of a transform matrix $\boldsymbol{A}$, such that, the relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ for a 2D separable transform is expressed as follows:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^\top \tag{2.1}$$

with the inverse 2D DCT (IDCT) by:

$$\boldsymbol{X} = \boldsymbol{A}^\top \boldsymbol{Y}\boldsymbol{A} \tag{2.2}$$

The elements of the N×N transform matrix $\boldsymbol{A}$ are defined as:

$$A_{ij} = c_i \cdot \cos\frac{(2j+1)\cdot i \cdot \pi}{2\cdot N} \quad \text{where} \quad c_i = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } i = 0, \\ \sqrt{\frac{2}{N}} & \text{else } i > 0. \end{cases} \tag{2.3}$$

Quantization

The quantization can be described as a mapping of a continuous set of values (or a very large set of possible discrete values) to a relatively small discrete and finite set. The resulting quantized signal should be possible to represent with fewer bits than the original since the range of values is smaller. The process is lossy (*i.e.* not reversible). We can distinguish two kinds of quantization: the *scalar quantization* and the *vector quantization.* A scalar quantization maps one sample of the input signal to one quantized output value, while the vector quantization maps a group of input samples (*i.e.* vectors) to a group of quantized values.

scalar quantization A simple example of scalar quantization of an input signal $\mathbf{X}$ into a quantized signal $\mathbf{Y}$, is the common *uniform quantization*:

$$\mathbf{Y} = \text{round}\left(\frac{\mathbf{X}}{Q_{step}}\right) \tag{2.4}$$

where Q_{step} is the quantization *step size.* The quantized output levels are spaced at uniform intervals of Q_{step}.

vector quantization A vector quantization maps a vector (such as a block of image samples) to a single value, *the codeword.* At the decoder, each codeword is mapped back to an approximation of the original vector. The set of all the approximated vectors are then stored at the encoder and the decoder in a *codebook.*

2.1.3 Entropy coding

The elements issued from the temporal prediction (motion vectors, mode, ...) and the spatial transformation (transform coefficients, ...), also denoted as symbols, are converted into a binary code and compressed by the entropy coder. The entropy encoder removes the statistical redundancy in the data. Note that the entropy coding is a lossless data compression scheme. Two of the most common entropy encoding techniques are Huffman coding and arithmetic coding. Entropy coding is used to exploit the "symbolic" redundancy contained in each block of the transform coefficients. This step is termed "entropy coding" to designate that the encoder is designed to minimize the source entropy.

2.1.4 Distortion measures

Since we know that most compression techniques that are operated on data are lossy, it is required an ability to measure the compression-induced distortion. However, the perceived distortion in visual content is a very difficult quantity to measure, as the characteristics of the human visual system are complex and not well understood. In practice, objective distortion models such as the sum of squared differences (SSD) or its equivalents, known as mean squared error (MSE) or PSNR, are used in most actual comparisons. They are defined between an original picture I_{org} and the reconstructed picture $\check{I}_{\text{org}}$ by:

$$SSD(I_{\text{org}}, \check{I}_{\text{org}}) = \sum_{i=1}^{H} \sum_{j=1}^{W} \left[I_{\text{org}}(i,j) - \check{I}_{\text{org}}(i,j) \right]^2, \tag{2.5}$$

$$MSE(I_{\text{org}}, \check{I}_{\text{org}}) = \frac{1}{W \times H} SSD(I_{\text{org}}, \check{I}_{\text{org}}), \tag{2.6}$$

$$PSNR(I_{\text{org}}, \check{I}_{\text{org}}) = 10 \log_{10} \left(\frac{(255)^2}{MSE(I_{\text{org}}, \check{I}_{\text{org}})} \right) \text{dB}, \tag{2.7}$$

where W and H are respectively the width and the height of the two pictures.

2.1.5 MPEG standard

MPEG is an acronym for Moving Picture Experts Group, a committee formed by the ISO (International Organization for Standardization) to develop video coding standards. MPEG was formed in 1988 to establish an international standard for the coded representation of moving pictures and association audio on digital storage media.

As illustrated in Fig. 2.3, in MPEG coding, the video sequence is first divided into groups of pictures (GOP). An MPEG picture consists of slices. A slice consists of a contiguous sequence of macroblocks (MB). A MB consists of a 16×16 block of pixels. A MB thus consists of four 8×8 blocks (sub-MB).

Each GOP may include three types of pictures as shown in Fig. 2.4:

- the intra coded pictures (I-pictures),
- the predictive coded pictures (P-pictures),
- and the bi-directionally predictive coded pictures (B-pictures).

A GOP is a coding unit of pictures. The first picture inside GOP must be an I-picture. That allows the decoding ability at the beginning of any GOP. The GOP structure contributes to provide random access as well as the ability to recovering from errors.

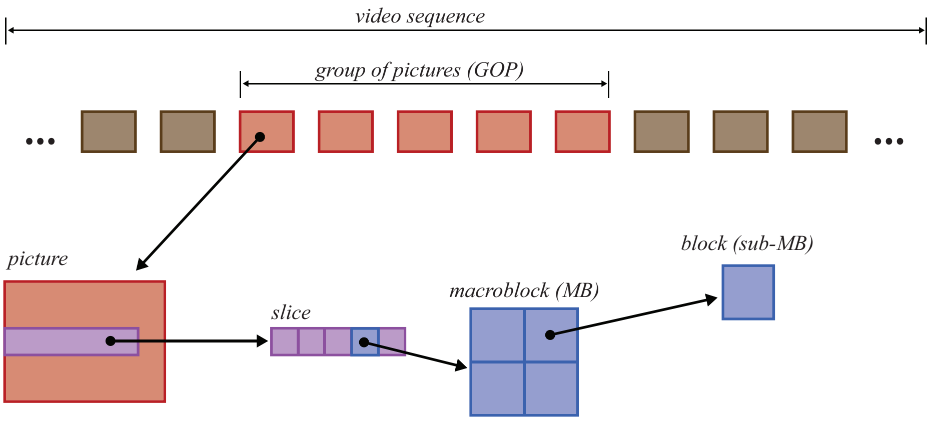


Figure 2.3: Video sequence syntax level

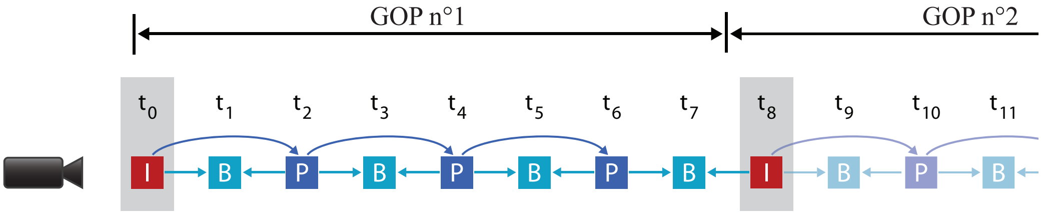


Figure 2.4: Example of GOP structure.

Intra pictures Intra pictures, or I-pictures, are coded using only information present in the picture itself, and provides potential random access points and error robustness inside the video sequence. They use no motion prediction, and only spatial transform coding.

Predicted pictures Predicted pictures, or P-pictures, are coded with respect to some previous I-picture or P-picture. This technique is known as *forward prediction*. Similarly to I-pictures, P-pictures can also serve as prediction reference for B-pictures and future P-pictures. Moreover, P-pictures use motion compensation to provide more compression efficiency than I-pictures.

Bi-directional pictures Bi-directional pictures, or B-pictures, uses both a past and future picture as a reference. This technique is called *bi-directional prediction*. B-pictures provide the best compression since they use the past and future pictures as reference. However, the computational time and delay efficiency is the largest.

2.2 MPEG-2 – Part 2: Video

In this section, we give more attention to the motion vector estimation and bitrate allocation features in MPEG-2.

2.2.1 Motion estimation

Among the various techniques for motion estimation, block matching approach has been adopted in MPEG-2 motion estimation framework due to its simplicity and effectiveness.

In this method, each frame is partitioned in non-overlapping square blocks of pixels, and motion estimation is applied on a block-by-block basis so that each block is associated with a motion vector. The motion vector of a block in the current frame I_{curr}, is estimated by minimizing the MSE (or sometimes the mean absolute error (MAE)) within a search window with respect to the reference frame I_{ref}, *e.g.*, the previous frame. The block with the minimum MSE is defined as the best match. Let $I_{\text{ref}}(x, y)$ denote the image intensity at the spatial location (x, y). The vector $\mathbf{v}(v_x, v_y)$ maps points in the current frame I_{curr} to their corresponding locations in the reference frame I_{ref}. For each MB, MSE is defined as follows:

$$\text{MSE} = \frac{1}{N^2} \sum_{x=0}^{N} \sum_{y=0}^{N} \left[I_{\text{curr}}(x, y) - I_{\text{ref}}(x + v_x, y + v_y) \right]^2 \tag{2.8}$$

where N×N is the size of the MB. Note that MPEG-2 uses half-pixel accuracy.

2.2.2 Test Model 5

Test Model 5 (TM5) [54] is recommended, but not required, by the MPEG-2 standard as the rate-control algorithm. TM5 works in three steps: target-bit allocation, rate control, and adaptive quantization.

Target-bit allocation

This step allocates the number of bits available for each GOP and before coding each picture involved in the GOP. When the first picture in a GOP is encoded, corresponding to the j^{th} picture in the video sequence, the number of remaining bits R_j to be allocated for the current GOP is updated as follows:

$$R_j = R_{j-1} + \frac{\text{bitrate}}{\text{picture\_rate}} \cdot N \tag{2.9}$$

where N is the number of pictures in the GOP, bitrate is the available bitrate of a transmission channel, picture_rate is the frequency of pictures per second. R_j thus becomes the total number of bits to be allocated for a GOP when the j^{th} picture is the first picture in the GOP.

After encoding each frame, R_j is updated as follows:

$$R_j = R_{j-1} - T_{I/P/B} \tag{2.10}$$

where $T_{I/P/B}$ is the number of bits spent for encoding the current picture. Note that the notation $T_{I/P/B}$ is used as a shorthand of T_I, T_P, and T_B respectively.

Target bits for each picture in the GOP are computed as follows:

$$T_I = \max \left\{ \frac{R_j}{1 + \frac{X_P}{X_I K_P} N_P + \frac{X_B}{X_I K_B} N_B}, \frac{\text{bitrate}}{8 \cdot \text{picture\_rate}} \right\} \tag{2.11a}$$

$$T_P = \max \left\{ \frac{R_j}{N_P + \frac{X_B K_P}{X_P K_B} N_B}, \frac{\text{bitrate}}{8 \cdot \text{picture\_rate}} \right\} \tag{2.11b}$$

$$T_B = \max \left\{ \frac{R_j}{N_B + \frac{X_P K_B}{X_B K_P} N_P}, \frac{\text{bitrate}}{8 \cdot \text{picture\_rate}} \right\} \tag{2.11c}$$

where K_P and K_B are constants that depend on quantization matrices. For the default quantization matrix $K_P = 1.0$ and $K_B = 1.4$. X_I, X_P, X_B are relative complexity measure of the three kinds of previous pictures and used in allocating target bits for the next picture.

Rate control

This step sets the reference value of the quantization parameter for each MB. The quantization parameter is obtained by measuring virtual buffer fullness, which adjusts the amount of bits for each MB. Before encoding the k^{th} MB, the virtual buffer fullness is computed as follows:

$$d_k^{I/P/B} = d_0^{I/P/B} + B_{k-1} - \frac{T_{I/P/B} \cdot (k-1)}{K} \tag{2.12}$$

where K is the number of MBs in the picture, B_{k-1} is the number of bits spent by encoding all the previous MBs in the current picture, including the $(k-1)^{\text{th}}$ one.

The reference quantization parameter Q_k that affects the quantization step size after being modulated in the next step, is calculated for the k^{th} MB as follows:

$$Q_k = \frac{d_k \cdot 31 \cdot \text{picture\_rate}}{2 \cdot \text{bitrate}}. \tag{2.13}$$

The increase of d_k makes Q_k increase, which results in decreasing the number of bits used for a MB, and *vice versa*.

Adaptive quantization

This step modulates the reference value of the quantization parameter Q_k according to the spatial activity in the MB to derive the optimal value $mquant_k$ of the quantization parameter as follows:

$$mquant_k = Q_k \cdot \frac{2 \cdot act_k + \overline{act}}{act_k + 2 \cdot \overline{act}} \tag{2.14}$$

where act_k is the spatial activity of the MB (*i.e.* minimum of the sub-MB variance), and $\overline{act}$ the average value of act_k over all the picture.

2.3 H.264/MPEG-4 AVC part 10

The latest H.264/MPEG-4 AVC standard, is similar in most respects to the MPEG-2 codec. It also consists of a hybrid of temporal and spatial prediction, in conjunction with transform coding. However, the H.264/MPEG-4 AVC video coding standard can achieve considerably higher coding efficiency compared to MPEG-2 and the previous standards. This is accomplished mainly due to the consideration of variable block sizes for the temporal prediction, multiple reference frames, intra prediction, context adaptive entropy coding, and in loop deblocking filtering. Moreover, also due to better exploitation of the spatio-temporal correlation that may exist between adjacent MBs, with the SKIP mode in predictive P-slices and the DIRECT mode in bi-predictive B-slices [125].

In this section, we restrict our attention to the block-based motion compensation for *inter* picture coding.

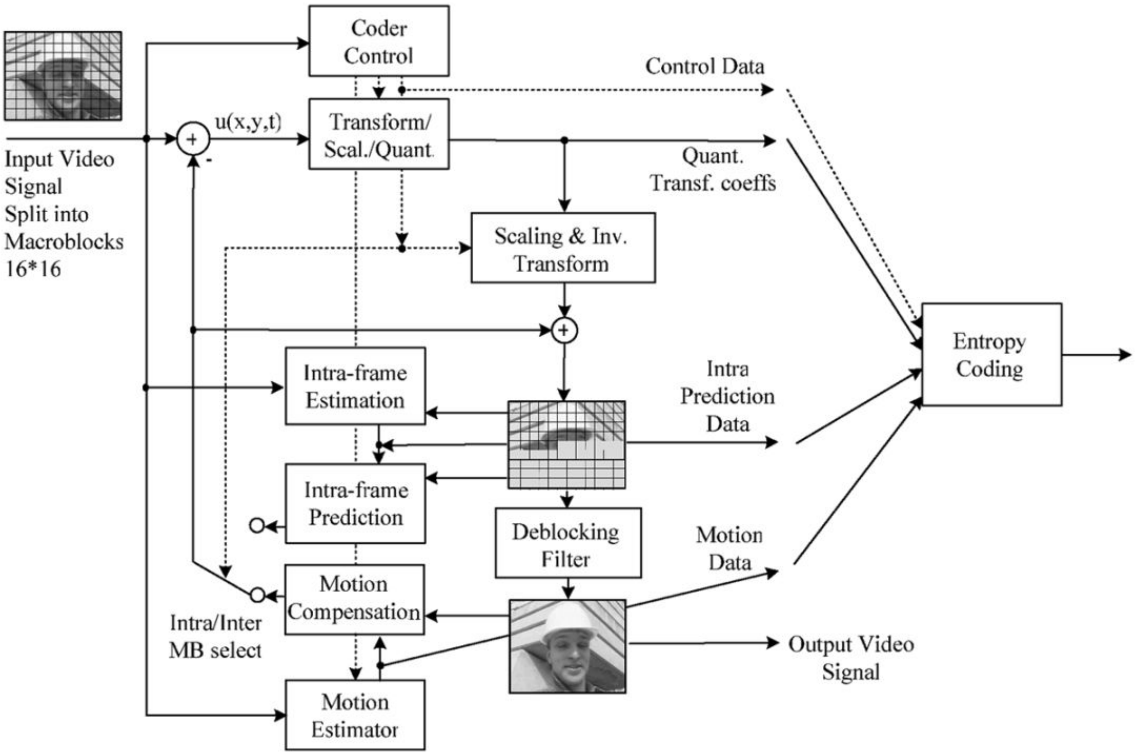


Figure 2.5: H.264 video encoder scheme

2.3.1 The Lagrangian cost

When coding a MB, many coding options (motion vector, reference frame, direction of prediction in B-slice, ...) have to be assigned in order to optimize the encoding process of the current frame. To choose for each coding option the best one, a rate-distortion optimization (RDO) based on a Lagrangian function cost is performed for each option, and the option that gives the smallest cost is selected. The problem of finding the best combination of coding options can be formulated as minimizing the following Lagrangian cost function:

$$J = \textbf{Distortion} + \lambda \cdot \textbf{Rate} \tag{2.15}$$

where the **Distortion** measurement quantifies the quality of the reconstructed MB while the **Rate** measures the bits needed to encode the MB with a particular combination of coding options. Here, $\lambda \geq 0$ is the Lagrangian multiplier.

Lagrangian multiplier By considering a given quantization parameter Q, H.264/MPEG-4 AVC employs two different Lagrangian multipliers: λ_{MODE} and λ_{MOTION}.

λ_{MODE} is used in computing the cost of a prediction mode and selects the best one according to the slice type. For *Intra* I-slice mode testing or *inter* P-slice mode testing, λ_{MODE} is given by:

$$\lambda_{\text{MODE}_{\text{I,P}}} = 0.85 \cdot 2^{(Q-12)/3} \tag{2.16}$$

For *inter* B-slice mode testing, λ_{MODE} becomes:

$$\lambda_{\text{MODE}_{\text{B}}} = \max\left(2, \min\left(4, \frac{Q-12}{6}\right)\right) \cdot \lambda_{\text{MODE}_{\text{I,P}}} \tag{2.17}$$

On the other hand, λ_{MOTION} is used in computing the motion vector in *inter* P-slice or B-slice according to the distortion measurement. If the distortion measurement is the SSD, the Lagrangian multiplier is given by:

$$\lambda_{\text{MOTION}} = 0.85 \cdot 2^{(Q-12)/3} = \lambda_{\text{MODE}} \tag{2.18}$$

Else if the distortion measurement is the sum of absolute differences (SAD) or the sum of absolute transformed differences (SATD), the Lagrangian multiplier is:

$$\lambda_{\text{MOTION}} = \sqrt{0.85 \cdot 2^{(Q-12)/3}} = \sqrt{\lambda_{\text{MODE}}} \tag{2.19}$$

The distortion measurements SSD, SAD and SATD are explained in the next section.

Distortion measurements: SAD, SATD, SSD

Let us consider three picture entities with a size of W×H pixels:

- the original picture I_{org},
- the predicted picture I_{pred}. For example I_{pred} may be a motion compensation of the reference picture I_{ref} with the motion vector field $\mathbf{v}(v_x, v_y)$, where I_{pred} is expressed as $I_{\text{pred}} = I_{\text{ref}}(\mathbf{v}) = \sum_{x=1}^{W}\sum_{y=1}^{H} I_{\text{ref}}(x+v_x, y+v_y)$.
- the reconstructed picture $\check{I}_{\text{org}}$ which is the result of a transform, quantization, inverse quantization and inverse transform of the original image I_{org}.

In the JM software [124], the sum of absolute differences (SAD) measurement is used when computing full-pel motion estimation, while the sum of absolute transformed differences (SATD) is used for sub-pel motion estimation. On the other hand, the sum of squared differences (SSD) is used when evaluating the reconstructed picture $\check{I}_{\text{org}}$, while SAD and $SATD$ is used when evaluating the quality of a prediction of I_{org} in full-pel and sub-pel motion search, respectively.

The SAD between a picture and its prediction is given by the expression:

$$SAD\big(I_{\text{org}}, I_{\text{pred}}\big) = \sum_{x=1}^{W}\sum_{y=1}^{H} \big|I_{\text{org}}(x,y) - I_{\text{pred}}(x,y)\big| \tag{2.20}$$

Since the residual is spatially transformed by an integer DCT, which is approximately unitary, the energy in the spatial domain can be estimated by the energy in the transform domain. We recall that the 4×4 Hadamard transform matrix H is defined as:

$$H = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \tag{2.21}$$

This technique is used when it may be too computationally intensive to perform a complete RDO using the DCT and the IDCT.

The SATD is given by:

$$SATD\big(I_{\text{org}}, I_{\text{pred}}\big) = \sum_{i=1}^{4}\sum_{j=1}^{4} |c(i,j)| \tag{2.22}$$

where $c(i,j)$ denotes the element of C at the location (i,j), which is the Hadamard transform of the difference between the original picture I_{org} and the prediction picture I_{pred} defined as:

$$C = H \cdot \left(I_{\text{org}_{4\times4}} - I_{\text{pred}_{4\times4}} \right) \cdot H^{\top} \tag{2.23}$$

where $I_{\text{org}_{4\times4}}$ and $I_{\text{pred}_{4\times4}}$ are respectively a sub-MB of 4×4 pixels of the original picture I_{org} and the prediction picture I_{pred}.

Finally, the SSD between the original picture I_{org} and the reconstructed picture $\check{I}_{\text{org}}$ is given by:

$$SSD\big(I_{\text{org}}, \check{I}_{\text{org}}\big) = \sum_{i=1}^{H} \sum_{j=1}^{W} \left[I_{\text{org}}(i,j) - \check{I}_{\text{org}}(i,j) \right]^2 . \tag{2.24}$$

2.3.2 *Inter* prediction

Macroblock prediction mode decision

H.264/MPEG-4 AVC supports seven MB modes. Each MB can be partitioned into blocks of different sizes (16×16, 16×8, 8×16, 8×8). An 8×8 block can further be sub-partitioned into sub-MBs with sizes 8×8, 8×4, 4×8 and 4×4 as shown in Fig. 2.6.

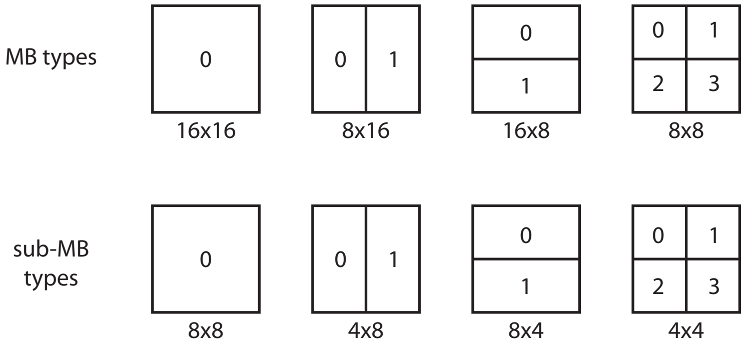


Figure 2.6: Macroblock partitions.

Each block is motion compensated using a separate motion vector by partition.

Motion vector predictor Compression of the motion vector field may be improved by predicting each motion vector from previously coded vectors. A motion vector predictor **mvp** is estimated, and the motion vector difference between the motion vector predictor **mvp** and the current vector **mv** is then encoded and transmitted. The motion vector predictor **mvp** is derived using the median of the motion vectors of the adjacent blocks (Fig. 2.7) on the left (A), top (B), and top-right (C). The location on the top-right is replaced with the top-left location (D), if it is unavailable.

Best motion vector The motion search is computed by minimizing the following Lagrangian cost function:

$$J\big(\mathbf{mv}|\lambda_{\text{MOTION}}\big) = SAD\Big(I_{\text{org}}, I_{\text{ref}}\big(\mathbf{mv}_{\text{ref}}\big)\Big) + \lambda_{\text{MOTION}} \cdot R\big(\mathbf{mv}_{\text{ref}} - \mathbf{mvp}_{\text{ref}}\big) \tag{2.25}$$

In sub-pel motion search, the distortion measurement is replaced by the SATD measurement.

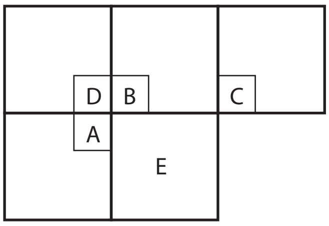


Figure 2.7: Motion vector prediction for E is the median value of A, B and C or D depending on position.

Best reference frame Finding the best reference frame I_{ref} and the associated motion vectors for the various *inter* modes, is done after motion estimation with regard to all reference frames by minimizing the following Lagrangian cost function:

$$J\big(\text{ref}|\lambda_{\text{MOTION}}\big) = SATD\Big(I_{\text{org}}, I_{\text{ref}}\big(\mathbf{mv}_{\text{ref}}\big)\Big) + \lambda_{\text{MOTION}} \cdot \Big(R\big(\mathbf{mv}_{\text{ref}} - \mathbf{mvp}_{\text{ref}}\big) + R\big(\text{ref}\big)\Big) \tag{2.26}$$

with ref being the frame index of the reference frame I_{ref}, $\mathbf{mv}_{\text{ref}}$ being the motion vector associated to the reference frame I_{ref} and $\mathbf{mvp}_{\text{ref}}$ the predicted motion vector. The rate term $R(\mathbf{mv} - \mathbf{mvp})$ represents the number of bits required to encode and transmit the predicted motion error, and $R(\text{ref})$ the number of bits associated with the reference frame index ref.

Multiple reference frames

Multiple reference frames approach extends the motion compensated prediction thereby permitting the use of more reference frames than just one as illustrated in Fig. 2.8. The use of multiple reference frames in most cases provides an improved prediction gain.

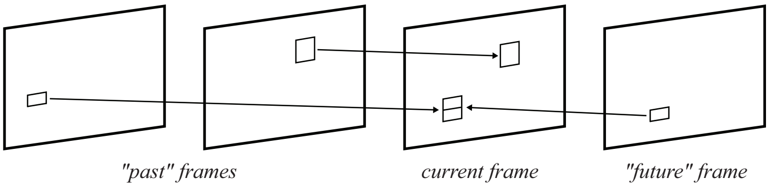


Figure 2.8: Multiple reference frames.

H.264/MPEG-4 AVC manages a decoded picture buffer (DPB) filled up with decoded frames for use as reference frames with regard to the current frame for predictive coding.

Bi-prediction

By considering the bi-predictive nature of B-slices, each MB may be predicted from one or two reference frames, before or after the current picture in temporal order as illustrated in Fig. 2.9. In the case of two reference frames, an average of two reference frames can be used as prediction.

The B-slices use two lists of previously coded reference frames, denoted as `LIST_0` and `LIST_1`. With respect to the current frame, `LIST_0` and `LIST_1` contain respectively the frames before and after the current frame in temporal order. These frames are also known

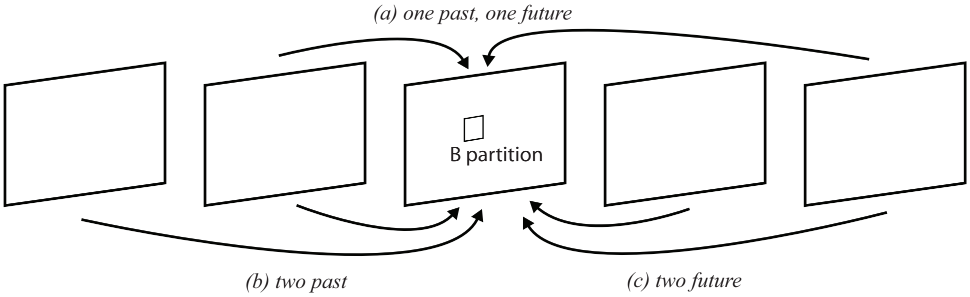


Figure 2.9: Bi-prediction examples in a B MB type: (a) past/future, (b) past, (c) future.

as *short term* frame and *long term* frame when belonging respectively to the LIST_0 and LIST_1.

Therefore, two motion compensated frames I_{pred_0} and I_{pred_1} are obtained respectively from the LIST_0 and the LIST_1, and hence using two motion vector fields. An average of these two predicted frames is calculated by using the following equation:

$$I_{\text{pred}} = \sum_x \sum_y \big(I_{\text{pred}_0}(x,y) + I_{\text{pred}_1}(x,y) + 1\big) >> 1 \qquad (2.27)$$

where I_{pred} is a bi-predictive motion compensated frame which will be subtracted from the current frame to obtain the motion compensated residual.

Furthermore, for each MB the prediction samples $I_{\text{pred}_0}(x,y)$ and $I_{\text{pred}_1}(x,y)$ can be scaled by a weighting factor w_0 or w_1. This method is denoted as *weighted prediction.* The weighting factors may be determined by the encoder and transmitted in the slice header. Or they can be estimated and implicitly transmitted to the decoder. They are calculated by a function depending on the relative temporal position of the LIST_0 and LIST_1 reference frames. For example, weighted prediction may be effective in coding fading transitions, where one scene fades into another.

SKIP mode and DIRECT mode

More bitrate savings may be achieved by identifying some motion information which can been skipped and retrieved at the decoder side. In this case, neither side information (motion vector, reference picture index, prediction mode, ...) nor the residual has to be coded. Only a flag in the slice header indicates the skipped nature of the MB to the decoder. This configuration has been introduced within P-slices and B-slices, and denoted as SKIP mode and DIRECT mode, respectively, when using the correlation between motion vectors from spatial and temporal adjacent MBs.

SKIP mode By only considering the spatial correlation inside the current frame, the cost function J_{SKIP} is calculated as follows:

$$J_{\text{SKIP}} = SSD\Big(I_{\text{org}}, I_{\text{ref}}\big(\mathbf{mvp}\big)\Big) + R_{\text{SKIP}} \qquad (2.28)$$

where R_{SKIP} is the number of bits used to signal the MB SKIP flag to the decoder. In general R_{SKIP} is very near to zero (*e.g.* only 1 bit).

Temporal DIRECT mode The *temporal* DIRECT mode uses bi-directional prediction and allows residual coding of the prediction error. Let us recall that B-slices use two lists of reference frames, denoted as `LIST_0` and `LIST_1`. To retrieve at the decoder, the forward and backward motion vectors $\mathbf{mv}_{l0}$ and $\mathbf{mv}_{l1}$, the motion vector $\mathbf{mv}_c$ of the co-located block in the first frame of the `LIST_1` is scaled according to the temporal distance, with regard to the reference frames involved (see Fig. 2.10). The motion vectors for DIRECT mode, $\mathbf{mv}_{l0}$ and $\mathbf{mv}_{l1}$, are estimated as follows:

$$\mathbf{mv}_{l0} = \frac{TD_B}{TD_D} \times \mathbf{mv}_c$$
$$\mathbf{mv}_{l1} = \frac{TD_B - TD_D}{TD_D} \times \mathbf{mv}_c$$

where TD_B is the temporal distance between the current B-frame and the frame pointed by $\mathbf{mv}_c$ (*i.e.* the frame in the list 0), and TD_D is temporal distance between the `LIST_0` and the first frame of the `LIST_1`.

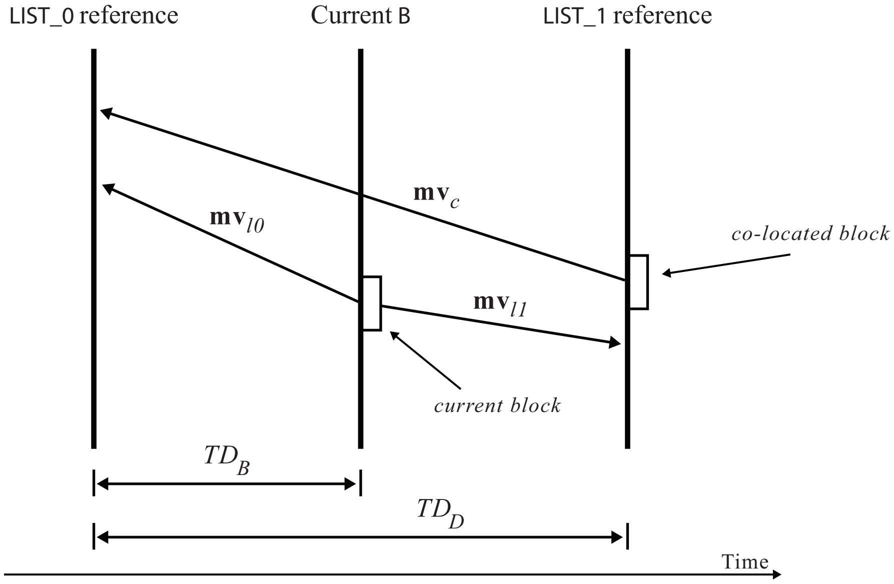


Figure 2.10: *Temporal* DIRECT prediction in B-slice coding.

2.4 Advanced video standards for 3D services

In this section we investigate the suitable video coding schemes for compression and transmission that are based on the 3D video formats introduced in the previous chapter.

2.4.1 Conventional stereo video coding

The pictures from a stereo video sequence are very similar, which makes them well suited for compression, for example when one predict the other.

MPEG-2 MVP

In the case of a stereo system, a multiview profile (MVP) has been defined in MPEG-2 standard, which allows the transmission of two video signals for stereoscopic TV applications. One of the main features of the MVP is the use of scalable coding tools to guarantee the backward compatibility with the MPEG-2 Main Profil. The MVP relies on a multi-layer representation such that one view is designed as the base layer and the other view is assigned as the enhancement layer. Also, the MVP conveys the camera parameters (*i.e.* geometry information, focal length, ...) in the bitstream. The base layer is encoded in conformance with the Main Profil, while the enhancement layer is encoded with the scalable coding tools. An example of the prediction structure for the MVP is shown in Fig. 2.11.

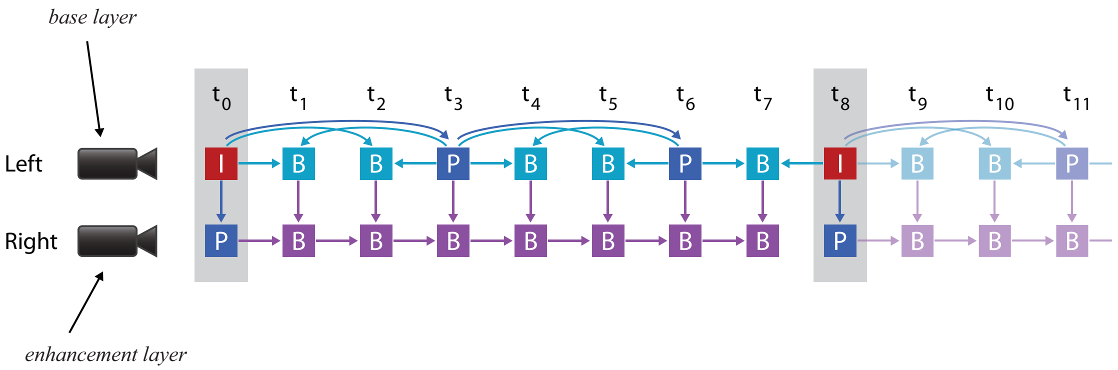


Figure 2.11: Prediction structure in MPEG-2 MVP using a GOP structure IBBP.

As we can see, temporal prediction only is used on the based layer, while temporal prediction and inter-view prediction are simultaneously performed on the enhancement layer. As a consequence, backward compatibility with legacy 2D decoders is achieved, since the the base layer represents a conventional 2D video sequence.

H.264/MPEG-4 AVC stereo SEI message

As discussed in the previous chapter, a stereo video data may be represented by stereo interleaving techniques, thereby time multiplexing or spatial multiplexing. This information has to be carried to the decoder by someway, so that the decoder will be able to distinguish the left and right view inside the multiplexed bitstream. With the help of the stereo video supplemental enhancement information (SEI) message defined in H.264/MPEG-4 AVC fidelity range extensions (FRExt) [76], a decoder can easily identify the left view and the right view, detect the multiplexing of the stereo video sequence, and so, extract distinctly the two views. Similar type of signaling has been introduced in the multiview video coding (MVC) extension of H.264/MPEG-4 AVC [127].

The overview diagram in Fig. 2.12 illustrates the coding procedure of H.264/MPEG-4 AVC SEI for a conventional stereo video pair. These two video sequences are interlaced line-by-line into one sequence, where the top field contains the left view and the bottom field the right view. The H.264/MPEG-4 AVC coder is applied to the interlaced sequence in field coding mode, resulting in one encoded bitstream. After transmission over the channel this stream is decoded, resulting in the distorted interlaced sequence. For output, this sequence is deinterlaced into the two individual view video sequences.

However, this approach breaks the existing 2D decoders, *i.e.*, no backwards compatibility is supported. If the transmitted bitstream is not demultiplexing, it is not possible for traditional 2D devices to extract, decode and display a 2D version of the 3D video content.

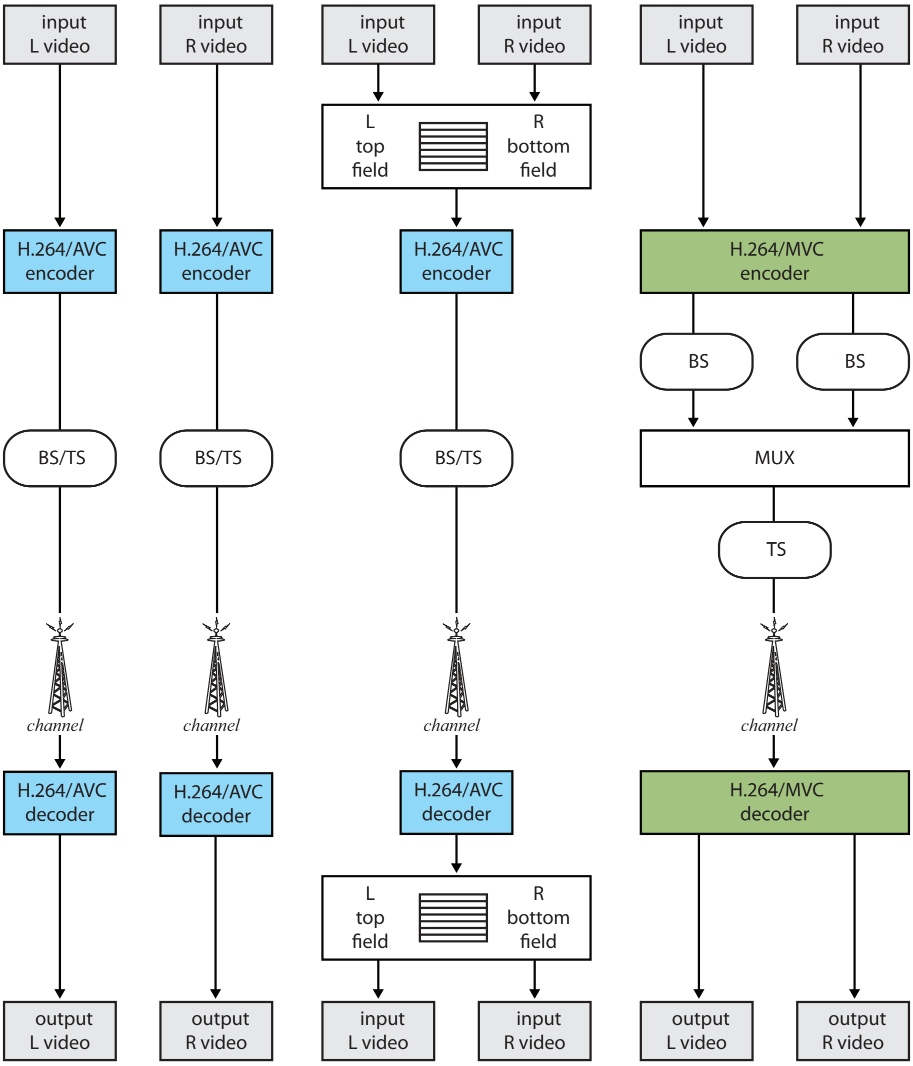


Figure 2.12: Schematic block diagrams for H.264/AVC Simulcast (left), H.264/AVC Stereo SEI Message (middle) and H.264/MVC (right) coding with stereo video format data.

2.4.2 Coding of video-plus-depth

MPEG-C Part 3

MPEG has presented the MPEG-C Part 3 (also referred as ISO/IEC 23002-3) specification, which standardizes the video-plus-depth coding [91]. This specification is based on the encoding of a 3D content inside a conventional MPEG-2 transport stream, which includes the texture video, the depth video and some auxiliary data. This standardized solution responds to the broadcast infrastructure needs. It provides interoperability of the content, display technology independence, capture technology independence, backward compatibility, compression efficiency and the ability of the user to control the global depth range.

Due to the very nature of the depth data, the smoothed gray level representation leads to a much higher compression efficiency than the texture video. Thus, only a small extra bandwidth is needed for transmitting the depth map. The total bandwidth for video-plus-depth data transmission is then reduced compared to the stereo video data, and some depth coding experiments [35] have shown that the depth bitstream bitrate required for good depth representation is around a rough number of 10-20% of the texture bitstream bitrate. Moreover, it does not introduce any specific coding algorithms. It supports different coding formats like MPEG-2 and H.264/MPEG-4 AVC. It is only necessary to specify high-level syntax that allows a decoder to interpret two incoming video streams correctly as texture and depth data.

A key advantage is that the MPEG-2 bitstream provides backward compatibility with legacy 2D devices.

MPEG-4 MAC

Another tool for encoding the video-plus-depth data are the multiple auxiliary components (MAC) defined by the version 2 of MPEG-4. Basically, the MAC is the grayscale shape that is not only used to describe transparency of the video object, but can also be defined in more general way to describe shape, depth shape or others secondary texture. It is also known as the alpha channel. Therefore, the depth video can be used as one of the auxiliary components [62].

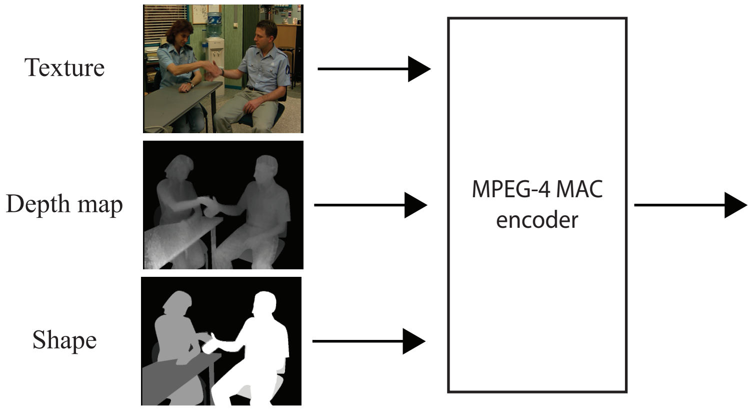


Figure 2.13: MPEG-4 MAC architecture

The encoding of the auxiliary components is similar to the texture component, which employs motion compensation and DCT. It uses the same motion vectors of the 2D video for the motion compensation of the auxiliary components. The configuration of MPEG-4 MAC to encode the 2D video and the depth information is shown in Fig. 2.13.

Furthermore, this coding scheme can be also be used for stereo video coding as proposed in [24], where the disparity vector field, the luminance and chrominance data of the residual texture are assigned as 3 components of MAC.

MPEG-4 AFX

Some of the computer graphics researches have been integrated into an extension of MPEG-4 called animation framework extension (AFX) [55]. Three of the new tools are of specific interest in the scope of 3D video: depth image based rendering, point rendering and view-dependent multi-texturing. Note that MPEG-4 AFX utilizes and enhances existing MPEG-4 tools without sacrificing backward compatibility.

Up to now AFX specifies a DepthImage structure, which consists of a computer graphics centric camera definition (*i.e.* position, orientation, field of view, near clipping plane, far clipping plane, etc.) and a pointer to a depth image. This can either be a SimpleTexture (*i.e.* a DI) or a PointTexture (*i.e.* an LDI). For a SimpleTexture, the texture and depth fields can be comprised of either an ImageTexture, a MovieTexture or a PixelTexture, as defined in the MPEG-4 Video/System documents. A PointTexture is comprised of a) a texture which stores for each pixel the number of layers as well as the color values for each layer; b) a depth map which stores for each pixel the 4-byte depth values for each layer. In either case, the depth values should be normalised to the distance from the near to the far clipping plane of the camera. Note that the format could also be used for animated objects (*i.e.* sequences) by storing sets of compressed video streams instead of images, together with "streams" of depth maps. For compression, they simply state that still and video coding formats of MPEG-4 should be used for textures and depth maps.

2.4.3 MVC extension of H.264

For a general case of two or more number of views, the JVT is developing a multiview extension of H.264/MPEG-4 AVC standard, known as MVC extension. The solutions which have been adopted in the draft MVC specification [128], provide new techniques to improve coding efficiency, to reduce decoding complexity and memory consumption for various applications.

Also, it is mandatory for the compressed multiview bitstream to include a base layer stream that could be easily extracted and used for backward compatibility with legacy 2D devices.

View dependency

A typical prediction structure of MVC, using both hierarchical temporal prediction scalability and inter-view prediction is shown in Fig. 2.14. There may be views which are dependent on other views to be accessed and decoded. For example in Fig. 2.14, the view 4 depends on the view 3 and the view 5, where the view 5 depends on the view 3, and the view 3 depends on the view 1.

The view dependencies are defined for each camera, and transmitted through the sequence-level header in the sequence parameter sets (SPS) MVC syntax.

The base view In Fig. 2.14, the first view, known as *base view*, is independently coded, as in a simulcast coding, *i.e.*, only temporal redundancies are exploited. Only the base view can provide synchronization and random access features by coding the key picture in *intra* mode, denoted as *anchor* picture. The other pictures are *non-anchor* pictures.

The non-base views The remaining views are encoded, denoted as *non-base views*, by motion and disparity compensation to exploit respectively temporal and inter-view dependencies. As a consequence, the coding efficiency is improved at the cost of losing random access for each view. To overcome this issue, V-pictures have been introduced in order to allow inter-view random access. A V-picture is then defined as a picture having no temporal dependence on other pictures in the same view, and may only be predicted from pictures in other cameras at the same time. In the non-base view the V-picture refers to the temporal key-picture.

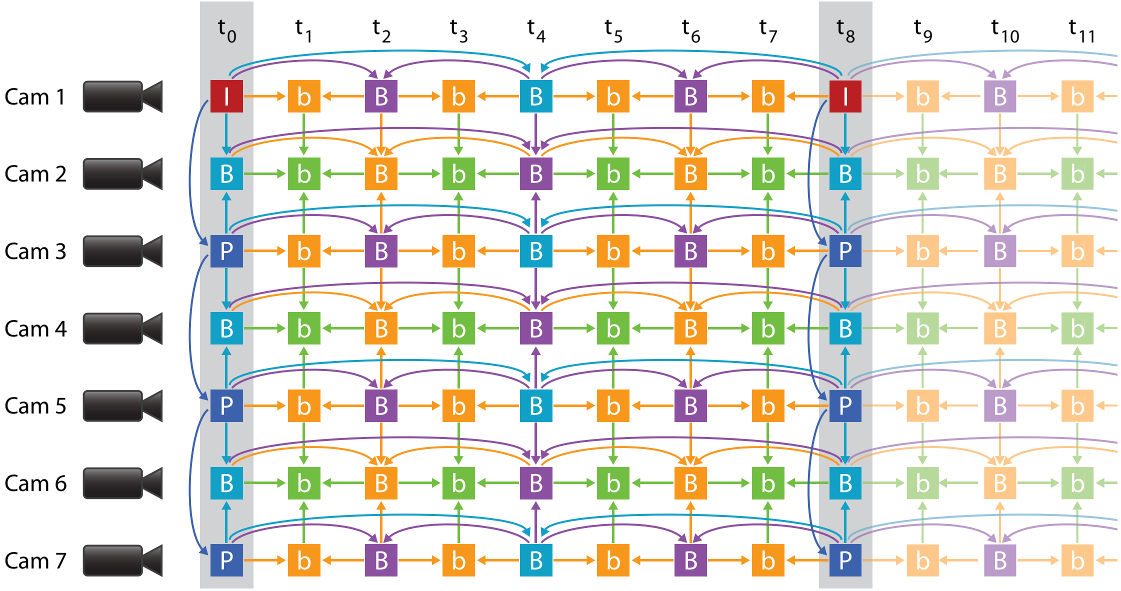


Figure 2.14: Typical MVC prediction structure.

Random access and view switching

Random access ability in MVC refers to the possibility to access a given picture in a required view at a specified time through minimal decoding of other pictures in dependent views. instantaneous decoding refresh (IDR) pictures are natural random access points. In the base view, IDR pictures refers to the anchor pictures, whereas in the non-base view, IDR pictures refers to the V-pictures, denoted as view-IDR (V-IDR) pictures [19]. IDR pictures can be considered as temporal/inter-view key picture, while V-IDR picture as temporal key picture.

Temporal/inter-view prediction

As shown in the view dependency structure in Fig. 2.14, a natural way to improve compression efficiency of multiview video content is to exploit temporal as well as inter-view statistical dependencies between adjacent cameras. Consequently, MVC takes advantage of the redundancies among the inter-pictures of one camera and the inter-view pictures of other cameras, and leads to an additional coding gain compared to the H.264/MPEG-4 AVC simulcast solution [79].

Inter-view prediction The inter-view correlation between adjacent cameras is removed via the so-called disparity estimation and compensation. MVC employs a variable block-based estimation just like the temporal prediction in H.264/MPEG-4 AVC. Rooms for improvement can be expected, since no geometrical constraints between views are taken into account.

Multiview reference picture list manager The MVC extension makes a good use of the multiple reference frames feature of H.264/MPEG-4 AVC, and more specifically of the DPB (as discussed in Section 2.3.2), by inserting and deleting among the temporal reference frames, frames from adjacent views [18, 20]. Therefore, the temporal plus the inter-view correlations are exploited for combined motion/disparity compensated prediction, where a predictive frame is not only predicted from temporally neighboring frames but also from corresponding frames in the adjacent views.

Inter-view SKIP mode As an inter-view extension of the classical temporal SKIP mode (see Section 2.3.2), inter-view SKIP mode is motivated by the idea of exploiting the similarity between motion vectors between neighboring views. With this mode, the motion information of the current MB is derived from the corresponding MB in the picture at the same temporal index of the neighboring view [64, 136].

The first step consists in estimating a global disparity vector (GDV) $\mathbf{D}_G(x,y)$ with 8-pel accuracy between the current picture I_t^{Vcurr} and the inter-view reference picture I_t^{Vref}, by minimizing the matching error:

$$\mathbf{D}_G(x,y) = \underset{-S\leq x,y\leq S}{\arg\min} \left\{MAD\left(8\cdot x, 8\cdot y\right)\right\} \tag{2.29}$$

where S is the search range with 8-pel accuracy. In a common trade-off between coding performance and computational complexity, S is set to 12×8-pel. The mean absolute difference (MAD) used to evaluate the matching error is defined as:

$$MAD(x,y) = \frac{1}{(H-y)(W-x)} \sum_{i=1}^{W-x} \sum_{j=1}^{H-y} \left|I_t^{\text{Vcurr}}(i,j) - I_t^{\text{Vref}}(x+i, y+j)\right| \tag{2.30}$$

with W and H being the width and height of the two pictures.

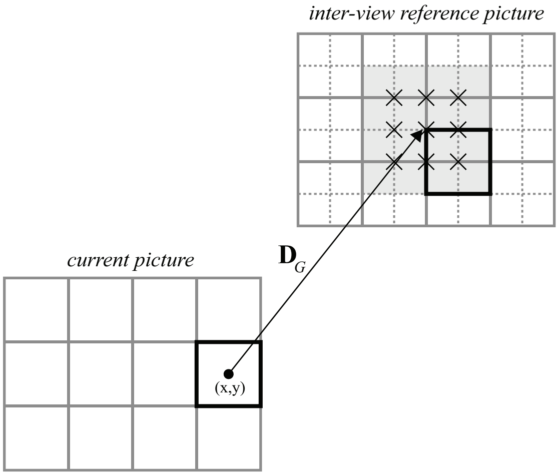


Figure 2.15: 8-pel accuracy motion matching.

After the estimation of $\mathbf{D}_G$, a finest motion matching is performed to find the optimal global disparity vector as shown in Fig. 2.15, where the grid with solid line indicates 16×16 MB, and the grid with dotted line indicates 8×8 sized blocks. As illustrated in Fig. 2.15, there are totally 9 motion vector candidates within the search window. Among the set of motion candidates, the one with optimum rate-distortion (RD) performance is selected. The side information is the offset value of the location of the block that contains the optimum motion vector relative to the center of the search window.

Compensation of illumination

The difference of viewpoints inside a multiview video sequence, may provoke illumination changes between the video sequences. To compensate the inter-view illumination changes between the pictures, the illumination change-adaptive motion compensation (ICA MC) has been proposed and implemented [66].

After partitioning of the pictures in blocks, let us consider the current block B_t^{Vcurr} and the inter-view reference block B_t^{Vref}. The conventional SAD calculation for the motion estimation of a S×T block (such as 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4) is performed as follows:

$$SAD(B_t^{\mathrm{Vcurr}}, B_t^{\mathrm{Vref}}(\mathbf{v})) = \sum_{i=1}^{S}\sum_{j=1}^{T} |B_t^{\mathrm{Vcurr}}(i,j) - B_t^{\mathrm{Vref}}(i+v_x, j+v_y)| \tag{2.31}$$

where $\mathbf{v}(v_x, v_y)$ is a motion vector. In order to take into account the change of illumination into the inter-view compensation process, a new distortion measurement NewSAD has been defined as follows:

$$NewSAD(B_t^{\mathrm{Vcurr}}, B_t^{\mathrm{Vref}}(\mathbf{v})) = \sum_{i=1}^{S}\sum_{j=1}^{T} |(B_t^{\mathrm{Vcurr}}(i,j) - M_{\mathrm{curr}}) - (B_t^{\mathrm{Vref}}(i+v_x, j+v_y) - M_{\mathrm{virt}})| \tag{2.32}$$

where M_{curr} and M_{virt} are the average pixel values of the current block and reference block, respectively.

Single loop decoding

When *target* views are decoded from an MVC bitstream for display, with regard to the view dependency, some views, denoted as *reference* views, may not be displayed but are needed for the inter-view compensation and decoding of the target views. The original MVC standard required that the pictures of the reference views have to be fully decoded and stored, known as multiple loop decoding (MLD). This involves both high decoding complexity and high memory consumption for the reference pictures that are not displayed. To overcome this issue a single loop decoding (SLD) scheme [21] has been introduced in MVC standard. SLD requires only partial decoding of pictures in reference views.

To achieve SLD, during the multiview encoding process, for each temporal *non-key* picture inside the reference view, the inter-view SKIP mode is applied for all the inter-view prediction. Therefore, when decoding the temporal *non-key* picture inside the reference views, the MBs are not fully decoded, but just the motion information is stored and used for decoding the corresponding MBs in the target views. Only the target views are fully decoded.

Experimental results showed that the SLD scheme implies however a slightly compression efficiency loss compared to MLD, but reduces significantly the decoder complexity and memory usage.

2.4.4 MVD coding

So far, MVD standard are still under investigation inside the JVT. The available and emerging specifications, such MPEG-C Part 3 and MVC, do not support efficiently the

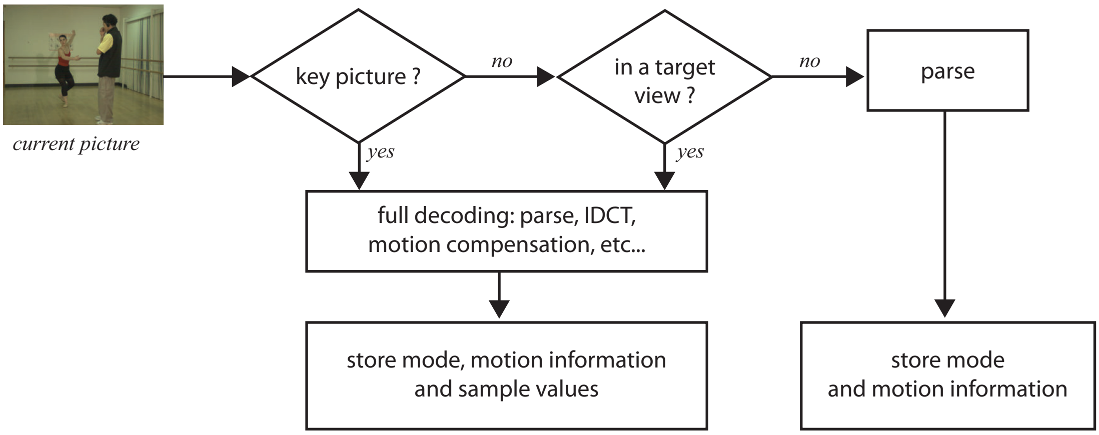

current picture
key picture ?
no
in a target view ?
no
parse
yes
yes
full decoding: parse, IDCT, motion compensation, etc...
store mode, motion information and sample values
store mode and motion information

Figure 2.16: SLD scheme in MVC

MVD data representation. An extension of MVC or/and MPEG-C part 3, or a combination of both may be envisaged [107].

2.5 Conclusion

2.5.1 MPEG video coding standards

A basic comparison of features is presented in Table 2.1.

Table 2.1: Comparison of MPEG-2 and H.264/MPEG-4 AVC.

Features	MPEG-2	H.264/MPEG-4 AVC
Macroblock size	16×16 (frame mode) 16×8 (field mode)	16×16
Block size	8×8	16×16, 8×16, 16×8, 8×8, 4×8, 8×4, 4×4
Transform	DCT	4×4 integer transform
Entropy coding	VLC	VLC, CAVLC and CABAC
Pel accuracy	Integer 1/2-pel	Integer 1/2-pel, 1/4-pel
Reference frame	Yes one ref. frame	Yes multiple ref. frames
Picture type	I, P, B	I, P, B, SI, SP
Transmission rate	2 - 15 Mbps	64 kbps - 150 Mbps
Encoder complexity	Medium	High
Backward compatibility with previous standards	Yes	No

2.5.2 3D video coding standards

As we have seen in this chapter, the advanced 3D video coding standards are mainly the 3D extension of existing 2D video coding standards to support the 3D application requirements.

Some tools are still under consideration like:

- the adaptive reference filtering to compensate for focus mismatches between views;
- the inter-view SKIP mode using depth information [142], replacing the global disparity vector by regional disparity vectors [52], or using the SKIP mode with residual prediction [59];
- the DIRECT mode extended in inter-view to infer temporal motion information from the corresponding blocks in neighboring views;
- the view synthesis or view interpolation prediction to generate synthesized views from neighboring views using estimated depth, then use synthesized view for prediction;

All these tools would require changes to slice/MB level AVC syntax. Additional 10-15% gains are expected depending on the content. It is still not clear if this amount of gains will be enough to make worthwhile to include these tools in the current or future 3D video standards.

In the remainder of the thesis, we describe in details the contributions along the following points.

- We start at the end of the 3D video communication process, the rendering part, where we give an attempt to enhance the quality of the novel synthesized view.
- Afterwards, we consider the impact of the compression on the rendering part. By knowing that the depth video is an important key information for the rendering process, we study the influence of the depth video compression through a wavelet transform.
- Then, by considering that the two video sequences (texture and depth) are correlated, we give an attempt to exploit this correlation in the coding process through a global motion field estimation, and a joint bit allocation strategy.
- Finally, in the case where no depth video is transmitted, we introduce a novel motion/disparity framework based on a dense estimation to increase the coding efficiency.

Part II

Implementation of an advanced 3D video codec

Chapter 3

Hole-filling for novel view synthesis

Contents

Recent researches give much attention to 3DTV, more specifically to the DIBR approaches, also called 3D image warping in the computer graphics literature [48]. DIBR technique has been recognized as a promising tool which can synthesize some new "virtual" views from the so-called video-plus-depth data representation (see Section 1.2.2). The most important problem in the generation of the "virtual" view is to deal with the newly exposed areas, appearing as *holes* and denoted as *disocclusions*, which may be revealed in each warped image.

In this chapter, we address the problem of restoring the disoccluded regions in the warped image with an original video-plus-depth sequence received at the client side (*i.e.* no quantification error due to the encoded/transmitted work flow). We investigate two camera configurations with a small and large baseline, *i.e.* small and large distance between the cameras.

This chapter starts by introducing the general formulation of the 3D image warping view synthesis equation based on DIBR. Afterwards, we address the special case of a stereoscopic rendering set-up. We finally discuss about the disocclusion problem involved by the warping process.

In a second part, in order to deal with the disoccluded regions provoked by a small baseline, we design a pre-filtering framework based on the contours of the associated depth map. More specifically, we propose a solution based on a weighted Gaussian filter taking into account the distance to sharp changes near object boundaries in the depth map.

In a third part, we address the problem of the recovery of larger disoccluded regions. To this end, we propose an inpainting-based post-processing of the warped image. Specifically, in the texture and structure propagation process we propose taking into account the depth information by distinguishing foreground and background parts of the image.

3.1 Video view synthesis using 3D image warping

In this section we describe the 3D image warping technique, one of the DIBR techniques [77, 74, 94], which enables the generation of the "virtual" view (as illustrated in Fig 3.1). This includes a function for mapping points from the reference image plane to the targeted image plane.

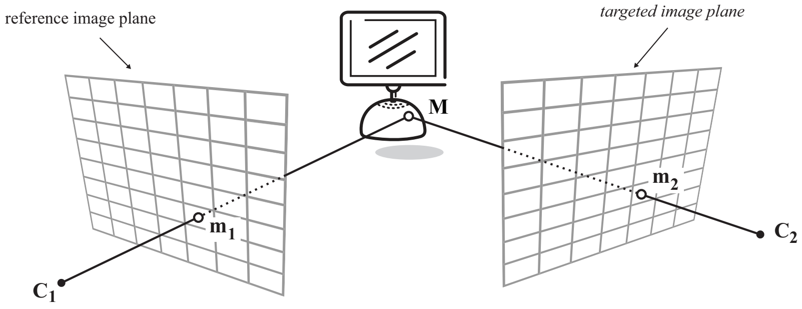


Figure 3.1: 3D image warping: projection of a 3D point on two image planes in homogeneous coordinates.

3.1.1 General formulation

First we introduce some notations: the intensity of the reference view image I_1 at pixel coordinates (u_1, v_1) is denoted by $I_1(u_1, v_1)$. In order to synthesize the second view $I_2(u_2, v_2)$ with the depth data $Z(u_1, v_1)$, the so-called pinhole camera model is used (see Section 1.1.2).

Conceptually, the 3D image warping process could be decomposed into two steps including a first back-projection of the reference image into the 3D-world, followed by projecting the back-projected 3D scene onto the targeted image plane [77].

Considering the pixel location (u_1, v_1), firstly a back-projection per-pixel is performed from the 2D reference camera image plane I_1 at to the 3D-world coordinates. Then, a second projection from the 3D-world to the image plane I_2 of the targeted "virtual" camera at pixel location (u_2, v_2), and so on for each pixel location. To perform such operations, three quantities are needed: $\boldsymbol{K}_1$, $\boldsymbol{R}_1$ and $\mathbf{t}_1$ which denote respectively the 3×3 intrinsic matrix, the 3×3 orthogonal rotation matrix and the 3×1 translation vector of the reference view I_1.

Thus, the 3D-world back-projected point $\mathbf{M} = (x, y, z)^\top$ is expressed in non-homogeneous coordinates as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{R}_1^{-1} K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 - \boldsymbol{R}_1^{-1}\mathbf{t}_1 \tag{3.1}$$

where λ_1 is a positive scaling factor.

Considering the targeted camera quantities, $\boldsymbol{K}_2$, $\boldsymbol{R}_2$ and $\mathbf{t}_2$, the back-projected 3D-world point $\mathbf{M} = (x, y, z, 1)^\top$ is then mapped into the targeted 2D-image coordinates $(u', v', 1)^T$ in homogeneous coordinates as follows:

$$\begin{pmatrix} u'_2 \\ v'_2 \\ w'_2 \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \boldsymbol{K}_2 \mathbf{t}_2 \tag{3.2}$$

We can therefore express the targeted coordinates function of the reference coordinates by:

$$\begin{pmatrix} u'_2 \\ v'_2 \\ w'_2 \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{R}_1^{-1} \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 - \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{R}_1^{-1} \mathbf{t}_1 + \boldsymbol{K}_2 \mathbf{t}_2 \tag{3.3}$$

It is common to attach the world coordinates system to the first camera system, such that $\boldsymbol{R}_1 = \boldsymbol{I}_3$ and $\mathbf{t}_1 = \mathbf{0}_3$, which simplifies the Eq. (3.3) into:

$$\begin{pmatrix} u'_2 \\ v'_2 \\ w'_2 \end{pmatrix} = \boldsymbol{K}_2 \boldsymbol{R}_2 \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \lambda_1 + \boldsymbol{K}_2 \mathbf{t}_2 \tag{3.4}$$

where $(u'_2, v'_2, w'_2)^\top$ is the homogeneous coordinates of the 2D-image point $\mathbf{m}_2$, and the positive scaling factor λ_1 is equal to:

$$\lambda_1 = \frac{z}{c} \quad \text{where} \quad \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \boldsymbol{K}_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{3.5}$$

Finally, the homogeneous result is converted into pixel location as $(u_2, v_2) = (u'_2/w'_2, v'_2/w'_2)$.

Notice that z is the third component of the 3D-world point $\mathbf{M}$, indicating the depth information at the pixel location (u_1, v_1) of the image I_1. This data may be considered as a key side information for retrieving the corresponding pixel location on the other image I_2.

3.1.2 Rectified cameras set-up

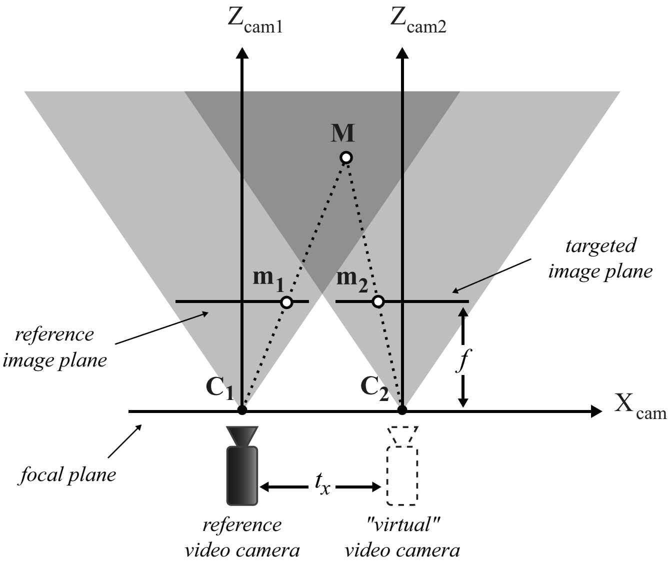


Figure 3.2: Shift-sensor camera setup

Let us consider here and in the next section, a stereoscopic set-up with identical parallel cameras with known camera parameters (internal and external). As a result:

- if the cameras are identical, then the cameras share the same intrinsic parameters with $\boldsymbol{K} = \boldsymbol{K}_1 = \boldsymbol{K}_2$,
- if the cameras are rectified (*i.e.* parallel), then there is no rotation between the cameras, which is equivalent to write that $\boldsymbol{R}_2 = \boldsymbol{I}_3$, and that there is only a horizontal translation along the horizontal axis according to the translation vector $\mathbf{t} = (t_x, 0, 0)^\top$.

Eq. (3.4) can therefore be updated as:

$$\begin{pmatrix} u'_2 \\ v'_2 \\ w'_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} z + \boldsymbol{K} \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}. \tag{3.6}$$

Note that the scaling factor λ_1 can be specified in this particular case by $\lambda_1 = z$. After solving, we obtain in non-homogeneous coordinates that:

$$u_2 = u_1 + \frac{f \times t_x}{z} \quad \text{and} \quad v_2 = v_1 \tag{3.7}$$

where $(u_2, v_2) = (u'_2/w'_2, v'_2/w'_2)$, t_x being the horizontal camera translation , and f being the focal length of the reference camera.

The pixel displacement $\Delta u = u_2 - u_1$ between the views is denoted as *disparity*. As defined in Eq. (3.7), the disparity value is inversely proportional to the depth data in the case of a rectified cameras set-up.

3.1.3 The disocclusion problem

An inherent problem of the described 3D image warping algorithm is due to the fact that each pixel does not necessarily exist in both views. Consequently, due to the sharp discontinuities in the depth data (*i.e.* strong edges), the 3D image warping can expose areas of the scene that are occluded in the reference view and become visible in the second view. In Fig. 3.3, we can see the resulting warped picture from the 3D image warping process according to the camera set-up previously described. The disoccluded regions are colored in magenta.

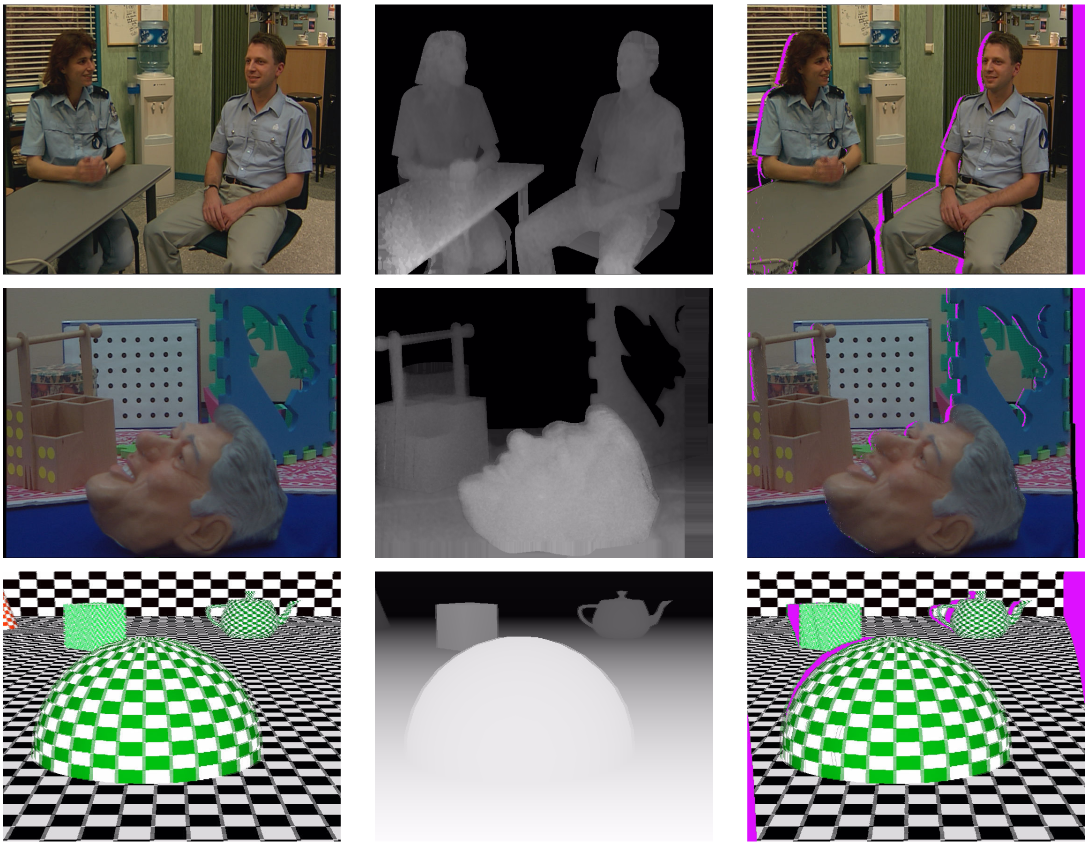

(a) reference image (b) depth map (c) warped image

Figure 3.3: In magenta: newly exposed areas in the warped picture (from the ATTEST test sequences (up)"Interview", (middle)"Orbi" and (bottom)"Cg").

To deal with these disocclusions, one solution would be to rely on more complex multi-dimensional data representation like layered depth image (LDI) data representation [104, 58] that allows to store additional depth and color values for pixels that are occluded in the original view. This extra data provides the necessary information that is needed to fill disoccluded areas in the rendered, novel views. However, this means increasing the overhead complexity of the system. On the other hand, disocclusions removal can be achieved by pre-processing the depth video in order to reduce the depth data discontinuities in a way that disocclusions decrease, followed by a post-processing of the warped image to replace the missing areas with some color informations.

Pre-Processing

Pre-processing the depth video allows to reduce the number and the size of the disoccluded regions.

General smoothing Tam *et al.* proposed in [115] to smooth overall the depth video to first minimize the effects of noise and distortions in the depth map, and also reduce the disoccluded areas. Usually some well known smooth filters are utilized, like the average filter, Gaussian filter, or median filter, which act like low frequency filters. Their effect is to remove sharp discontinuities from the depth data, thus reducing artifacts at object boundaries after the 3D image warping process. The most usually used filter for the smoothing process is the Gaussian filter and experiments in [115] have shown that the perceived stereo image quality is improved by increasing the smoothing strength of the depth map.

Asymmetric filtering Smoothing the depth video is a good way to reduce the sharp transitions, but using symmetric filters introduces geometric distortions, *i.e.*, some vertically boundaries become curved. To reduce or even remove this phenomenon, Zhang and Tam recommended in [141] to use asymmetric filters, consisting in having a stronger smoothing in the vertical direction.

Edge dependent filtering Considering the fact that smoothing the whole depth video before 3D image warping damages more than simply applying a correction around the edges, Chen *et al.* proposed in [17] a non-linear edge filter. Thus, holes identified as big may become smaller, which simplifies the interpolation for filling the holes.

Adaptive smoothing Within DIBR over Terrestrial Digital Multimedia Broadcasting (T-DMB), Jung *et al.* presented in [60] an adaptive pre-processing of the depth map based on gradient direction and spatial gradient. The two adaptive pre-processing algorithms share a common framework which iteratively convolve the depth map with an averaging mask whose coefficients reflect, at each point, the specific measurement of the depth map.

Bilateral filtering More recently, a bilateral filtering [123] was used to enhance the depth video, as proposed by Cheng *et al.* [23], Gangwal *et al.* [43] and Mori *et al.* [85]. Compared with conventional averaging or Gaussian filter, the bilateral filter preserves the sharp depth changes in conjunction with the intensity variation in color space, which results in consistent boundaries between texture and depth images.

Post-processing

Nevertheless, after depth pre-processing, some disocclusions may still remain, which require a next stage, consisting in interpolating the missing values. The process of filling in the disocclusions is also known as *hole-filling*.

Average filtering For hole-filling, the average filter has been commonly used [141]. However, the average filter does not preserve edge information of the interpolated area. Therefore, using average filter results in obvious artifacts in highly-textured areas.

Multiple reference images Mark *et al.* [75] and Zhan-wei *et al.* [140] proposed merging two warped images provided by two different spatial or temporal viewpoints, where the pixels in reference images are processed in an occlusion-compatible order.

Inpainting In a review of inpainting techniques in [119], Tauber *et al.* argue that inpainting can provide an attractive and efficient framework to fill the disocclusions with texture and structure propagation, and should be integrated with image based rendering (IBR).

Conclusion

Although these methods are effective in some respect, there still exist problems such as the degradation of non-disoccluded areas in the depth map, the depth-induced distortion of warped images and undesirable artifacts in inpainted disoccluded regions. The depth map is pre-processed while only some of the sharp discontinuities located at the edges of objects need to be treated. In the following we propose an adaptive depth map pre-processing operating on the edges and taking into account the distance to the edges. This step can be followed by an inpainting-based post-processing using depth information in case of a large baseline distance.

3.2 Pre-processing of the depth video

In this section, we address the problem of filling in the disocclusions within a stereoscopic camera framework, with a small camera inter-distance (around to the human eyes inter-distance).

As previously discussed, one way to deal with the disocclusion problem is to pre-process the depth video, for example by smoothing, commonly operated with a Gaussian filter. Instead of smoothing the whole depth video, we propose here an adaptive filter taking into account the distance to the edges. The proposed scheme is summarized in Fig. 3.4. First we apply a preliminary pre-processing stage to extract the edges of the depth map capable of revealing disoccluded areas, that we will refer to in the following as Contours of Interest (CI). This spatial information permits then to compute the distance data, and also to compute the weight information for the proposed filtering operation.

3.2.1 Extraction of regions around the Contours of Interest (CI)

In Fig. 3.3, we can see the resulting warped picture from the 3D image warping process according to the camera set-up described in Section 3.1.2. The 3D image warping has exposed areas of the scene for which the reference camera has no information (here colored in magenta). These areas are precisely located around the CI of objects and we can identify the location of these regions before the 3D image warping by applying the following pre-processing.

The CI are generated from the depth map by applying a directional edge detector, such that only one edge side is detected, as illustrated in Fig. 3.4. To handle the problem

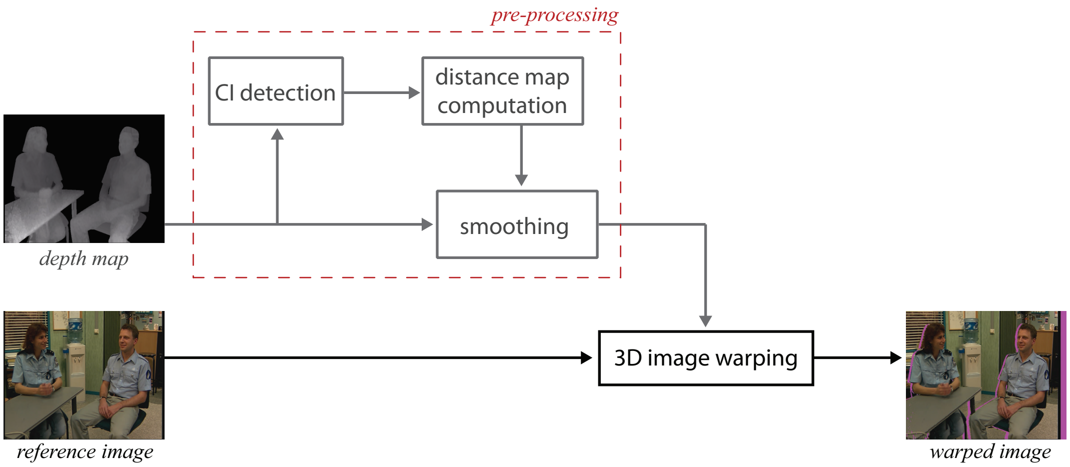
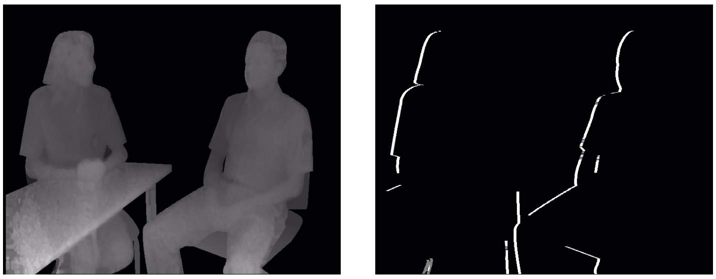


Figure 3.4: Pre-processing of the depth map before the 3D image warping.



(a) depth map (b) CI of the depth map

Figure 3.5: Examples of CI of depth map (from the 1st frame of the ATTEST test sequences "Interview").

of choosing an appropriate threshold, we use an approach by *hysteresis*[1], where multiple thresholds are used to find an edge. The resulting binary map reveals on the depth map areas where displacement is high, and thus, where it is necessary to apply a strong smoothing, leading to a reduction or even an elimination of the disoccluded areas in the targeted view.

3.2.2 Distance map

Discrete distance map computing is commonly used in shape analysis to generate skeletons of objects [44]. Here, we propose utilizing the distance map computation to calculate the shortest distance from a point to a CI. Moreover, we use the distance information as a weight for the filter adaptation.

In a distance map context, a zero value indicates that the pixel belongs to the CI. Subsequently, non-zero values represent the shortest distances from a point to the CI. Among all the possible distances in a discrete space we only consider here the city-block

[1] Hysteresis is used to track the more relevant pixels along the contours. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. If the magnitude is between the two thresholds, then it is set to zero unless the pixel is located near a edge detected by the high threshold.

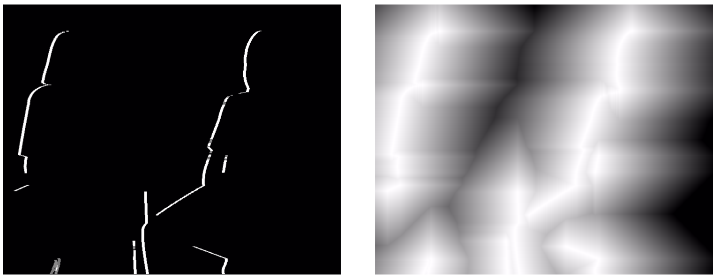
(or 4-neighbors) distance $d(.,.)$, a special case of the Minkowski distance [2]. The city-block distance is defined for two pixels $p_1(u_1,v_1)$ and $p_2(u_2,v_2)$ by:

$$d(p_1,p_2) = |u_1 - u_2| - |v_1 - v_2|\,, \quad \text{where} \quad p_1,p_2 \in \mathbb{Z}^2 \tag{3.8}$$

We define the distance map D of the depth map Z, with respect to the given CI, by the following function:

$$D(u,v) = \min_{p\in\mathrm{CI}} \{d(Z(u,v),p)\} \tag{3.9}$$

It is possible to take into account the spatial propagation of the distance, and compute it successively from neighboring pixels with a reasonable computing time, with an average complexity linear in the number of pixels. The propagation of distance relying on the assumption that it is possible to deduce the distance of a pixel from the value of its neighbors, is well suited for sequential and parallel algorithms. One example of distance map is shown in Fig. 3.6.



(a) CI of the depth map (b) distance map

Figure 3.6: Examples of distance map derived from the CI (from the 1st frame of the ATTEST test sequences “Interview”).

3.2.3 Edge-distance dependent pre-filtering

With the per-pixel distance information obtained in the previous subsection it becomes feasible to compute an adaptive Gaussian filter with a stronger smoothness near an edge and a lower smoothness far from an edge. The new depth value in the depth map $\widetilde{Z}$ at the pixel location (u,v) is then defined by:

$$\widetilde{Z}(u,v) = \alpha(u,v)\cdot Z(u,v) + \Big(1-\alpha(u,v)\Big)\cdot (Z * g_{2\mathrm{D}})(u,v) \tag{3.10}$$

where $u,v \in \mathbb{Z}$, and with

$$\alpha(u,v) = \begin{cases} \frac{D(u,v)}{D_{\max}}, & \text{if} \quad D(u,v) < D_{\max} \\ 1, & \text{otherwise} \end{cases} \tag{3.11}$$

where $\alpha \in [0,1]$, normalized by the maximum distance D_{max}, controls the smoothing impact on the depth map by means of the distance map $\mathcal{D}$. The quality of the depth map is thus preserved for the regions far from the object boundaries.

[2] For two 2D points (u_1,v_1) and (u_2,v_2) the Minkowski distance of order k is defined as: $\left(|u_1-u_2|^k - |v_1-v_2|^k\right)^{1/k}$.

Note that, over all the discrete image support, the case $\alpha = 1$ corresponds to not filtering the depth map, whereas the case $\alpha = 0$ refers to applying a Gaussian filtering of equal strength on the whole depth map.

The 2D discrete Gaussian convolution is defined by:

$$(Z * g_{2\mathrm{D}})(u, v) = \sum_{x=\frac{-w}{2}}^{\frac{w}{2}} \sum_{y=\frac{-h}{2}}^{\frac{h}{2}} Z(u - x, v - y) \cdot g_{2\mathrm{D}}(x, y) \tag{3.12}$$

where the two-dimensional approximation of the discrete Gaussian function $g_{2\mathrm{D},\sigma}$ is separable into x and y components, and expressed as follows:

$$\begin{aligned} g_{2\mathrm{D}}(x, y) &= g_{1\mathrm{D}}(x) \cdot g_{1\mathrm{D}}(y) \\ &= \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{x^2}{2\sigma_x^2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \end{aligned} \tag{3.13}$$

where $g_{1\mathrm{D}}$ is the one-dimensional discrete approximation of the Gaussian function, the parameters w and h being respectively the width and the height of the convolution kernel, σ_x and σ_y being the standard deviation of the Gaussian distributions respectively along the direction x and y.

In the case of a symmetric Gaussian distribution, we update Eq. (3.13) to:

$$g_{2\mathrm{D}}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3.14}$$

where $\sigma = \sigma_x = \sigma_y$.

We shall see in the experimental results section that the asymmetric nature of the distribution may help to reduces the geometrical distortion in the warped pictured induced by the pre-filtering of the depth map.

3.2.4 Experimental results

For the experiments, we have considered the three ATTEST video-plus-depth test sequences "Interview", "Orbi" and "Cg" (720×576, 25fps) [38]. The experimental parameters for the camera are t_x=60 mm for the horizontal camera translation and f=200 mm for the focal length, and concerning the smoothing process the parameters σ and $D_{\max}$ have been chosen respectively 20 and 50. When experimenting the asymmetric pre-filtering, the level of vertical smoothing equals to three times that in the horizontal direction, such that $\sigma_x = 20$ and $\sigma_y = 64$.

In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice the discrete convolution depends on the kernel width w and σ. It is common to choose $w = 3\sigma$ or $w = 5\sigma$. In our experiments we let w equal to 3σ.

We start by comparing our solution with the classical "all blur" solution consisting in applying the Gaussian filter on the whole image. The conditions of the experiments are done in a symmetric and asymmetric fashion way.

Symmetric filtering

We can see in Fig. 3.7 examples of resulting pre-processing of the depth map through the means of a Gaussian filtering and the proposed framework. While a Gaussian filtering smooths all the depth map uniformly, our proposed approach focuses on the areas susceptible of being revealed in the warped image. As a consequence, less depth-filtering-induced distortions are introduced in the warped picture Fig. 3.8, and in the meantime the disoccluded regions are removed in the warped image, as we can see in Fig. 3.9.

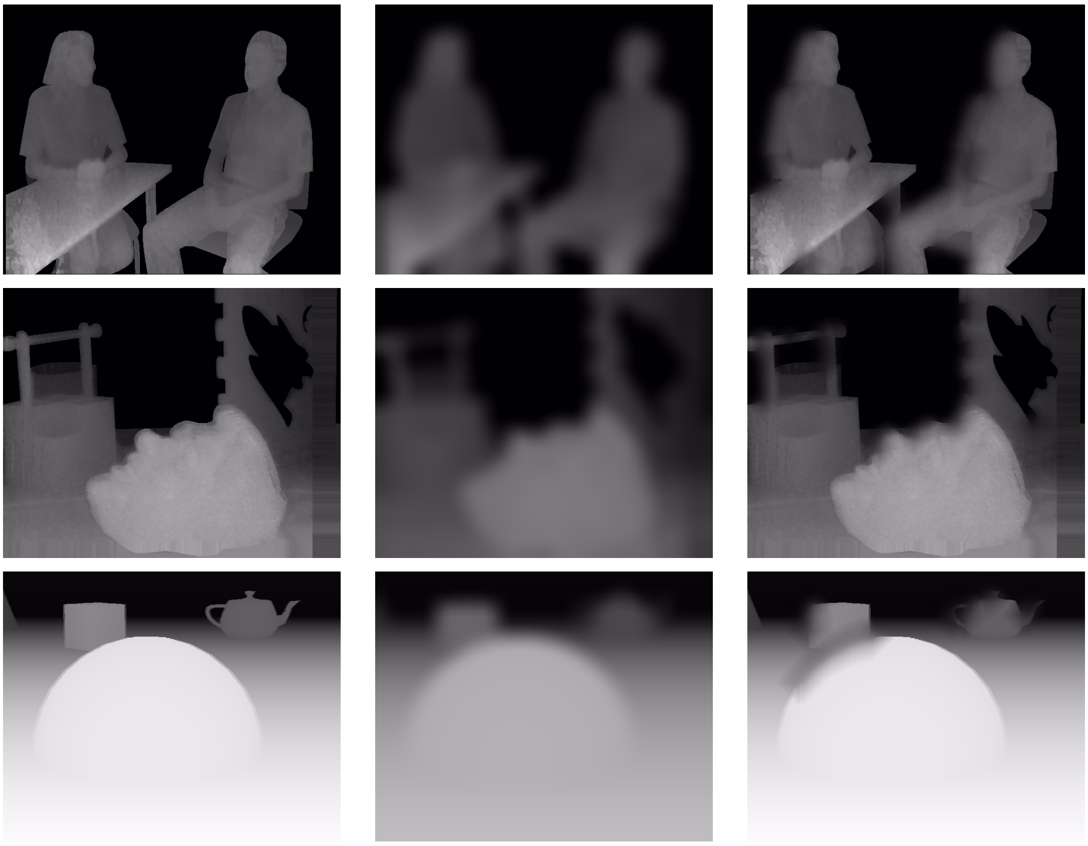

(a) original depth map (b) Gaussian filtering (c) proposed pre-filtering

Figure 3.7: Examples of symmetric pre-filtering depth map with the proposed framework (from the ATTEST test sequences (top)“Interview”, (middle)“Orbi” and (bottom)“Cg”).

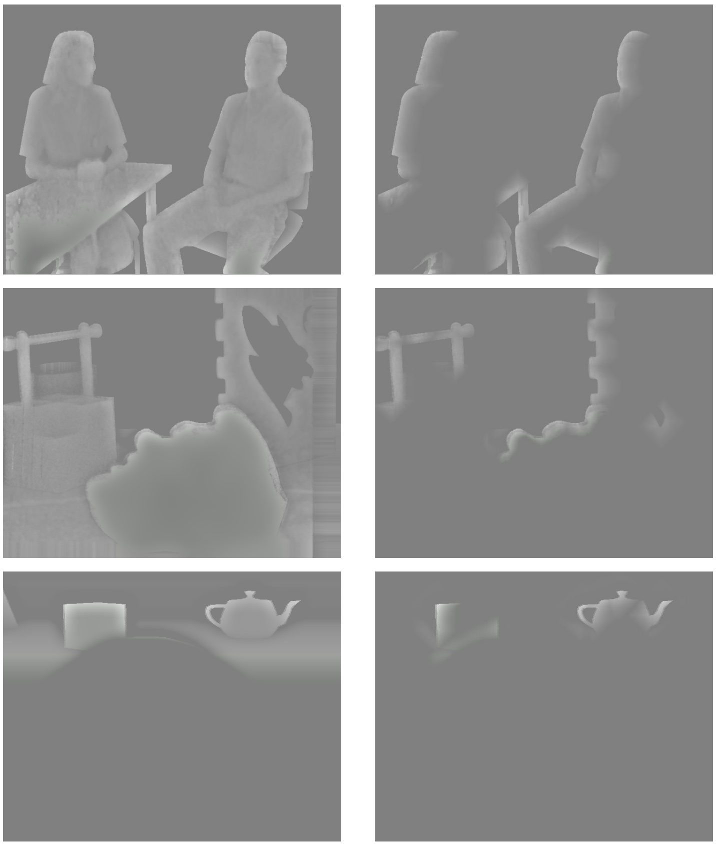
Asymmetric filtering

However, the depth-filtering-induced distortion may provoke geometric distortion in the warped picture, where vertical line bents (as shown in Fig 3.9 around the head of the cop). To overcome this issue, we investigate, as proposed by Zhang *et al.* [141], an adaptive asymmetric filtering of the depth map. As we can see in Fig. 3.11, the asymmetric nature of the filter tends to reduce the amount of geometric distortion that might be perceived, and straightens the vertical lines.

PSNR comparison

Let us consider the original depth map Z and the pre-processed depth map $\widetilde{Z}$, and also, the warped pictures I_{virt} and $\widetilde{I}_{\text{virt}}$ respectively according to Z and $\widetilde{Z}$.

In order to measure the filtering-induced distortion in the depth map, and in the warped pictures, we defined two objective PSNR measurements before and after the 3D



(a) Gaussian filtering (b) proposed pre-filtering

Figure 3.8: Error comparison between the original depth map of the pre-processed depth map (from the ATTEST test sequences (top)“Interview”, (middle)“Orbi” and (bottom)“Cg”).

image warping (see Fig. 3.12) as follows:

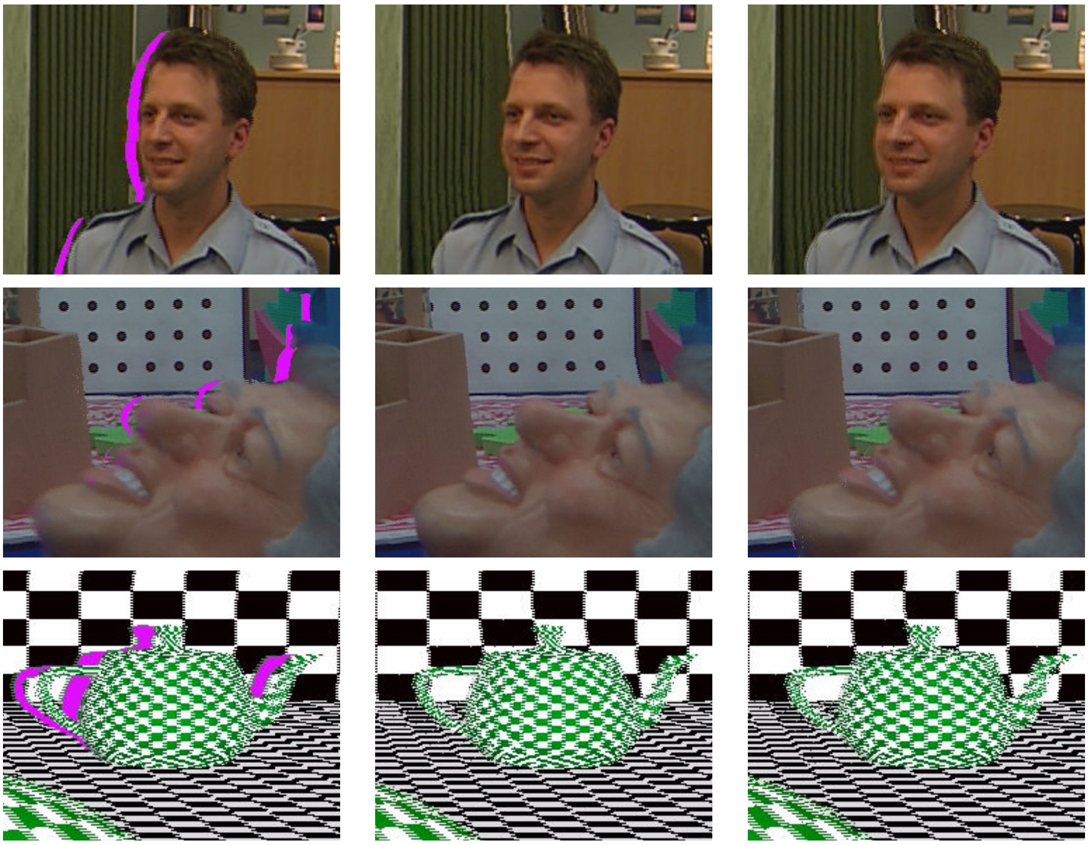
$$\text{PSNR}_{\text{depth}} = \text{PSNR}(Z, \widetilde{Z}) \quad \text{and} \quad \text{PSNR}_{\text{virt}} = \text{PSNR}(I_{\text{virt}}, \widetilde{I}_{\text{virt}})_{\mathcal{D}\backslash\mathcal{O}\cup\widetilde{\mathcal{O}}} \tag{3.15}$$

where $\mathcal{D} \in \mathbb{N}^2$ is the discrete image support, $\mathcal{O}$ and $\widetilde{\mathcal{O}}$ being the occlusion image support of the 3D image warping using respectively Z and $\widetilde{Z}$.

$\text{PSNR}_{\text{depth}}$ is calculated between the original depth map and the filtered one. Hence, $\text{PSNR}_{\text{depth}}$ only considers coding artifacts in the depth map. However, it does not reflect the overall quality of the warped image. Then, $\text{PSNR}_{\text{virt}}$ is calculated between the warped image mapped with the decoded depth map and the original depth map. In order not to introduce in the $\text{PSNR}_{\text{virt}}$ measurement the warping-induced distortion, $\text{PSNR}_{\text{virt}}$ is computed only on the non-disoccluded areas $\mathcal{D}\backslash\mathcal{O} \cup \widetilde{\mathcal{O}}$.

We can observe on the Fig. 3.13 the important quality improvement obtained with our method. In a subjective way, we also remark less degradation on the reconstructed images due to the fact that our method preserves more details in the depth map.

The main gain comes from the conservation of the non-disoccluded regions. Indeed, by introducing the concept of CI, we attempt to limit any unnecessary filtering-induced distortion, as shown in Fig. 3.14. Thus, the right side is preserved and the vertical line does not bent in our framework.



(a) no pre-processing (b) Gaussian filtering (c) proposed pre-filtering

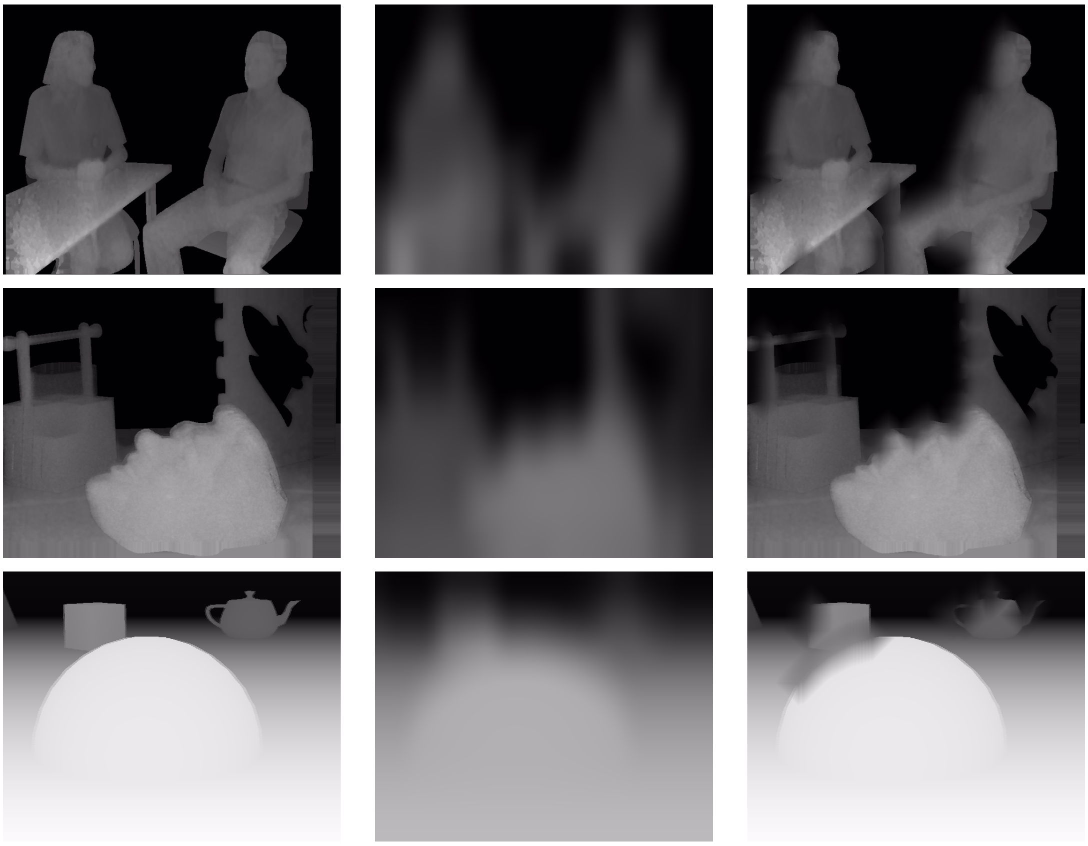
Figure 3.9: Warped images issue from the 3D image warping using the different symmetric pre-processed depth map (from the ATTEST test sequences (top)“Interview”, (middle)“Orbi” and (bottom)“Cg”).

3.2.5 Conclusion

In this section, we have introduced a new adaptive filter for 3D image warping, taking into account the distance to object boundaries. The main advantage consists in limiting any unnecessary filtering-distortion in the depth map. Experiment results have illustrated the high efficiency of the proposed method. Of course, any smoothing filter can be used instead of the used Gaussian one.

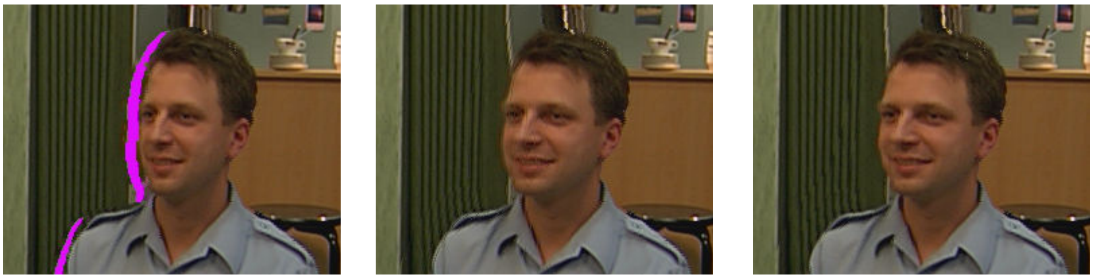
To deal with the geometric distortion, we applied an asymmetric filter, which is possible since the Gaussian filter is separable. An improvement could be expected by applying according to the direction of the gradient in each part of the image, an adaptive asymmetric filtering to prevent the vertical and horizontal lines from bending.

The depth pre-processing approach is particularly efficient when the baseline is relatively small, for example in the case of a stereoscopic rendering using a baseline equivalent to the average human inter-eye distance. When the baseline becomes larger, this approach would introduce bigger geometric distortions, that will be difficult to handle by just applying an asymmetric filtering strategy. To deal with large baselines, we propose in the next section a post-processing approach on the warped image to fill in the disoccluded regions by inpainting techniques.



(a) original depth map (b) Gaussian filtering (c) proposed pre-filtering

Figure 3.10: Examples of asymmetric pre-filtering depth map with the proposed framework.



(a) no pre-processing (b) symmetric pre-filtering (c) asymmetric pre-filtering

Figure 3.11: Comparison between symmetric and asymmetric filtering with the proposed framework.

3.3 Video inpainting aided by depth information

In this section, in opposition to the previous section, we consider a large baseline camera set-up which corresponds more to FVV applications rather than stereoscopic vision. As a consequence, the disoccluded areas become larger in the warped image. One solution suggested by Tauber *et al.* in [119] consists of combining 3D image warping with inpainting techniques to deal with large disocclusions, due to the natural similarity between damaged holes in painting and disocclusions in 3D image warping.

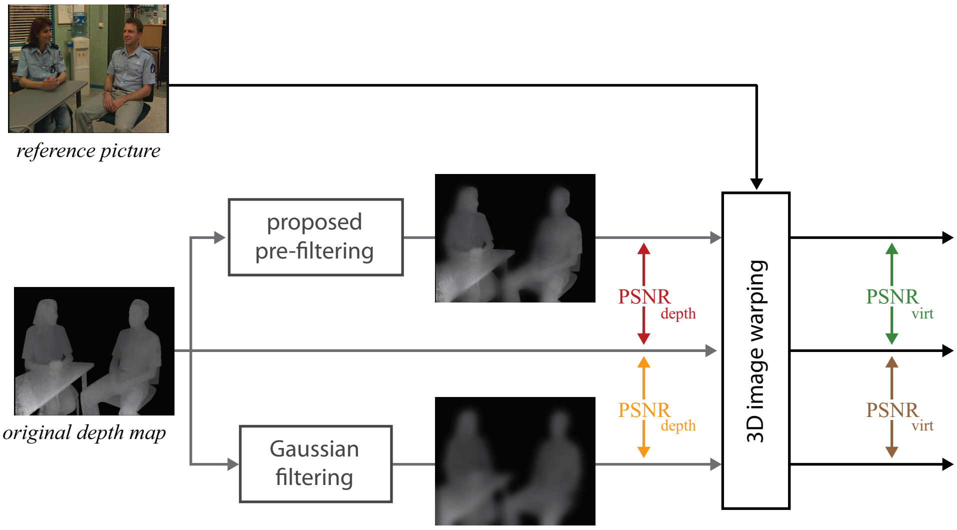


Figure 3.12: Practical disposition of the two PSNR computations

Image inpainting, also known as image completion [42], aims at filling in pixels in a large missing region of an image with the surrounding informations. Image and video inpainting serve a wide range of applications, such as removing overlaid text and logos, restoring scans of deteriorated images by removing scratches or stains, image compression, or creating artistic effects. State-of-the-art methods are broadly classified as structural inpainting or as textural inpainting.

Structural inpainting reconstructs using prior assumptions about the smoothness of structures in the missing regions and boundary conditions, while textural inpainting considers only the available data from texture exemplars or other templates in non-missing regions.

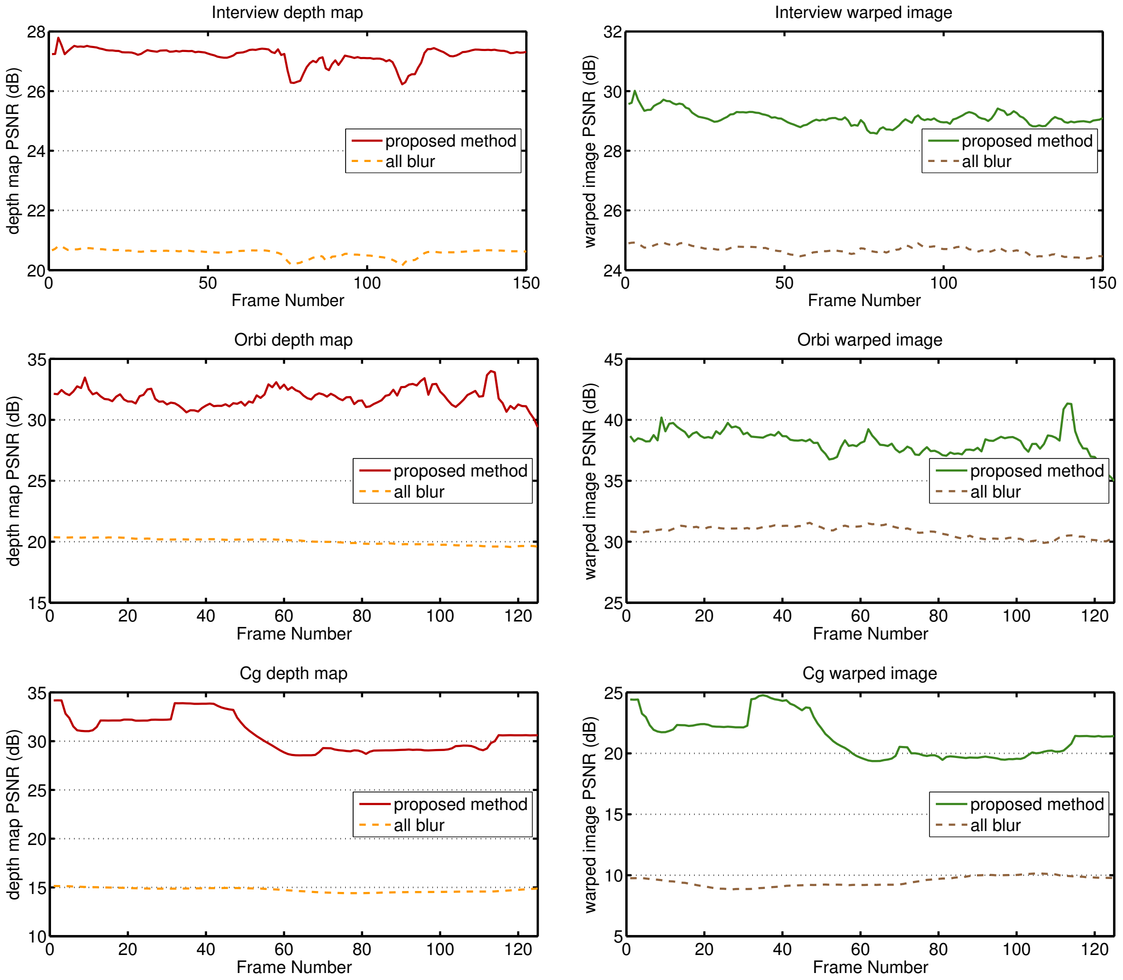
Initially introduced by Bertalmio *et al.* [12], structural inpainting uses either isotropic diffusion or more complex anisotropic diffusion to propagate boundary data in the isophote[3] direction, and prior assumptions about the smoothness of structures in the missing regions. Textural inpainting considers a statistical or template knowledge of patterns inside the missing regions, commonly modeled by Markov random fields (MRF). Thus, Levin *et al.* suggest in [67] to extract some relevant statistics about the known part of the image, and combine them in an a MRF framework.

In this work, we start from the Criminisi *et al.* work [30], in which they attempted to combine structure into textural inpainting advantages by using a very insightful principle, whereby the texture is inpainted in the isophote direction according to its strength. We propose extending this idea by adding depth information to distinguish pixels belonging to foreground and background. Let us first briefly review the Criminisi's inpainting algorithm.

3.3.1 Criminisi's inpainting algorithm

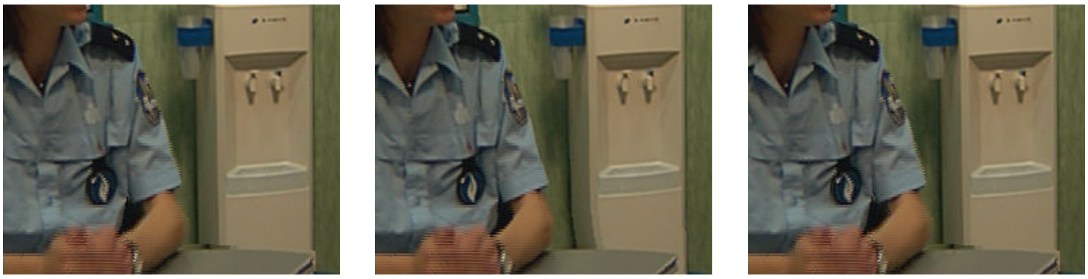
Criminisi *et al.* [30] first note that exemplar-based texture synthesis contains the essential process required to replicate both texture and structure, and use the sampling concept from Efros and Leung's approach [34]. Moreover, they demonstrate that the quality of

[3] Isophotes are level lines of equal gray-levels. Mathematically, the direction of the isophotes can be interpreted as $\nabla^{\perp} I$, where $\nabla^{\perp} = (-\partial_y, \partial_x)$ is the direction of the smallest change.



(a) Depth map (b) Warped image

Figure 3.13: PSNR comparison between the depth and warped images through a symmetric framework (from the top to bottom: the video sequence "Interview", "'Orbi" and "Cg").

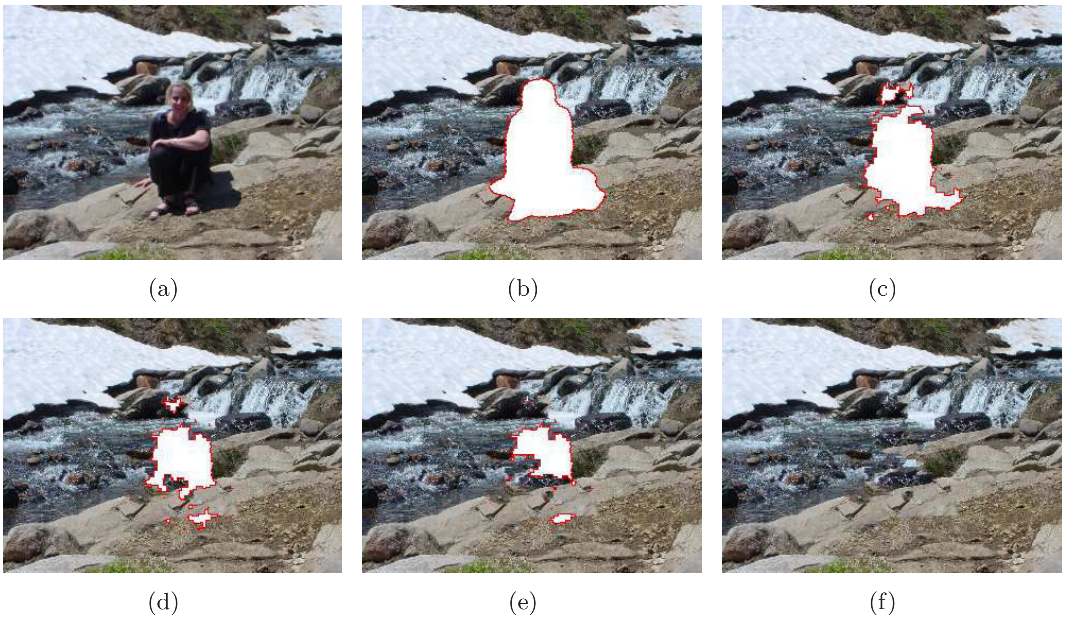


(a) no pre-processing (b) Gaussian filtering (c) proposed pre-filtering

Figure 3.14: Non-disoccluded areas of the warped images issue from the 3D image warping using the different pre-processed depth map (from the ATTEST test sequence "Interview").

the output image synthesis is highly influenced by the order in which the inpainting is processed.

Considering an input image I, and a missing region Ω, the source region Φ is defined as



(a) (b) (c)

(d) (e) (f)

Figure 3.15: Removing large objects from photographs using Criminisi's inpainting algorithm. (a) Original image, (b) the target region (10% of the total image area) has been blanked out, (c...e) intermediate stages of the filling process, (f) the target region has been completely filled and the selected object removed (from [30]).

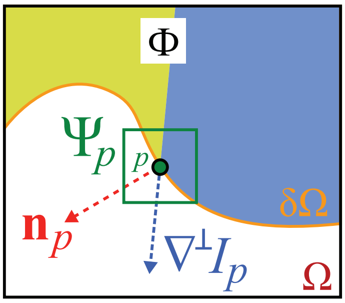


Figure 3.16: Notation diagram (from [30]).

$\Phi = I - \Omega$ (see Fig. 3.16). The algorithm performs the synthesis task through a best-first filling strategy that depends entirely on the priority values that are assigned to each patch on the boundary $\delta\Omega$. Given a patch Ψ_p centered at the point p for some $p \in \delta\Omega$ (see Fig. 3.16), they define its priority $P(p)$ as the product of two terms:

$$P(p) = C(p) \cdot D(p), \tag{3.16}$$

where $C(p)$ is called the *confidence* term that indicates the reliability of the current patch, and $D(p)$ the *data* term that gives special priority to isophote direction. They are defined as follows:

$$C(p) = \frac{1}{|\Psi_p|} \sum_{q \in \Psi_p \cap \Phi} C(q) \quad \text{and} \quad D(p) = \frac{\langle \nabla^{\perp} I_p, \mathbf{n}_p \rangle}{\alpha} \tag{3.17}$$

where $|\Psi_p|$ is the area of Ψ_p (in terms of number of pixels within the patch Ψ_p), α is a normalization factor (*e.g.* $\alpha = 255$ for a typical gray-level image), $\mathbf{n}_p$ is a unit vector

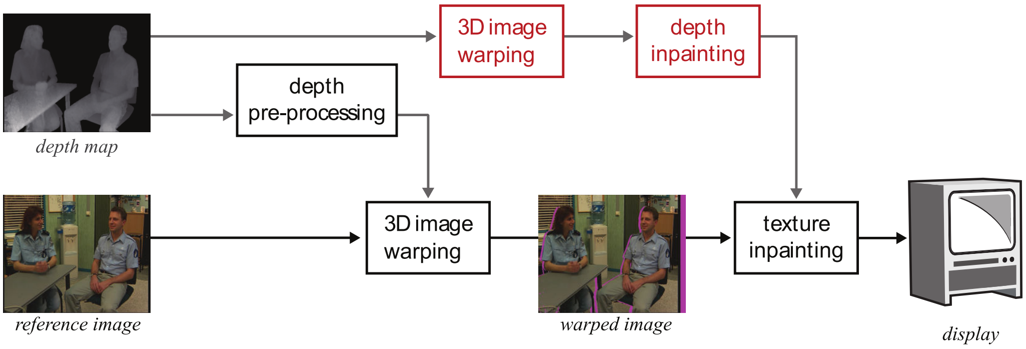


Figure 3.17: Rendering process aided by depth information.

orthogonal to $\delta\Omega$ at the point p, and $\nabla^{\perp} = (-\partial_y, \partial_x)$ is the direction of the isophote. Actually, $C(p)$ represents the percentage of non-missing pixels in the patch Ψ_p, and is set at initialization to $C(p) = 0$ for missing pixels in Ω, and $C(p) = 1$ everywhere else.

Once all priorities on $\delta\Omega$ are computed, a block matching algorithm derives the best exemplar $\Psi_{\widehat{q}}$ to fill in the missing pixels under the highest priority patch $\Psi_{\widehat{p}}$, previously selected, as follows:

$$\Psi_{\widehat{q}} = \underset{\Psi_q \in \Phi}{\arg\min} \left\{ d(\Psi_{\widehat{p}}, \Psi_q) \right\} \tag{3.18}$$

where $d(.,.)$ is the distance between two patches, defined as the Sum of Squared Differences (SSD).

Having found the source exemplar $\Psi_{\widehat{q}}$, the value of each pixel-to-be-filled $p' \in \Psi_{\widehat{p}} \cap \Omega$ is copied from its corresponding pixel in $\Psi_{\widehat{q}}$. After the patch $\Psi_{\widehat{p}}$ has been filled, the confidence term C(p) is updated as follows:

$$C(p) = C(\widehat{p}), \quad \forall p \in \Psi_{\widehat{p}} \cap \Omega. \tag{3.19}$$

3.3.2 Depth-aided texture and structure propagation

Although that missing areas in painting and disocclusions from 3D IBR are naturally similar, we have some *a priori* knowledge about disocclusions. In fact, disocclusions are the result of the displacement of a foreground object revealing some background areas. Therefore, filling in the disoccluded regions using background pixels rather than foreground ones makes more sense.

To fulfill that purpose, Cheng *et al.* developed a view synthesis framework in [23], in which the depth information constraints the search range for the texture matching, followed by a trilateral filter that utilizes the spatial and depth information to filter the texture image which has the benefit of enhance the view synthesis quality. On the other hand, Oh proposed in [93] to replace the foreground boundaries to the background ones located on the opposite side. Thus, they intentionally manipulated the disocclusion to have neighborhood only come from background. Afterwards, they applied one of the existing inpainting techniques.

Based on these works, we propose a depth-aided texture inpainting in line with Criminisi's algorithm principles, by giving higher priority to background pixels over the foreground ones.

Priority computation Given the associated depth patch Z in the "virtual" image plane, in our definition of the priority computation, we propose weighting the previous priority computation in Eq. (3.16) by adding a third multiplicative term as follows:

$$P(p) = C(p) \cdot D(p) \cdot L(p), \tag{3.20}$$

where $L(p)$ is the level regularity term, defined as the inverse variance of the depth patch Z_p:

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum\limits_{q \in Z_p \cap \Phi} \left(Z_p(q) - \overline{Z_p}\right)^2} \tag{3.21}$$

where $|Z_p|$ is the area (in terms of number of pixels) of the depth patch Z_p centered in p, $Z_p(q)$ is the pixel depth value at the pixel location q under Z_p, and $\overline{Z_p}$ the pixel mean value. Hence, we give more priority to patch overlaying at the same depth level, which naturally favors background pixels over foreground ones.

Patch matching Considering the depth information, we update the Eq. (3.18) as follows:

$$\Psi_{\widehat{q}} = \underset{\Psi_q \in \Phi}{\arg\min} \left\{ d(\Psi_{\widehat{p}}, \Psi_q) + \beta \cdot d(Z_{\widehat{p}}, Z_q) \right\} \tag{3.22}$$

where β controls the importance given to the depth distance minimization.

Depth inpainting

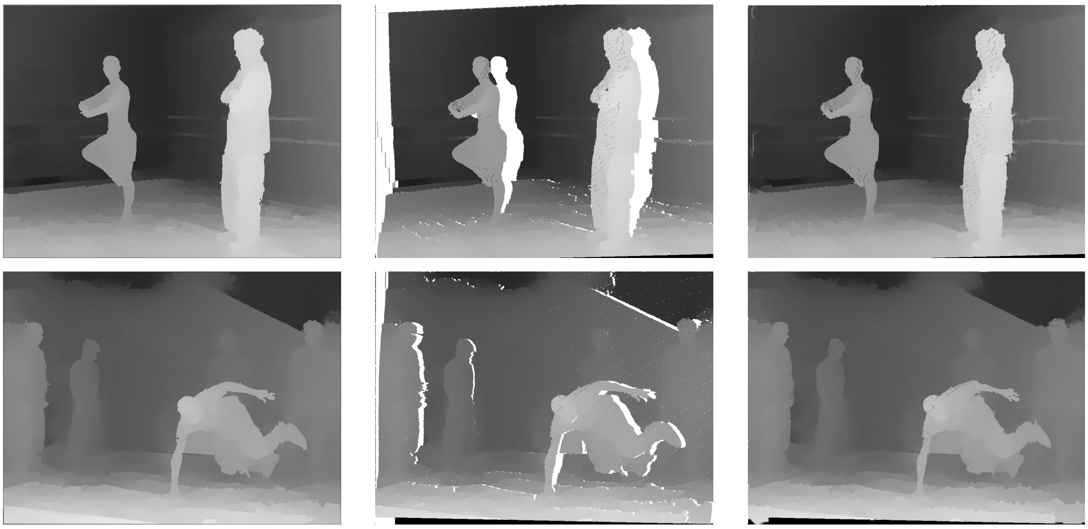
At this point, we implicitly assumed that the depth map is available at the "virtual" image plane. However, this assumption may not always hold. We handle this issue by also projecting the depth map into the "virtual" image plane, and perform hole-filling on it. Due to its smooth nature, depth disocclusions can be straightforwardly inpainted through isotropic diffusion since the assumption of the smoothness inside disoccluded regions is verified. We propose performing the state-of-the-art inpainting developed by Bertalmio *et al.* in [11] that uses the Navier-Stokes and fluid dynamics equations.

As we can see in Fig. 3.18, the texture-less nature of the depth map enables an efficient hole-filling. Thus, we perform our depth-aided texture and structure propagation described above.

3.3.3 Experimental results

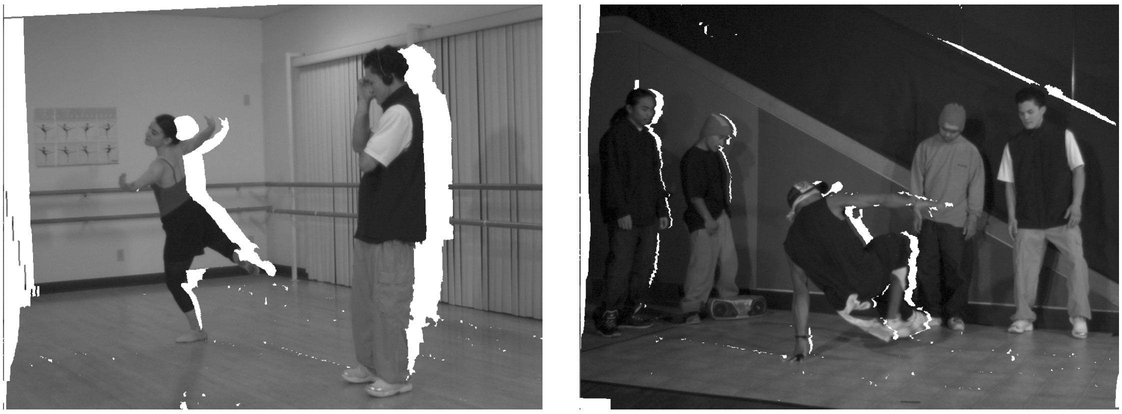
The multiview video-plus-depth (MVD) sequences "Ballet" and "Breakdancers" from the camera 5 provided by Microsoft [7] have been used to test the proposed method. The calibration parameters are supplied with the sequences. The depth video provided for each camera is estimated via a color-based segmentation algorithm [143]. The two provided depth videos have two different ranges of depth values, which directly impact on the warped video. The "Ballet" sequence represents two ballet dancers at two different depth levels, more distant than the "Breakdancers" sequence, where the protagonists are mostly at the same depth level. As a result, more disocclusions appear in the "Ballet" synthesized view.

Fig. 3.3.3 compares the results of our region-filling method with the original Criminisi's algorithm. As it can be noticed by comparing the two methods, our algorithm better preserves the contours of foreground objects, and enhances the visual quality of



(a) Original depth map (b) Projected depth map (c) Inpainted depth map

Figure 3.18: Example of inpainting of the projected depth map using Bertalmio's inpainting [11]. (top) "Ballet", (bottom) "Breakdancers".



(a) "Ballet" (b) "Breakdancers"

Figure 3.19: Disoccluded regions to fill in.

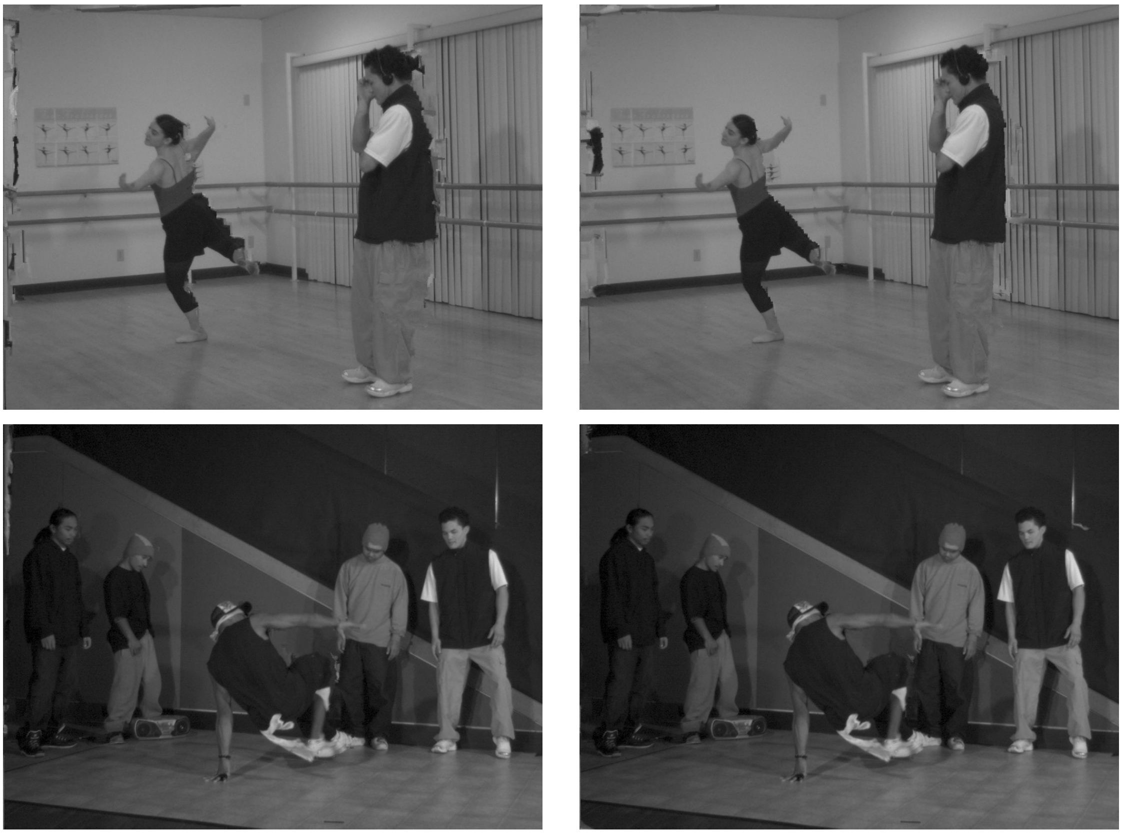
the inpainted images. This is achieved by propagating the texture and structure from background regions, while the Criminisi's algorithm makes no distinction.

In order no to introduce in the objective PSNR measurement the warping-induced distortion, and so to consider only the inpainting-induced distortion, the PSNR is computed only on the disoccluded areas.

We can observe on the Fig. 3.21 the quality improvement obtained with our method. In a subjective way, we also remark less degradation on the reconstructed images due to the fact that our method preserves more continuities in the generated view.

3.3.4 Conclusion

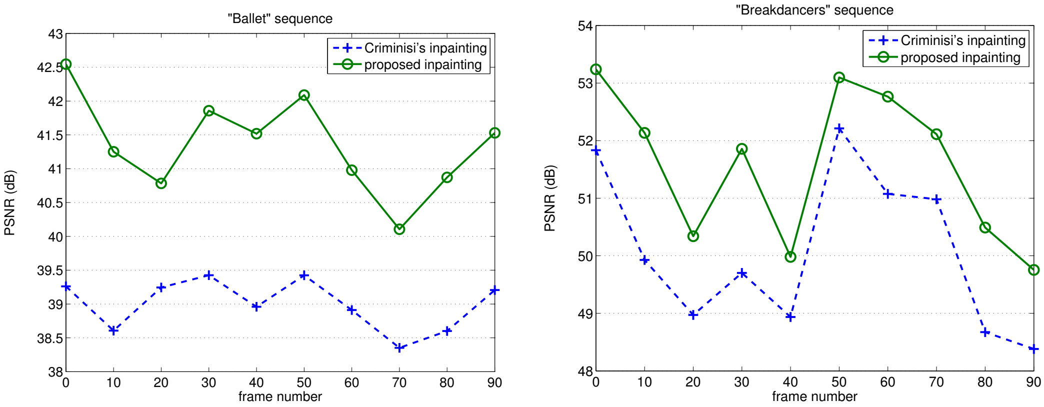
In this section, we presented a depth-aided texture inpainting algorithm for disocclusion restoration. The proposed algorithm is an extension of the Criminisi's algorithm, where the depth information is added in the priority computation and the patch matching. Thus, we



(a) Criminisi's inpainted

(b) Proposed inpainted

Figure 3.20: Example of hole-filling using the original Criminisi's algorithm and the one enhancement by warped depth information ((top row) "Ballet" sequence, Yview, (bottom row) "Breakdancers" sequence, Yview).



(a) "Ballet" sequence

(b) "Breakdancers" sequence

Figure 3.21: Objective PSNR results.

are capable to distinguish background from foreground pixels, which allows us to enhance the quality of the inpainted warped image, and the preservation of foreground contours.

3.4 Conclusion

In this chapter we addressed the problem of filling the disoccluded regions generated by the 3D image warping process. We first consider the framework with a small baseline by proposing to pre-process the depth map according the depth contours. In addition to removing distortions that might be contained in the depth video, the smoothing reduces or completely removes disoccluded regions. Moreover, the proposed strategy preserves non-disoccluded regions by weighting a Gaussian filter according to the distance to the contours, resulting in not attenuating the depth perception overall the image, while state-of-the-art approaches do.

In the case of a larger disoccluded regions, we suggested to utilize inpainting techniques well-known for their abilities to propagate texture and structure along isophotes. In addition, we proposed solving the hole-filling problem by using the depth information to distinguish foreground pixels from background ones. Therefore, we favored the propagation of background pixels which results in a better preservation of foreground contours.

Chapter 4

Wavelet-based depth coding

Contents

As discussed in the previous chapter, the depth video is a key side information in rendering “virtual” view by 3D image warping in 3DTV system or FVV applications.

In this chapter, we give more attention to the coding and transmission of the depth video, and its impact on the quality of the view synthesis.

In the first part, we investigate the impact of different wavelet filter banks for depth compression and “virtual” view synthesis with MVD data.

In the second part, we propose for depth compression an adaptive wavelet filter bank implemented by the lifting scheme, where we improve edge representation, and in the meantime we show better compression efficiency and quality of the view synthesis.

4.1 Impact of the wavelet-based depth coding on view synthesis

A first study of the impact of the depth map compression on the view synthesis has been investigated under the MPEG 3DAV AHG activities [103], in which a MPEG-4 compression scheme has been used. They conclude in applying after decoding a median filter to limit the coding-induced artifacts on the view synthesis, in a similar way of applying the deblocking filter in H.264/MPEG-4 AVC.

Afterwards, Merkle and al. proposed a comparative study in [78] between H.264 intra coding and platelet-based depth coding [87] on the quality of the view synthesis. The platelet-based depth coding algorithm models smooth regions by using piecewise-linear functions and sharp boundaries by straight lines. The results indicate that a worse depth coding PSNR does not necessarily imply a worse synthesis rendering PSNR. Indeed, platelet-based depth coding leads to the conclusion that preserving depth discontinuities in the depth compression scheme enables higher rendering quality than H.264 intra coding.

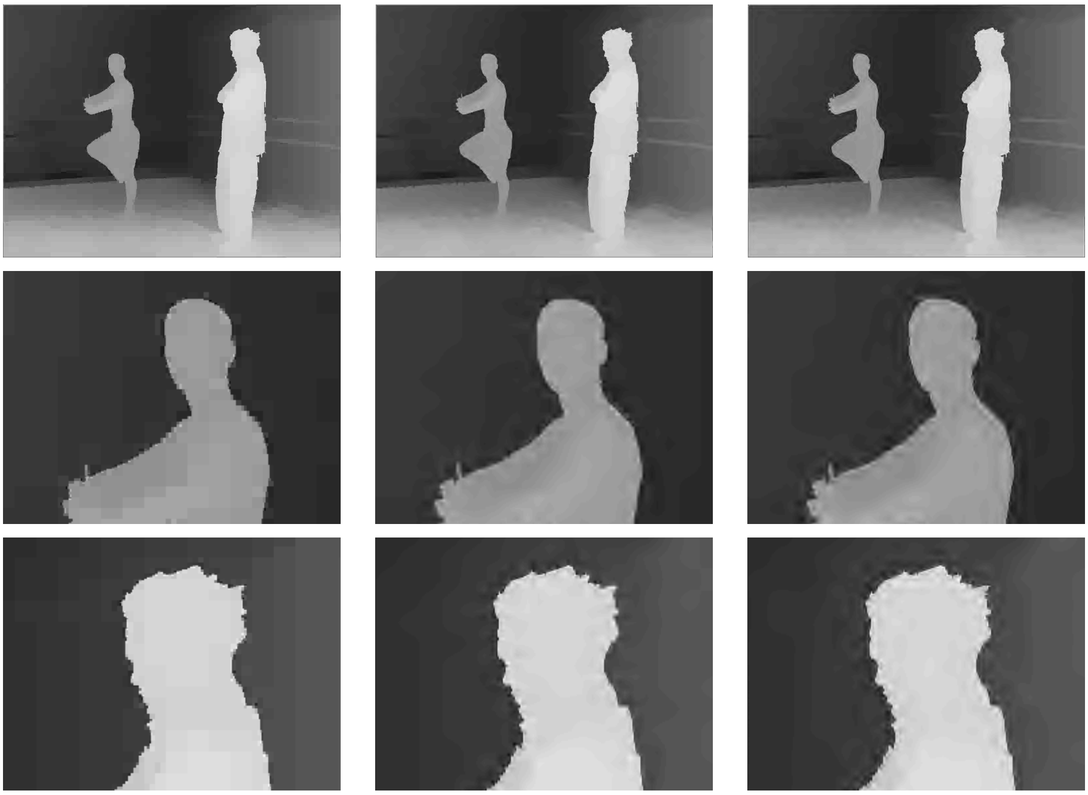
In this section, we propose studying the impact of the choice of different wavelet filter banks in the discrete wavelet transform (DWT) [73, 99] of the depth map, and the influence of geometry distortions resulting from the disturbing wavelet-based depth compression artifacts, on the quality of the “virtual” view synthesis for MVD data.

4.1.1 Wavelet-based depth coding results

In opposition to texture image, a depth map has a very singular texture-less structure, where singularities are located for the most along the edges of objects. After wavelet transform, when the depth wavelet coefficients are quantized and thresholded, one can notice on the decoded depth map, the visual apparition of artifacts, denoted as Gibbs (ringing) artifacts, along the edges as shown in Fig. 4.1 and Fig. 4.2.

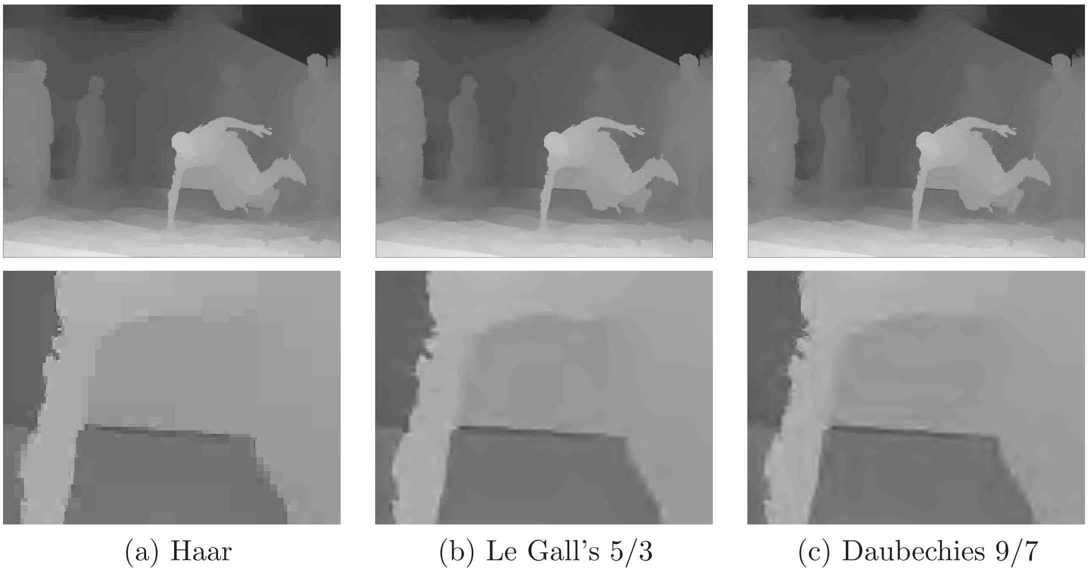
The depth coding experiments are realized with two test MVD datasets “Ballet”and “Breakdancers” (1024×768, 15fps) produced by Microsoft Research [7]. We use Haar, Le Gall’s 5/3 [65] and Daubechies 9/7 [9, 27] filter banks at a multiresolution equal to 4. These filter banks are different in particular by their filtering length. Haar being the shortest one and Daubechies 9/7 the longest one. As we can see in Fig. 4.1 and Fig. 4.2, the shortest Haar filter bank is more efficient in reducing the Gibbs (ringing) effects in the decoded depth map.

In this way, depth map edges points up the weakness of wavelet-based coding methods in efficiently preserving the structures very well localized in the space domain but with large frequency band.



(a) Haar (b) Le Gall's 5/3 (c) Daubechies 9/7

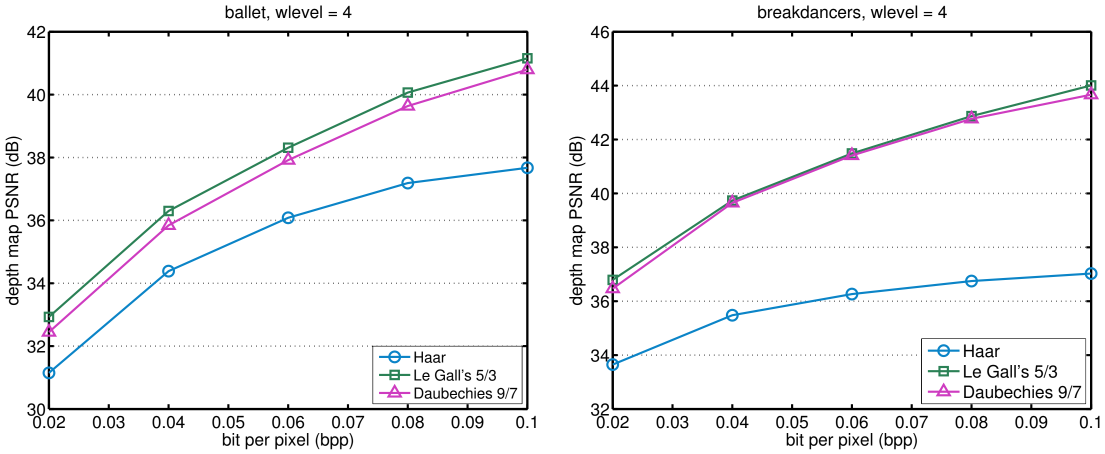
Figure 4.1: Appearance of the Gibbs (ringing) effects along "Ballet" depth contours at 0.04 bpp.



(a) Haar (b) Le Gall's 5/3 (c) Daubechies 9/7

Figure 4.2: Appearance of the Gibbs (ringing) effects along "Breakdancers" depth contours at 0.04 bpp.

However, the depth data is mainly composed of smooth areas, which favors longer filter when considering the rate-distortion (RD) performance as we can see in Fig. 4.3, at



(a) "Ballet" (b) "Breakdancers"

Figure 4.3: RD comparison of depth compression with different wavelet filter banks.

the cost of edge-localized error on the depth map. But when considering that the depth map is a 2D representation of a 3D scene surface, it become relevant to evaluate the depth compression-induced artifacts on the 3D image warping process as follows in the next section.

4.1.2 Effect on 3D image warping

The edges of the depth map are extremely relevant in revealing disoccluded areas in the warped image[1] as shown in Fig. 4.4 and Fig. 4.5, and studied in the previous chapter.

At high compression, the Gibbs phenomena become more visible in the disoccluded areas, leading to a degradation of the warped image as shown in Fig. 4.6 and in Fig. 4.7.

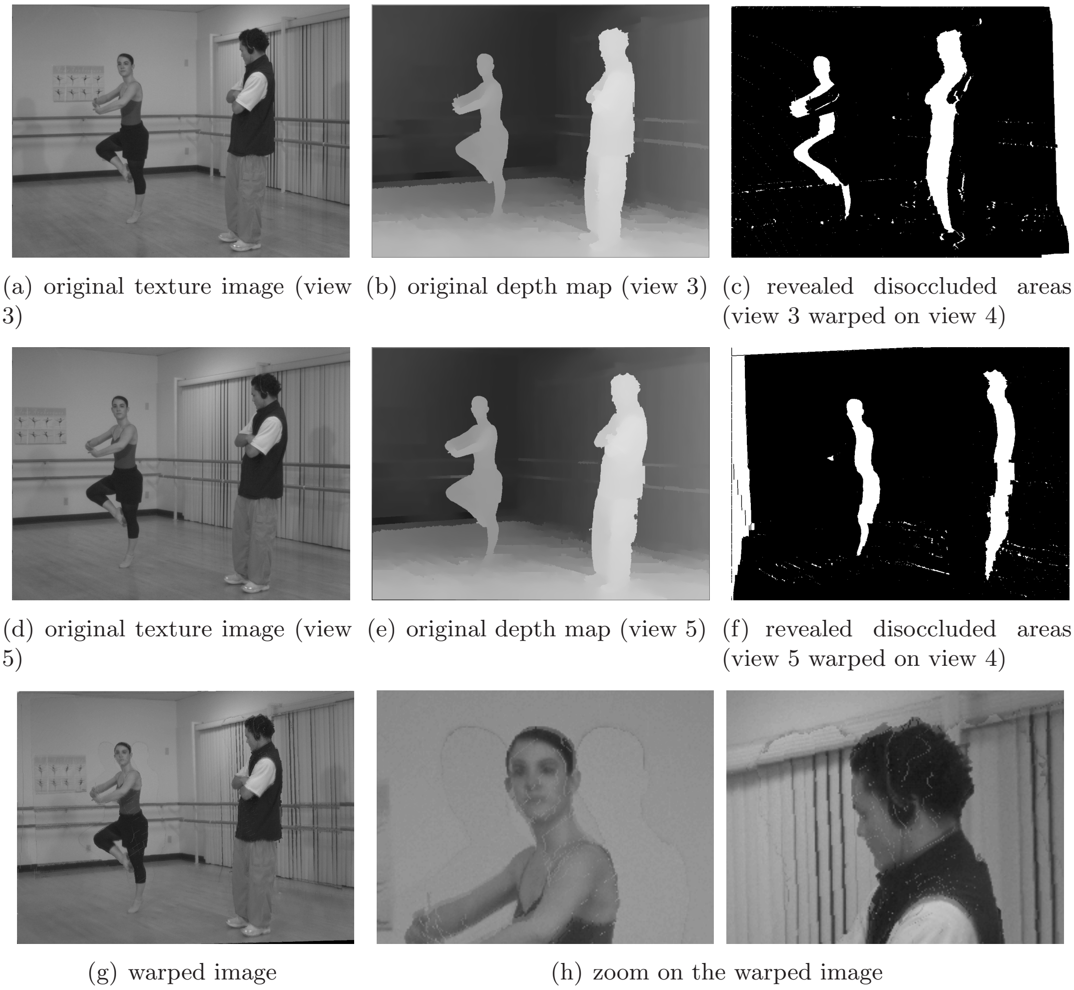
We can see that the shorter the wavelet filter bank is, the better the edges of the disoccluded areas are preserved, and thus, the quality of the warped image is better. Therefore, we can consider the Haar filter bank as the more efficient filter bank among the usual ones with respect to the rendering quality of the warped image, in despite of a lower compression ratio of the depth map compared with longer filter.

As a conclusion, preserving the discontinuities can be considering as an important constraint when compressing the depth data.

4.2 Adaptive edge-dependent lifting scheme

As discussed above, to perform a good 3D image warping of the reference view onto the reference view, it is essential to preserve the edge locations and sharpness in the depth map. In this section, we target edge preservation by designing an adaptive lifting scheme allowing to apply shorter filters over the edges of the depth map, allowing to reduce Gibbs (ringing) artifacts, and longer ones in homogeneous areas, to accomplish better compression efficiency.

[1] In opposition to the previous chapter where the 3D image warping is performed with the video-plus-depth data, here the multiview context allows to better handle the disocclusion problems by using two reference views.


(a) original texture image (view 3)

(b) original depth map (view 3)

(c) revealed disoccluded areas (view 3 warped on view 4)

(d) original texture image (view 5)

(e) original depth map (view 5)

(f) revealed disoccluded areas (view 5 warped on view 4)

(g) warped image

(h) zoom on the warped image

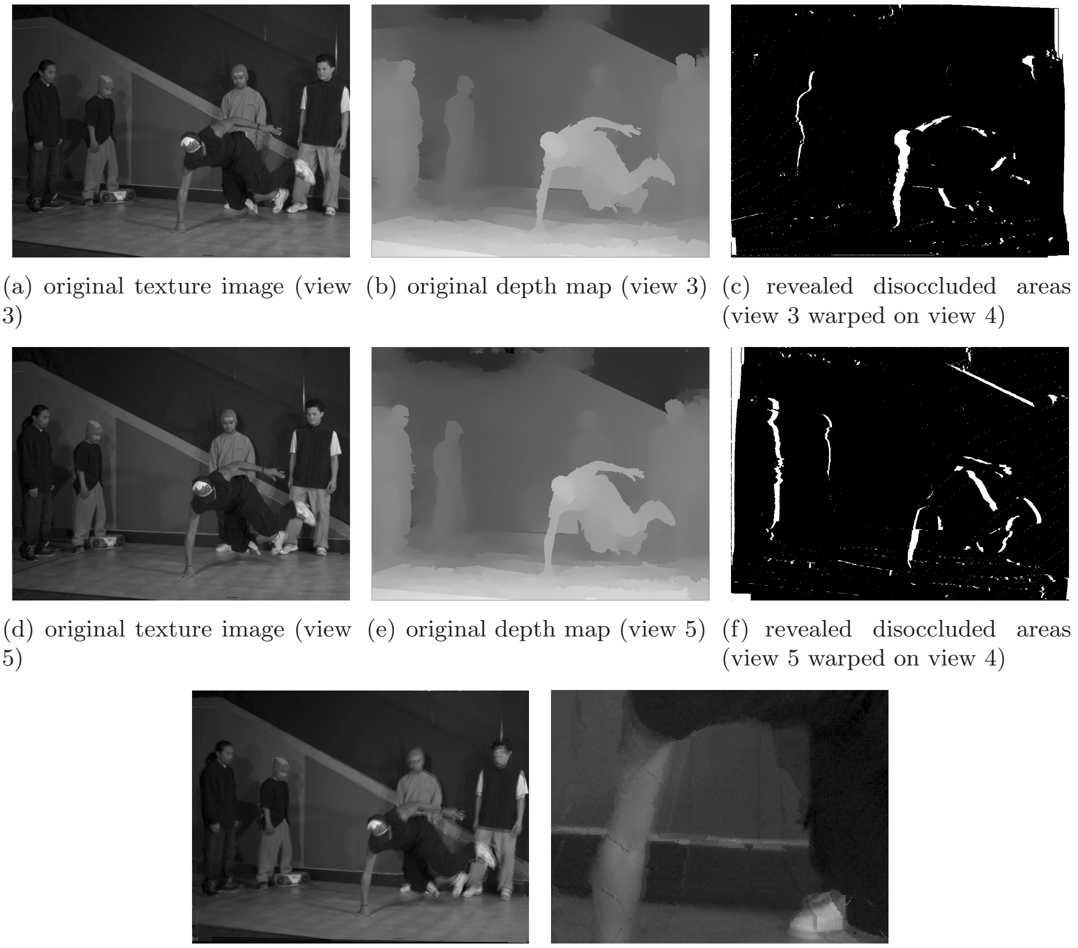
Figure 4.4: The edges of the depth map revealing disoccluded areas in the warped image (3D image warping of the view 3 and 5 on the view 4 of the multiview sequence "Ballet".

According to the RD results shown in Fig. 4.3 and Fig. 4.8, the Haar and Le Gall's 5/3 filter bank are well "fit" when considering respectively the quality of the synthesised view and the depth compression ratio.

In the following, we first introduce the lifting scheme, an efficient mechanism for implementing a wavelet filter bank. Moreover, the lifting scheme allows an easy implementation of adaptive DWT.

4.2.1 Lifting scheme

The lifting scheme, formally introduced by Sweldens [112, 113, 114], has proved its efficiency in wavelet-based image coding, due to its properties as in-place calculations and possible parallel implementation. Moreover, lifting schemes enable an easy design of non-linear and adaptive DWT. A very important feature of the lifting scheme is that every filter bank based on lifting automatically satisfies perfect reconstruction properties. The lifting scheme starts with a set of well known filters, where lifting steps are used in an attempt to improve (lift) the properties of a corresponding wavelet decomposition.



(a) original texture image (view 3)

(b) original depth map (view 3)

(c) revealed disoccluded areas (view 3 warped on view 4)

(d) original texture image (view 5)

(e) original depth map (view 5)

(f) revealed disoccluded areas (view 5 warped on view 4)

(g) warped image

(h) zoom on the warped image

Figure 4.5: The edges of the depth map revealing disoccluded areas in the warped image (3D image warping of the view 3 and 5 on the view 4 of the multiview sequence "Breakdancers".

Lifting steps: split, predict and update

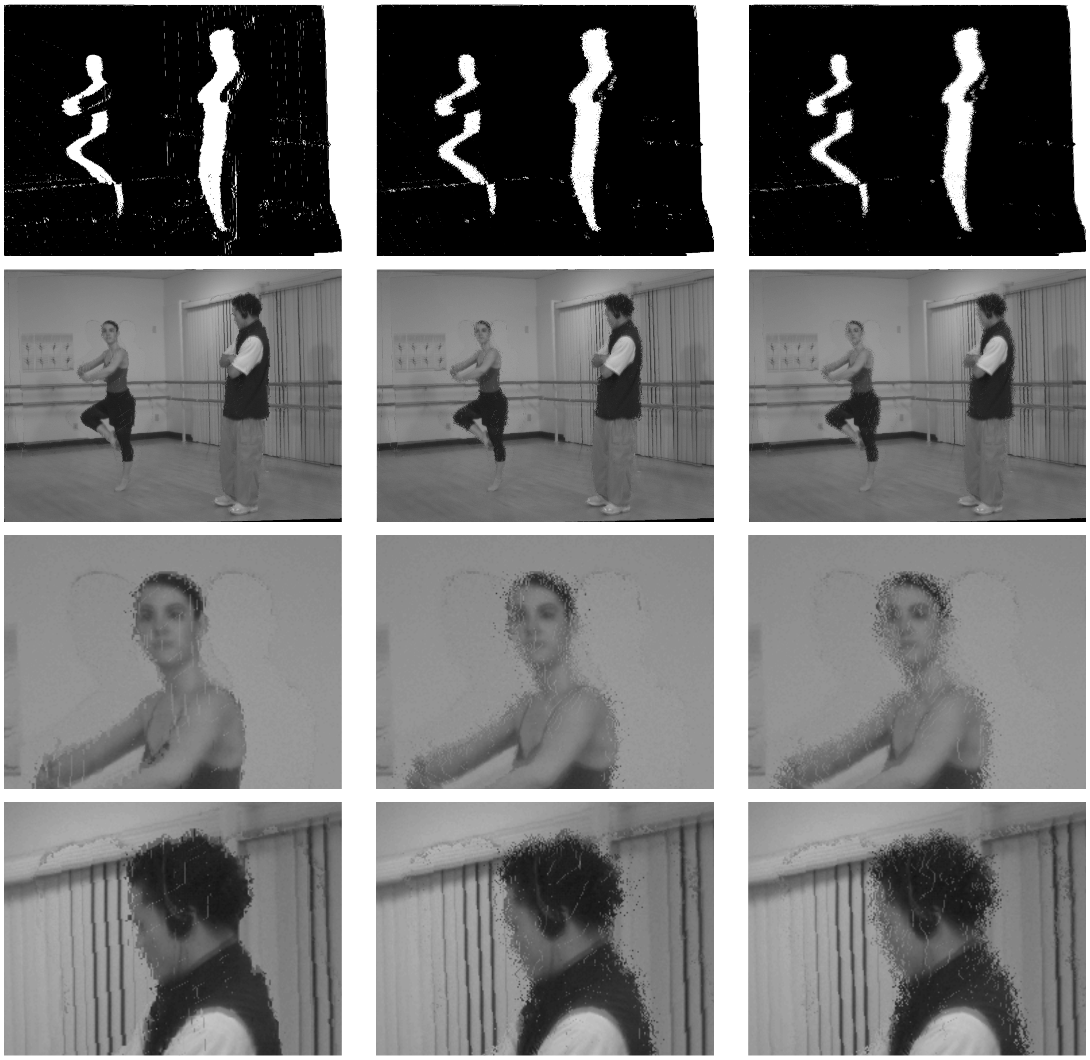
A typical DWT constructed by lifting scheme consists of three stages, as illustrated in Fig. 4.9.

Splitting The first step consists of splitting the input signal into two polyphase components: the even and odd samples x_{2i} and x_{2i+1} by means of a Lazy Wavelet Transform (LWT).

Predict As the two components x_{2i} and x_{2i+1} are correlated, the next stage predicts the odd values x_{2i+1} from the even ones x_{2i}, using a prediction P, and produces the residue

$$h_i = x_{2i+1} - \mathrm{P}\left\{(x_{2i})_{i \in \mathbb{N}}\right\} \tag{4.1}$$

where h denotes the detail subband coefficients.



(a) Haar (b) Le Gall's 5/3 (c) Daubechies 9/7

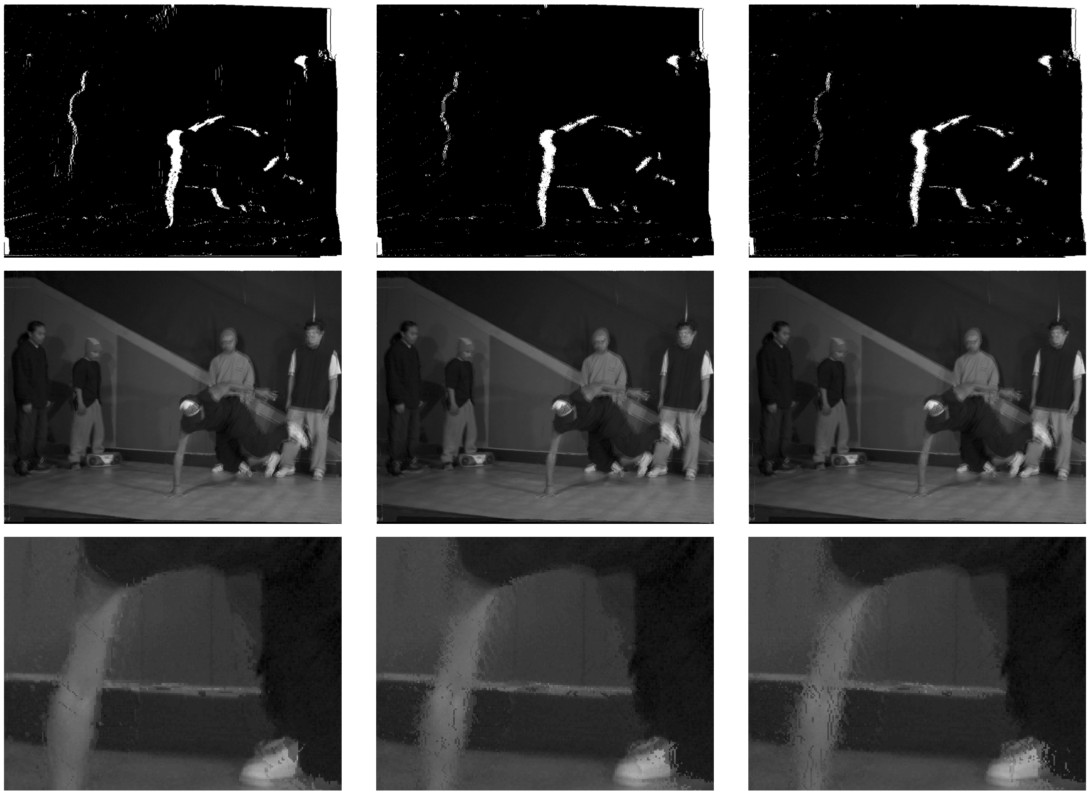
Figure 4.6: Effect of the wavelet-based depth compression onto the disoccluded areas and the warped image at 0.10 bpp ((up) the disoccluded areas revealed by the compressed depth map, and (down) the warped image onto the view 4 using reference views 3 and 5.)

Update An update stage U of the even values follows, such that

$$l_i = x_{2i} + \mathrm{U}\left\{(h_i)_{i\in\mathbb{N}}\right\} \tag{4.2}$$

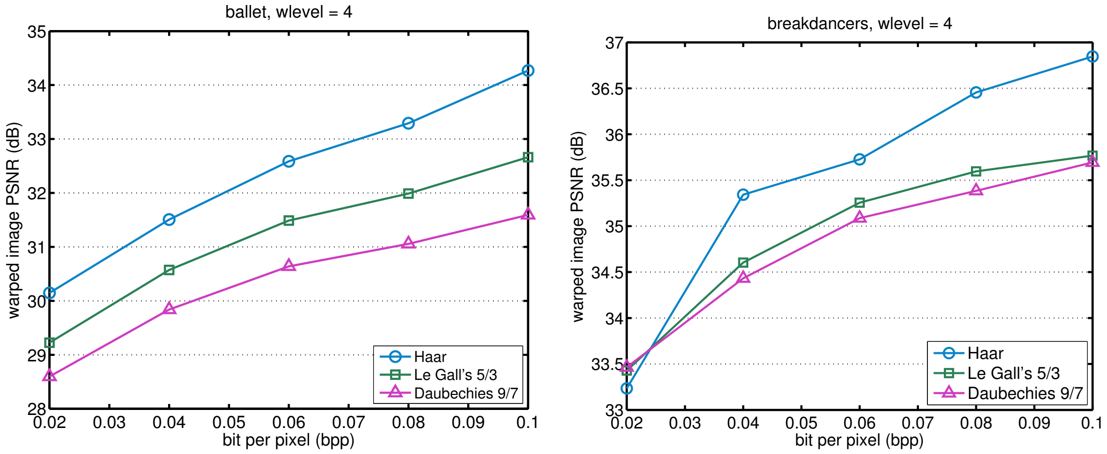
where l denotes the approximation subband coefficients.

These three steps can be repeated by iterating on the approximation subband l, thus creating a multi-level transform or a multi-resolution decomposition. The perfect reversibility of the lifting scheme is one of its most important properties. The reconstruction is done straightforwardly by reverting the order of the operations, inverting the signs in the lifting steps, and replacing the splitting step by a merging step. Thus, inverting the three steps procedure above results in:



(a) Haar (b) Le Gall's 5/3 (c) Daubechies 9/7

Figure 4.7: Effect of the wavelet-based depth compression onto the disoccluded areas and the warped image at 0.10 bpp ((up) the disoccluded areas revealed by the compressed depth map, and (down) the warped image onto the view 4 using reference views 3 and 5.)



(a) "Ballet" (b) "Breakdancers"

Figure 4.8: RD comparison of warped image using decoded wavelet-based depth map with different wavelet filter banks.

Undo Update

$$x_{2i} = l_i - \mathrm{U}\left\{(h_i)_{i\in\mathbb{N}}\right\}$$

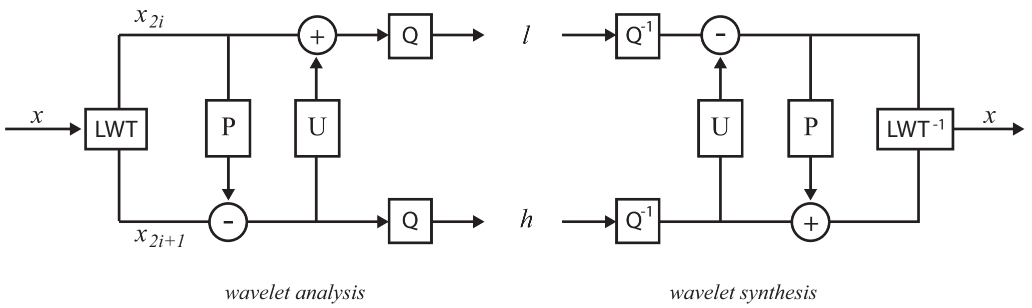


Figure 4.9: Lifting scheme composed of the analysis (left) and the synthesis (right) steps.

Undo Predict

$$x_{2i+1} = h + \mathrm{P}\left\{(x_{2i})_{i\in\mathbb{N}}\right\}$$

Merging

$$x = x_{2i} \cup x_{2i+1}$$

Lifting advantages

Some of the advantages of the lifting wavelet implementation with respect to the classical DWT are:

- simplicity: it is easier to understand and implement.
- the inverse transform is obvious to find and has exactly the same complexity as the forward transform.
- the in-place lifting computation avoids auxiliary memory requirements since lifting outputs from one channel may be saved directly in the other channel. Such implementation considerations are explained in [178].
- Daubechies and Sweldens proved in [32] that every biorthogonal DWT can be factorized in a finite chain of lifting steps.
- can be used on arbitrary geometries and irregular samplings.

Lifting implementations of some wavelet filter banks

As mentioned above, in this chapter we will use mainly two filters: Haar and Le Gall's 5/3 [65], where their corresponding lifting steps for one transform level of a discrete 1D signal $x = [x_k]$ are presented in the following.

Haar analysis lifting steps:

- Predict:

$$h_i = \frac{1}{\sqrt{2}}\left(x_{2i+1} - x_{2i}\right)$$

- Update:

$$l_i = \sqrt{2}x_{2i} + h_i$$

Le Gall's 5/3 analysis lifting steps:

- Predict:

$$h_i = x_{2i+1} - \frac{1}{2}\left(x_{2i} + x_{2i+2}\right)$$

- Update:

$$l_i = x_{2i} + \frac{1}{4}\left(h_{i-1} + h_i\right)$$

- Scaling:

$$h_i = \frac{1}{\sqrt{2}}h_i \quad \text{and} \quad l_i = \sqrt{2}l_i$$

4.2.2 Edge-dependent adaptive operators

As we seen in Fig. 4.1 and Fig. 4.2, applying shorter filters on the edges reduces Gibbs (ringing) artifacts.

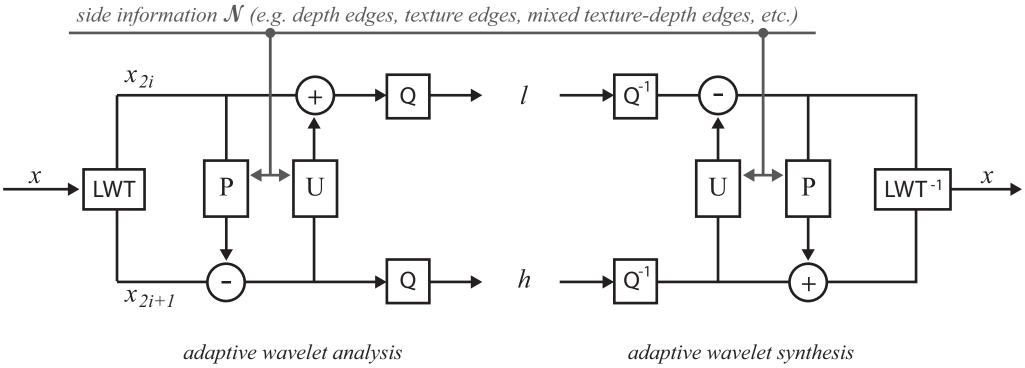

side information 𝒩 (e.g. depth edges, texture edges, mixed texture-depth edges, etc.)
x
LWT
x 2i
x 2i+1
P
U
+
-
Q
Q
l
h
Q-1
Q-1
-
+
U
P
LWT-1
x
adaptive wavelet analysis
adaptive wavelet synthesis

Figure 4.10: Adaptive lifting scheme.

We propose then applying in a close neighborhood $\mathcal{N}$ of an edge, a Haar-based prediction, and everywhere else a Le Gall's 5/3-based prediction. The prediction step applied (in a separable manner) in Eq. (4.1) becomes:

$$h_i = \begin{cases} x_{2i+1} - x_{2i} & \text{if } 2i+1 \in \mathcal{N} \\ x_{2i+1} - \frac{1}{2}\left(x_{2i} + x_{2i+2}\right) & \text{otherwise.} \end{cases} \tag{4.3}$$

The update step has to be adapted accordingly:

$$l_i = \begin{cases} x_{2i} + \frac{1}{2}h_i & \text{if } 2i \in \mathcal{N} \\ x_{2i} + \frac{1}{4}\left(h_{i-1} + h_i\right) & \text{otherwise.} \end{cases} \tag{4.4}$$

And the coefficients are scaled as follows:

$$h_i = \frac{1}{\sqrt{2}}h_i \quad \text{and} \quad l_i = \sqrt{2}l_i \tag{4.5}$$

As shown in Fig.4.10, the reversibility of the proposed scheme is based on the assumption that it is possible to obtain the same neighborhood $\mathcal{N}$ of the edges at the encoder side and the decoder side by utilizing for example the edges information of the independently coded texture image, or directly the depth edges using additional bits to transmit.

Edge detector The edges are computed by using a symmetric separable derivative of the image (but any edge detector can be used instead), defined in 1D as follows:

$$x'_i = x_i - \frac{1}{2}\left(x_{i-1} + x_{i+1}\right) \tag{4.6}$$

Next, we apply a threshold on the coefficients to find relevant edges. To handle the problem of choosing an appropriate threshold, we use an approach by *hysteresis*[2], where multiple thresholds are used to find an edge. Note that the image is pre-filtered thereby bilateral 3-by-3 filtering to enhance the edge detection.

We define then the neighborhood $\mathcal{N}$ of an edge as the set of all the positions i defined by $|i - \delta| < \tau$, where δ is the coordinate of the nearest edge and τ is a parameter allowing to indicate the proximity to the edge.

The main difficulty in such an adaptive scheme is to retrieve the same edges at the encoder and decoder side, and thus to maintain the reversibility of the spatial transform. To fulfill this condition, we propose several possible side informations by taking advantage of the MVD representation.

Depth edges as side information

A straightforward solution would be to utilize the depth edges themselves as side information, at the cost of increasing the bandwidth in losslessly transmitting the key edge locations required to operate the inverse spatial transform.

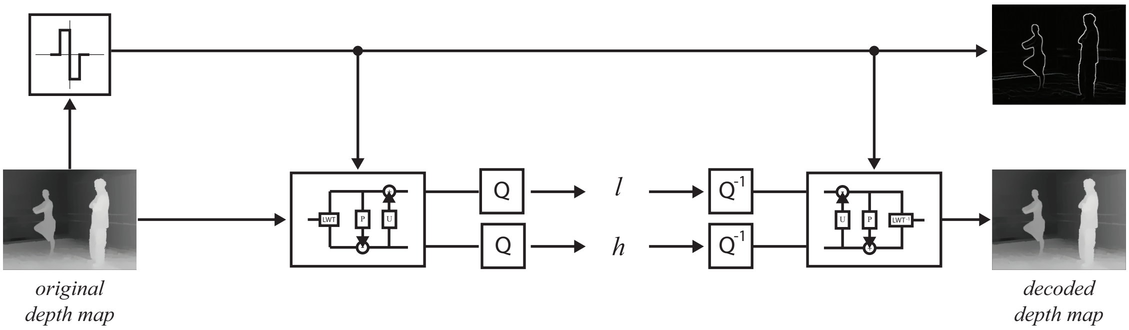


Figure 4.11: Adaptive lifting scheme using depth edges as side information.

Despite the simplicity of this solution, depth edges have to be sent for each level of multi-resolution wavelet analysis, which may increase rapidly the bandwidth.

In the following, we propose different approaches that do not increase the bandwidth with any additional data.

Texture edges as side information

On the other hand, in order to suppress the additional bits to send, we propose exploiting the contour information from the texture image as depth map side information, which is independently encoded and previously transmitted to the decoder.

[2] Hysteresis is used to track the more relevant pixels along the contours. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless the pixel is located near a edge detected by the high threshold.

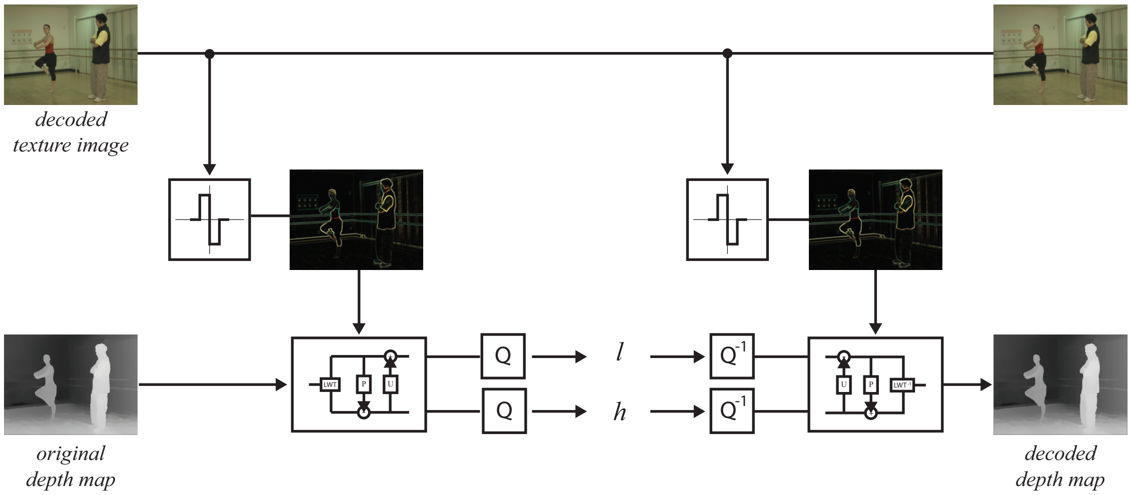
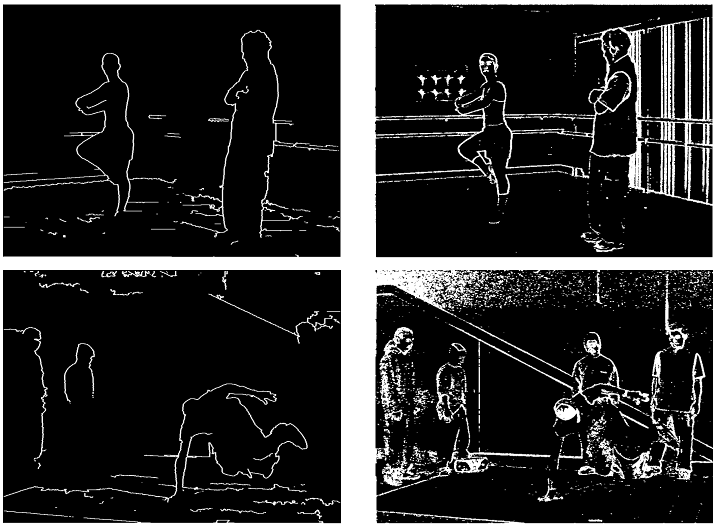


Figure 4.12: Adaptive lifting scheme using texture edges as side information.

However, the texture presents much more contours than the depth map, which leads to unnecessary short filters for the depth map.



(a) depth edges (b) texture edges

Figure 4.13: Example of edges of the texture image and depth map.

Interpolated depth edges as side information

Here, we propose extracting the spatial locations of the edges from the approximation coefficients of the depth map. These coefficients are used to obtain an interpolated depth map.

At the encoder, the decimation and interpolation operation are done by first using the Le Gall's 5/3 filter analysis, and then by putting all the detail coefficients to zero, performing the Le Gall's 5/3 filter synthesis to compute the interpolated depth map. Edge detection is now performed on this interpolated depth map. This operation is lossy and an attempt to simulate the edge detection at the decoder side.

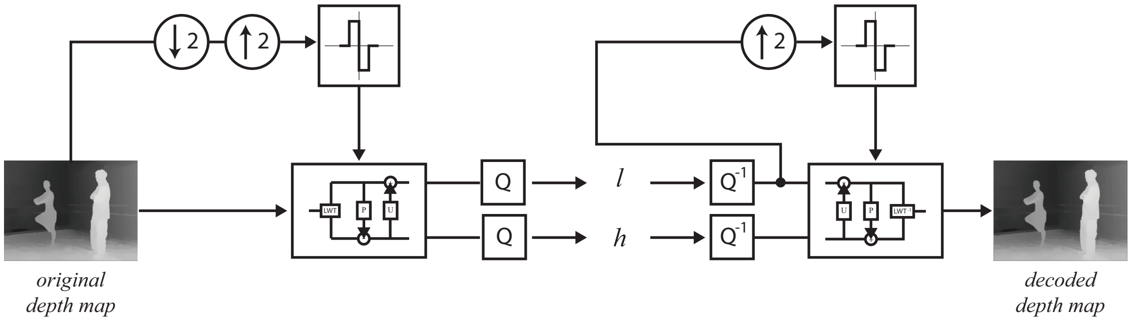
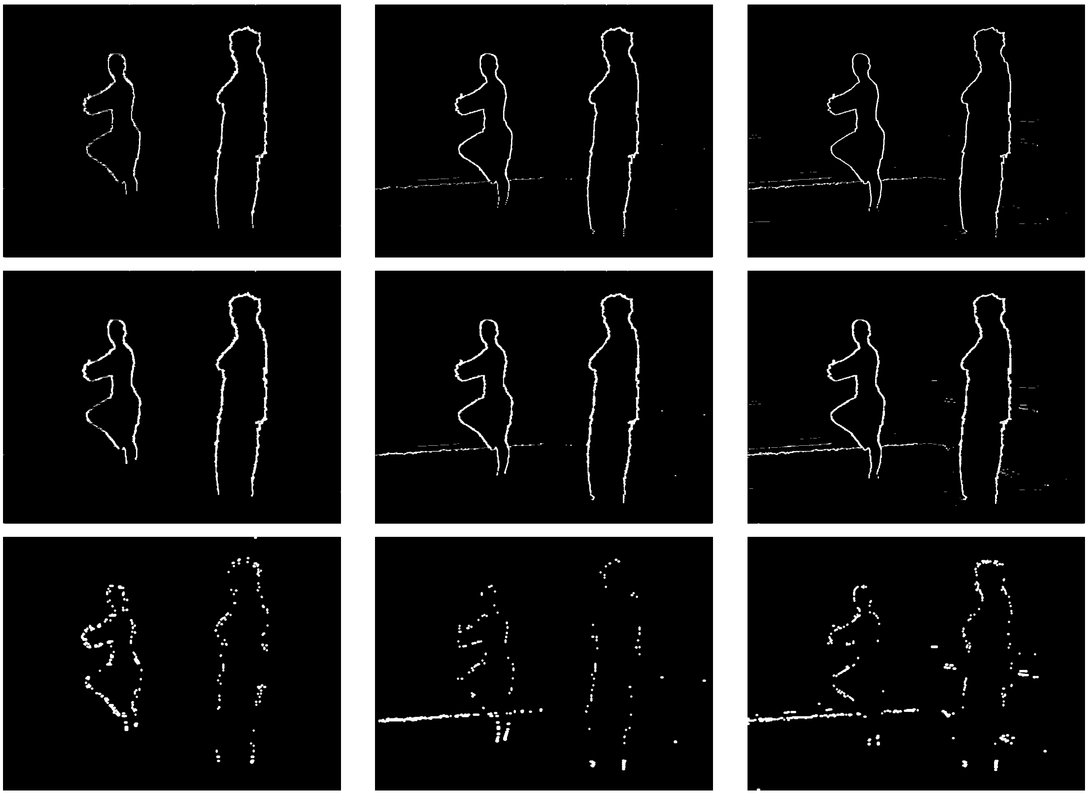


Figure 4.14: Adaptive lifting scheme using the approximated depth edges as side information.

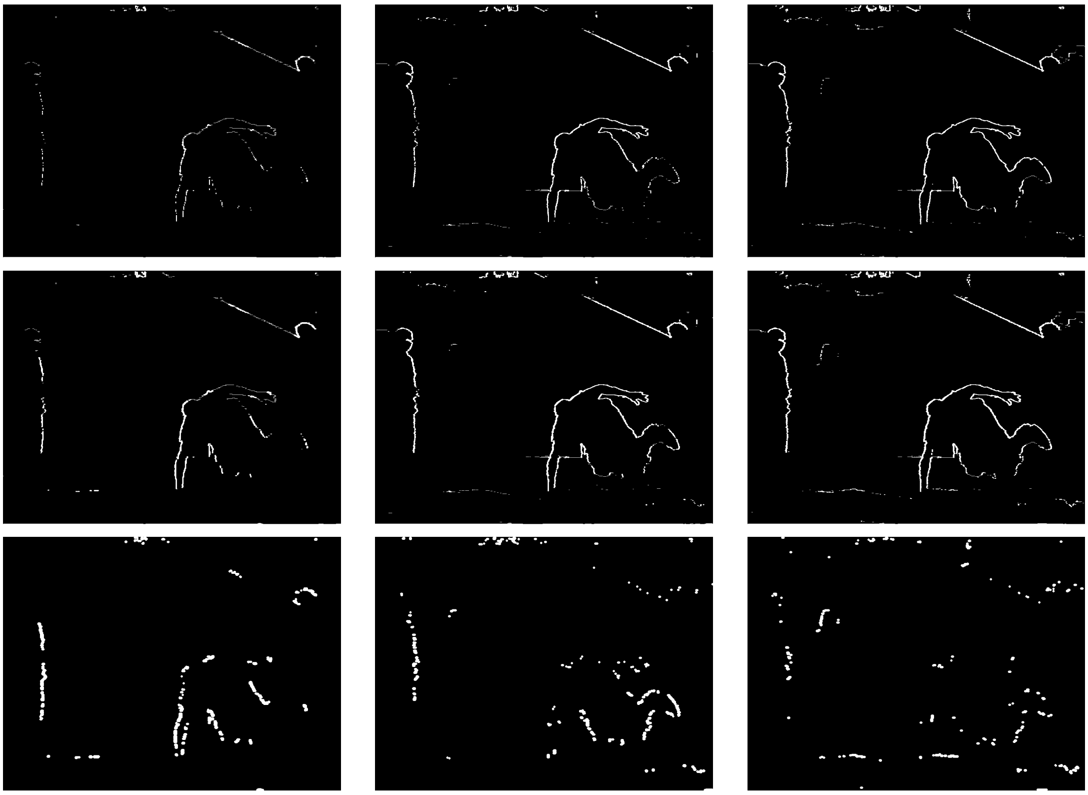
At the decoder, the interpolated depth map is built from the approximation coefficients, while at the encoder, a "dummy" linear transform based on the long filters is used. The reconstruction is still possible due to the particular properties of smoothness of the depth map, which permits to preserve the location of the edges when using the two slightly different decompositions.

However, the edges of the interpolated depth map are very sensitive to the bitrate, as we can see in Fig. 4.15 and Fig. 4.16. And the slightly difference between the two interpolated depth edges does not allow a perfect reversibility of the adaptive lifting scheme.



(a) 0.02 bpp (b) 0.04 (c) 0.06

Figure 4.15: Interpolated depth edges at different bitrate ((up) at the encoder, (middle) at the decoder, (down) difference between both.



(a) 0.02 bpp (b) 0.04 (c) 0.06

Figure 4.16: Interpolated depth edges at different bitrate ((up) at the encoder, (middle) at the decoder, (down) difference between both.

Mixed texture-depth edges as side information

The edges of the texture image and those of the depth map present strong similarities (see Fig. 4.13). One of the previous ideas was to use the edges of the texture image, which is independently coded, to detect where to apply shorter filters on the depth map.

We propose here strengthening the previous ideas by using jointly the texture edges and the interpolated depth edges, where the information extracted from the decoded texture image is validated with some information from the depth map. Such information will be obtained from the approximation coefficients of the depth map.

The contours in the texture image will be validated if they have in a neighborhood $\mathcal{N}$ a corresponding edge in the interpolated depth map. This is possible due to the correlation between the texture image and the depth map.

Then, we validate a pixel to belong to the final mixed texture-depth edge $\mathcal{E}$ if it belongs both to the original edge of the texture $\widetilde{I}$ and to a neighborhood $\mathcal{N}$ of the interpolated depth map $\widetilde{D}$ edges, as described in pseudo-code bellow:

```
$\check{I}$ : decoded texture image
$\widetilde{D}$ : interpolated depth map
for all $(i,j) \in \check{I} * h$ do
  if $(i,j) \in \mathcal{N}(\tilde{D} * h)$ then
    $\mathcal{E} \leftarrow (i,j)$
  end if
```

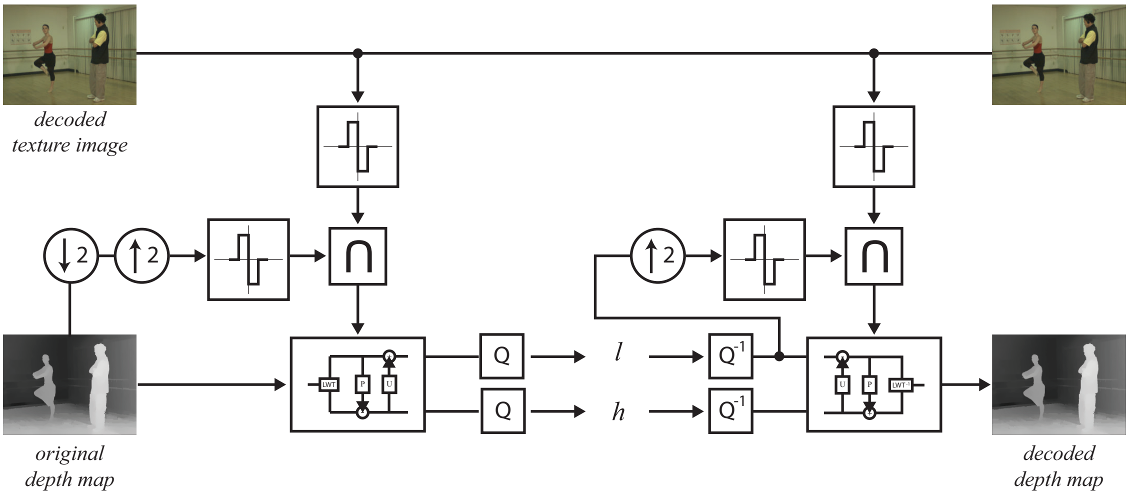
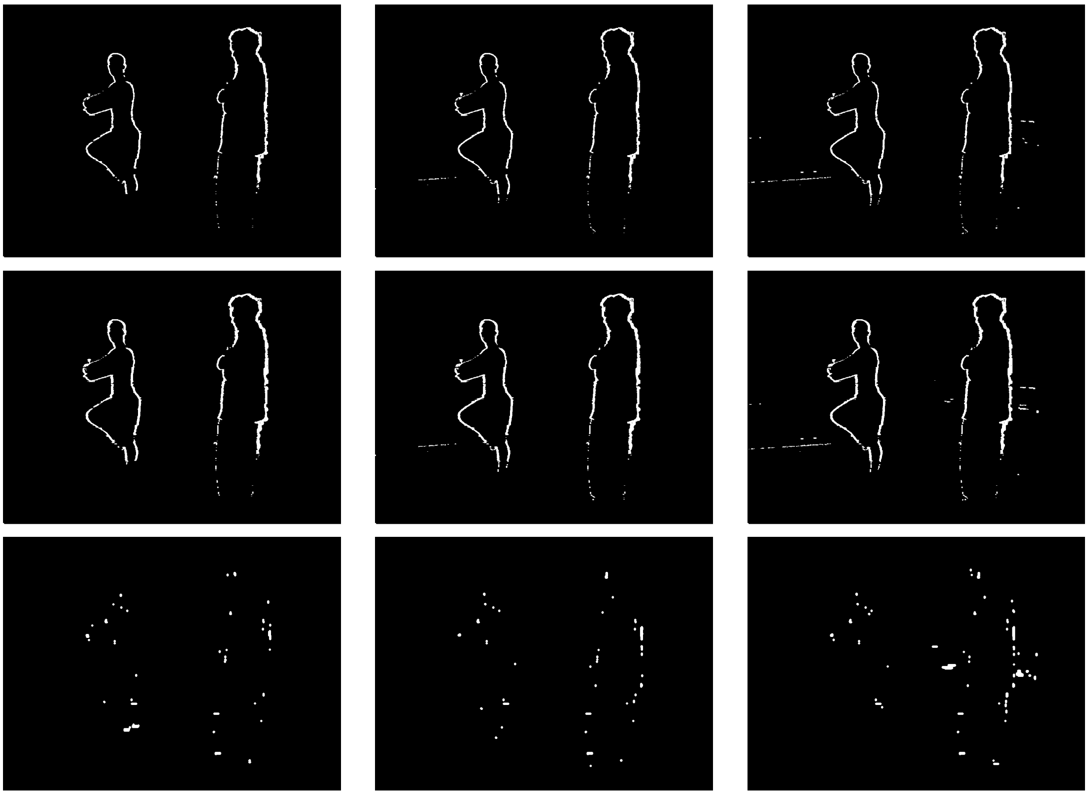


Figure 4.17: Adaptive lifting scheme using mixed texture-depth edges as side information.

end for

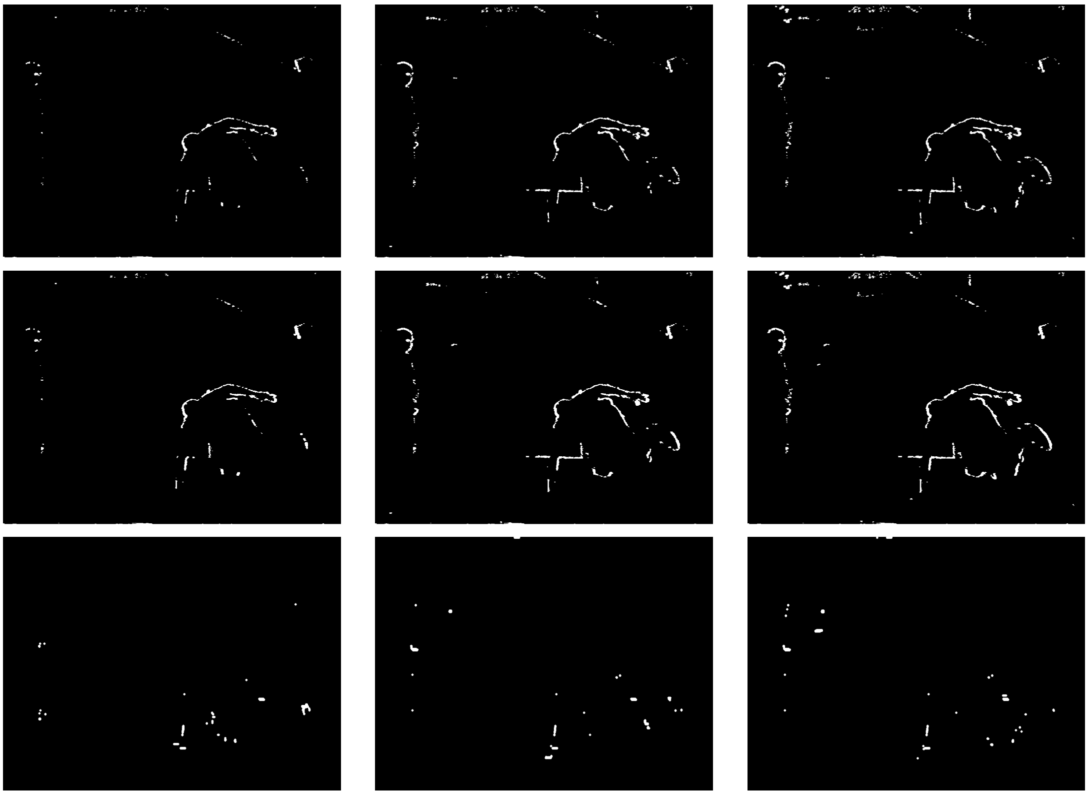
where h denotes the impulse response of the high-pass filter described by Eq. (4.6).

Moreover, the real differences between the two interpolated maps are not crucial for the edge detection in the texture image, since only a neighborhood of the edges is used to validate the texture contours.



(a) 0.02 bpp (b) 0.04 (c) 0.06

Figure 4.18: Mixed depth edges at different bitrates (up) at the encoder, (middle) at the decoder, (down) difference between both.



(a) 0.02 bpp (b) 0.04 (c) 0.06

Figure 4.19: Mixed depth edges at different bitrates ((up) at the encoder, (middle) at the decoder, (down) difference between both.

As shown in Fig. 4.18 and Fig. 4.19, this allows to retrieve from the pair texture plus interpolated depth map the location of the original edge of the depth map.

4.2.3 Experimental results

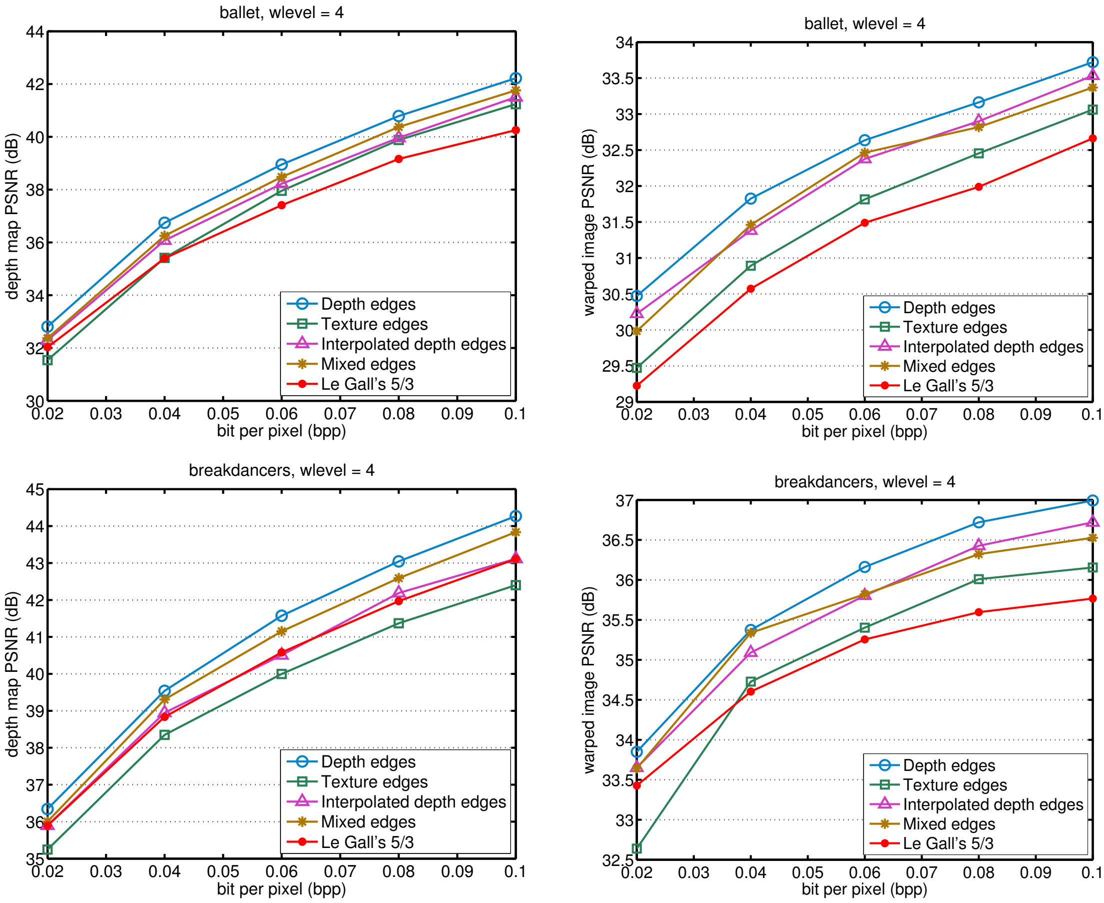
This section is devoted to evaluating the coding efficiency of the proposed method against the linear 5/3 filter bank with the JPEG2000 codec as the entropy coding. For the results, the multi-resolution wavelet decomposition is performed on 4 levels, by applying the described adaptive procedure at each decomposition level.

The MVD sequences “Ballet” and “Breakdancers” from the camera 3 and 5 provided by Microsoft [7] have been used to test the proposed method. The calibration parameters are supplied with the sequences. The depth map provided for each camera is estimated via a color-based segmentation algorithm [143]. The synthesized view is rendered by merging the ones from the two reference cameras.

In order to fill in the missing values from the 3D image warping, two reference views has been used, and the remaining occluded areas are inpainted based on the fast marching method [121].

Fig. 4.20 compares the coding efficiency between the different proposed adaptive lifting scheme method, and the linear wavelet filter bank Le Gall’s 5/3. As seen in Section 4.1.1, the 5/3 performs better than the 9/7, contrary to natural images. This is due to the very particular features of the depth map, which is much smoother than natural images, and presents sharp edges. The allocated bitrate used to encoded the depth map is equal to 20%

of the bitrate of the texture image. Note that in Fig. 4.20, the increase in bitrate related to the depth edges side information has been neglected when reporting experimental data about the coding rate, which is equivalent to perfectly retrieve the depth edges at the decoder side.

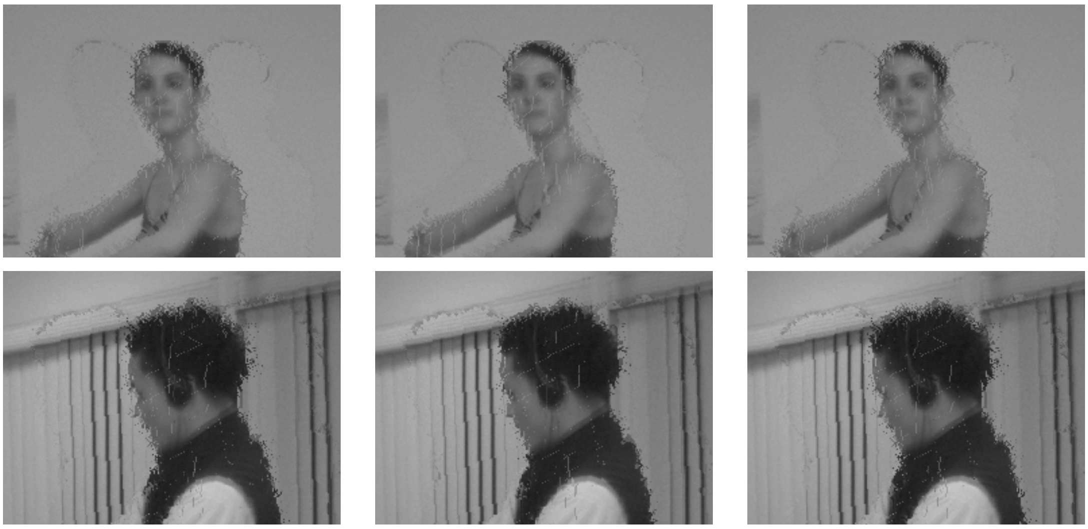


(a) depth map

(b) warped image

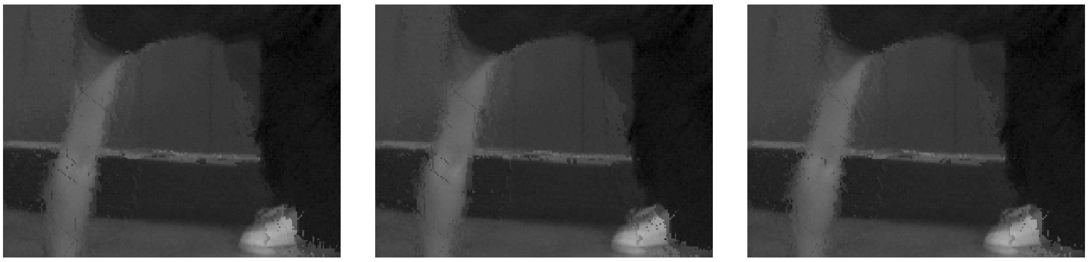
Figure 4.20: RD results of the depth maps and warped images((up) “Ballet”, (down) “Breakdancers”).

The gain in the depth map coding becomes more perceptible when measuring the PSNR of the warped image. Actually, the warped image PSNR measurements indicate a quality gain around 1.8 dB for the “Ballet” image and around 1 dB for the “Breakdancers” image. Thus, the proposed method does not necessarily improve the overall quality of the transmitted depth map over a classic 5/3 lifting scheme, but for a similar PSNR, our method by its adaptive decomposition better preserves the edges, and consequently a quality improvement of the synthesized view can be perceived by the final user. Here, all the three decompositions have almost the same depth PSNR. However, the corresponding RD characteristics in the reconstructed second view shows a sensible improvement for the adaptive scheme.



(a) depth edges (b) texture edges (c) mixed edges

Figure 4.21: Effect of the wavelet-based depth compression onto the “Ballet” warped image at 0.10 bpp.



(a) depth edges (b) texture edges (c) mixed edges

Figure 4.22: Effect of the wavelet-based depth compression onto the “Breakdancers” warped image at 0.10 bpp.

4.3 Conclusion

In this chapter, an adaptive lifting scheme for encoding depth maps has been proposed. By applying shorter filters over edges, the energy of the detail coefficients has been reduced and the location of the edges better preserved in the reconstructed depth map. Adaptivity has been introduced through the use of side informations, representing with some respect the edges of the depth map, to switch filters in the depth map decomposition.

Experimental results illustrate the capacity of the proposed adaptive lifting to achieve efficient compression of the depth map, which also led to a sensible quality improvement of the synthesized second view. Consequently, the development of advanced algorithms for MVD coding needs to optimize the RD performance with respect to not only the distortion of the depth video, but also the distortion of synthesized views. This requires future research to address joint texture/depth compression scheme based on view synthesis distortion.

For depth coding with layered depth video (LDV) data, a utilization of the contour of the residual data can be considered, which will be straightforwardly used as side information.

Future works concern the extension of the proposed method to the depth map coding in multi-view video sequences decomposed with a motion-compensated $t + 2D$ wavelet decomposition using multiple auxiliary components in reference to MPEG-4 MAC.

Naturally, this work can be applied with texture coding by using the edges of the depth map. In this case, we can expect to bring into play the Daubechies 9/7 filter in the design of the adaptive DWT.

Chapter 5

MPEG-2-based video-plus-depth coding

Contents

In this chapter, we give an attempt to jointly encode a video-plus-depth sequence thereby a joint temporal prediction and a joint target-bit allocation strategy. Indeed, the video-plus-depth data representation uses a regular texture video enriched with the so-called depth video, providing the depth-distance for each pixel. The compression efficiency is usually higher for smooth gray level data representing the depth map than for classical video texture. However, improvements of the coding efficiency are still possible, taking into account the fact that the video and depth sequences are strongly correlated.

In the first part, we propose reducing the amount of information for describing the motion of the texture and depth sequences by sharing one common motion vector field. The shared motion vector field is then optimized by a joint motion estimator. The saved bits are then used to encode the depth residual transformed coefficients.

In the second part, we give more interest to the rate control algorithm. In the literature, the bitrate control scheme generally fixes for the depth video sequence a percentage of 20% of the texture stream bitrate within MPEG-2 framework [36]. This value has been proposed for example in the project ATTEST by considering a separable scheme where the texture video is encoded independently with MPEG-2 (for backward-compatibility with existing TV solutions) and the depth video with H.264/MPEG-4 AVC. However, this fixed percentage can affect the depth video coding efficiency, and this percentage should also depend of the specificities of each video sequences. For these reasons, we propose a new bitrate allocation strategy, which considers both the texture and its associated per-pixel depth informations, by taking into account the texture/depth motion activity relationship.

5.1 Global motion vector sharing

In this section, we present an alternative method for encoding video-plus-depth sequences that utilizes a novel joint motion estimator.

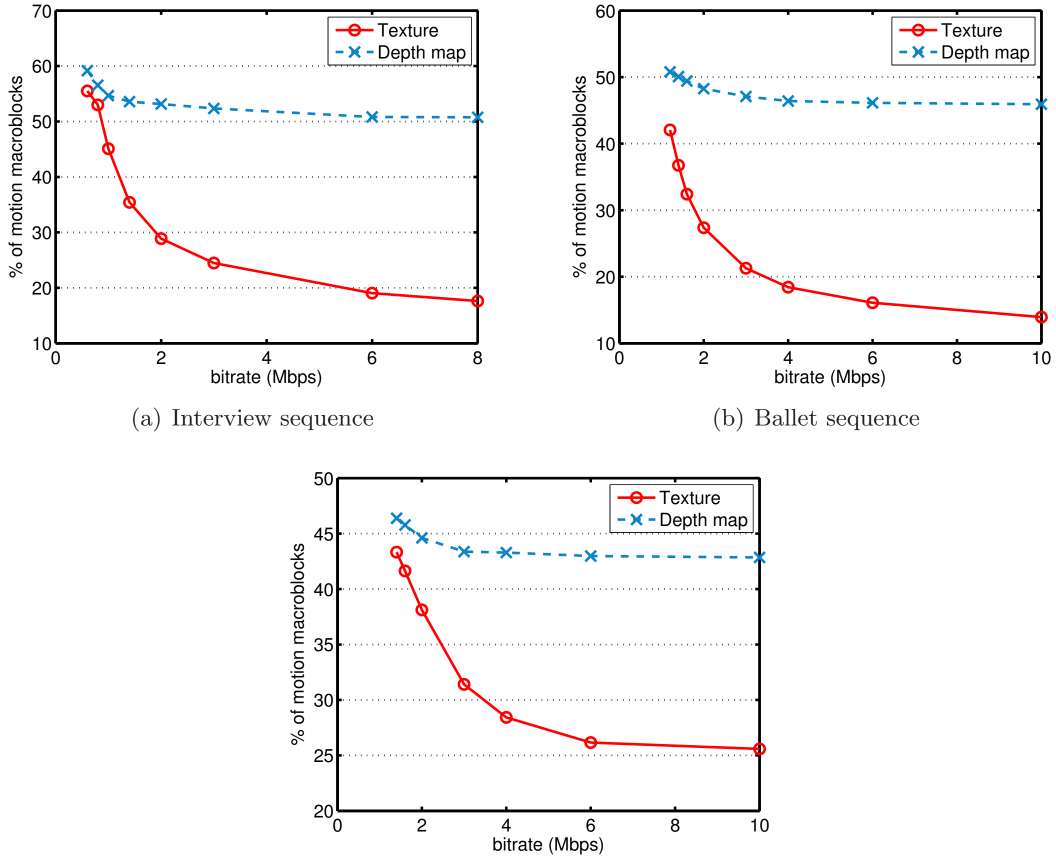
As we know, motion vector (MV) fields are the result of the motion estimation between pictures in a video sequence, in which temporal redundancies are removed by estimating the inter-picture motion and then generating the MV field that minimizes the temporal prediction error.

According to MPEG standard, encoded MVs reside in predictive P-pictures and bi-directional B-pictures bitstream. Consequently, in a typical MPEG-2 GOP for broadcasting purposes, having the structure IBBP, the number of coded MBs in temporal predictive mode (in opposition to *intra* mode) can reach 40% of the total number MBs at low bitrate (as shown in Fig. 5.1), and as a result, the transmission of motion data consumes a large part of the bitstream for low bitrate coders.

The video-plus-depth stream contains usually twice this number of motion data, corresponding to the MVs in the texture stream and in the depth stream.

Instead of working on the separate efficiency of the two MV fields, in order to minimize the prediction error in both cases, we show that only one MV field is enough inside the global stream, since the motion in both videos is correlated. As the texture and the depth videos are spatially correlated, the MVs in the two sequences should also be correlated. Hence, one of the aims of this work is to reduce the amount of information for describing the motion of video-plus-depth sequences by sharing one common MV field as illustrated in Fig.5.2.

Intuitively, the texture video and its associated depth video have common characteristics, since they describe the same scene with the same viewpoint. For that reason, in

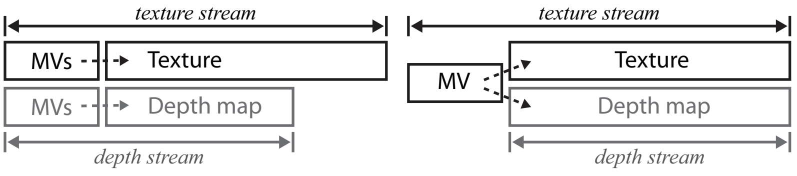


(a) Interview sequence

(b) Ballet sequence

(c) Breakdancers sequence

Figure 5.1: Percentage of the coded predictive MBs (forward and backward) inside the video sequence.



(a) Independent MV fields

(b) Shared MV fields

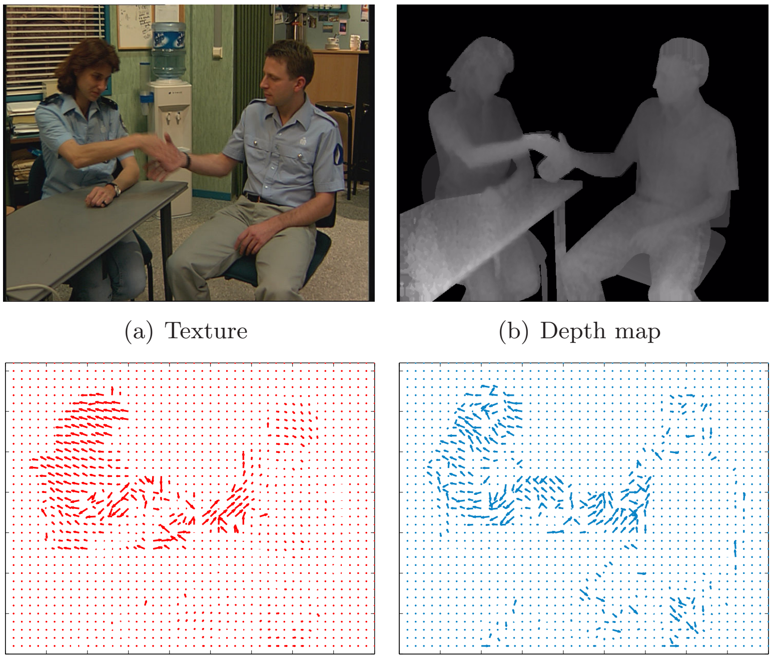
Figure 5.2: Different strategies for MV encoding: (a) separate MV for texture and depth videos, and (b) a common MV field for texture and depth sequences.

both domains (*i.e.* texture structure and distance information) boundaries coincide and the direction of motion should be the same.

To prove this hypothesis, we first observe the MVs correlation inside a video-plus-depth sequence. Afterwards, we propose exploiting this observed physical relation of the motion in both videos (the texture and depth video sequences).

5.1.1 Correlation

By construction, a video-plus-depth image contains two similar data representations, the texture and the depth map. Consequently the motion contained in the two sequences may be similar at the same spatial location, and should take similar directions as we can see in Fig. 5.3, Fig. 5.4 and Fig. 5.5.



(a) Texture

(b) Depth map

(c) Texture MV field

(d) Depth MV field

Figure 5.3: Example of MV fields from the frame 80 of the sequence "Interview".

Despite that the motion in both scene representation are the same, the resulting MV fields differ, which is due to the fact that MVs are optimized in term of coding efficiency, so, they do not necessarily represent in every particular the real motion present in the scene, resulting in a not perfect match between the texture video and the depth video. However the two MV fields are still correlated.

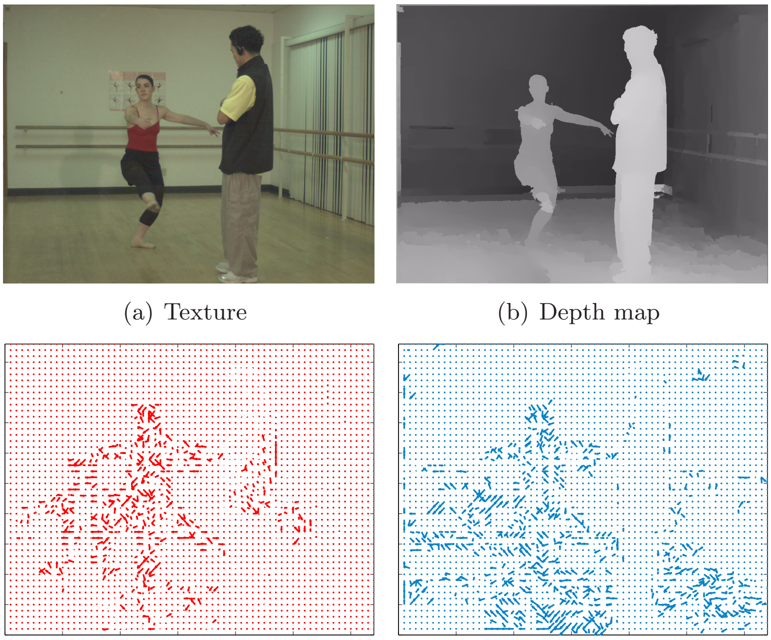
Let us define the correlation coefficient between the MV field of a specific frame of the texture $\mathbf{v}_t$ and the depth $\mathbf{v}_d$ as follows:

$$r(\mathbf{v}_t, \mathbf{v}_d) = \frac{Cov(\mathbf{v}_t, \mathbf{v}_d)}{\sqrt{Var(\mathbf{v}_t) \cdot Var(\mathbf{v}_d)}}, \tag{5.1}$$

where $Cov(.,.)$ is the covariance and $Var(.)$ is the standard deviation.

As expected, the motion analysis thereby using the correlation coefficient and the average difference (see Fig. 5.6) confirms the correlated location of the motion information. Therefore, coding efficiency would be improved if one can efficiently use this correlation.

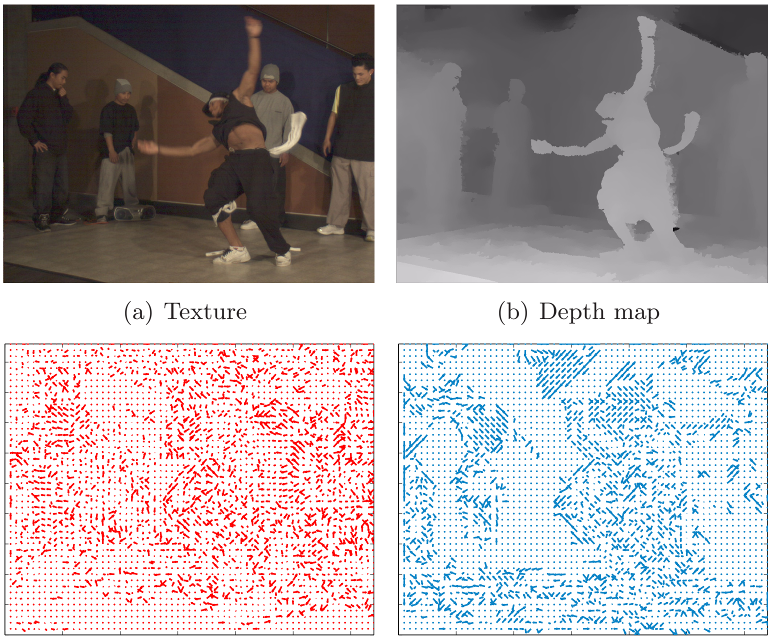
We found in the literature some works about the utilization of the motion correlation between the texture and the depth videos sequence. For example, Grewatsch and Müller proposed in [45] to re-utilize the MVs found in the texture video into the depth video coding scheme, without any modification. On the other hand, Oh et Ho exploited the motion texture information in [92] to reduce the motion estimation complexity of the depth video



(a) Texture (b) Depth map

(c) Texture MV field (d) Depth MV field

Figure 5.4: Example of MV fields from the frame 80 of the sequence “Ballet”.

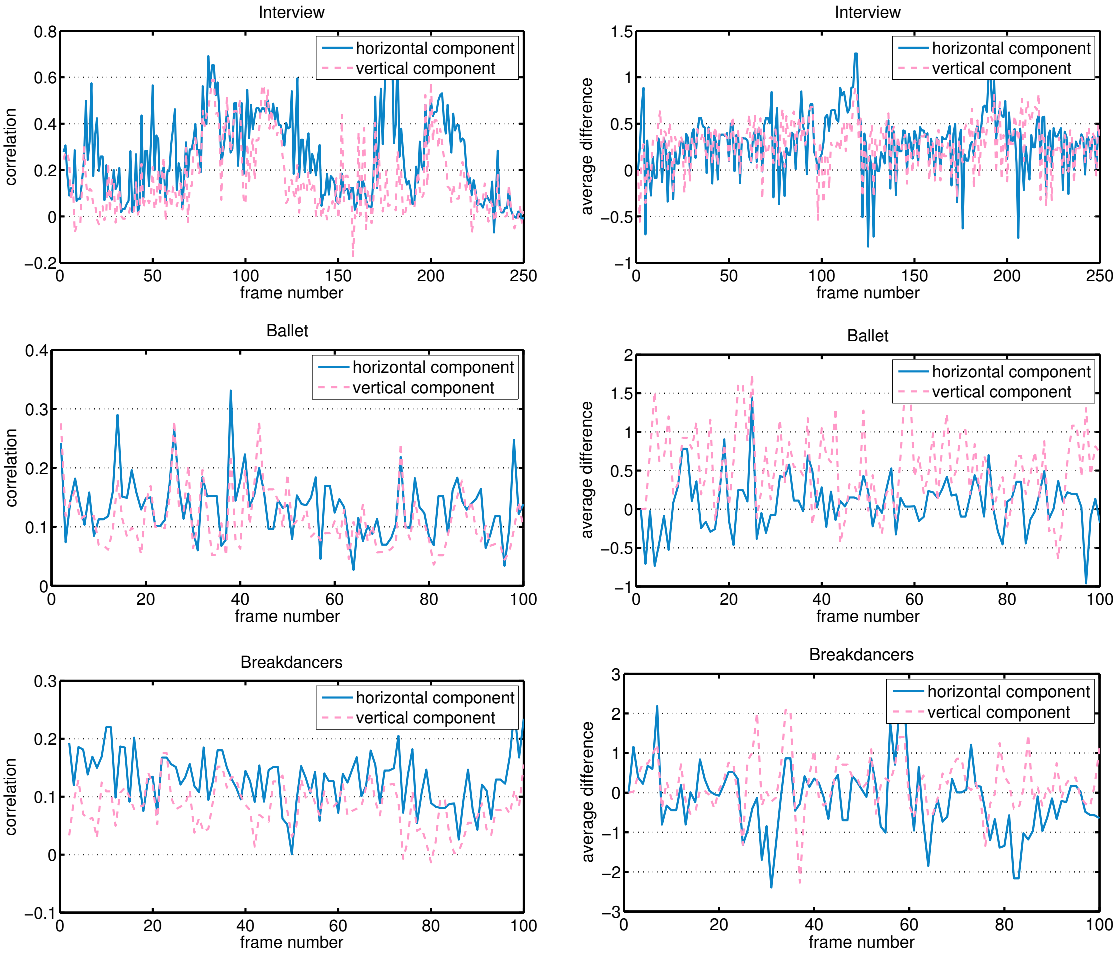


(a) Texture (b) Depth map

(c) Texture MV field (d) Depth MV field

Figure 5.5: Example of MV fields from the frame 80 of the sequence “Breakdancers”.

encoding. More recently, Tao *et al.* utilize the scalable extension of H.264/MPEG-4 AVC in [118] to encode the video-plus-depth data, where the texture video is coded as the base layer, and the depth map as the enhancement layer. Thus, texture MVs can be directly used as predictors for the depth MVs.



(a) Correlation coefficient (b) Average difference

Figure 5.6: MV analysis the between the texture and depth videos (from the top to bottom: frame 80 from the sequence “Interview”, “Ballet” and “Breakdancers”).

In our approach, the MV sharing idea is extended, by introducing into the estimation a joint criterion for the minimization of the two energies, of the texture video, and of the depth video.

5.1.2 Joint distortion measure

The MV of a MB is estimated by considering the best matching block, corresponding in general to the minimum mean squared error (MSE) or mean absolute error (MAE) [89] with regard to the previous frame. Let $I_{\text{ref}}(x, y)$ denote the image intensity at the spatial location (x, y). The vector $\mathbf{v}(v_x, v_y)$ maps points in the current frame I_{curr} to their corresponding locations in the reference picture I_{ref}. For illustration, MSE is defined as follows:

$$\text{MSE} = \frac{1}{N^2}\sum_{x=0}^{N}\sum_{y=0}^{N}\left[I_{\text{curr}}(x,y) - I_{\text{ref}}(x+v_x, y+v_y)\right]^2 \qquad (5.2)$$

Based on the observation in the previous section, we argued about the need to share the MVs by encoding and transmitting only one MV field for both the texture and depth videos. That leads to account for both the distortion in texture and depth videos by defining a new motion estimation, where the distortion criterion to minimize is this time defined jointly for the texture and the depth videos as follows:

$$MSE_{\text{joint}} = (1-\alpha) \cdot MSE_{\text{texture}} + \alpha \cdot MSE_{\text{depth}} \tag{5.3}$$

where $\alpha \in [0,1]$ controls the relative importance given to the texture and to the depth for this estimation procedure.

According to the proposed distortion metric, the resulting MV field is used for the two streams, and then encoded only once. The value $\alpha = 0$ is a particular case already studied in [45], and later inside MPEG-4 MAC [62], where only the MV field from the texture information is considered to encode both the texture and the depth video sequences. In our method, we generalize this concept and investigate the problem of estimating a MV field which can reduce the temporal correlation as well for the depth information as for the texture data, by means of the joint estimation criterion. In the experiments, we tune the parameter α to find the optimal value depending on the content of the sequence.

Once the common motion field is found, it has to be encoded for transmission. The motion field used to encode both the texture and the depth video sequences is placed in the texture bitstream, to ensure the required backward compatibility with current TV set-top boxes.

As illustrated in Fig. 5.2, the MVs are shared and only sent once in the global video-plus-depth stream. Consequently, this strategy allows more bandwidth resources to the depth residues. Moreover, it overcomes the imperfect match between the two MV fields. In fact, the correlation error is less significant compared with the gain in bandwidth.

5.2 Joint target-bit allocation

In this section we consider the problem of finding a RD allocation strategy, which may jointly optimize the resulting video quality and the required bitrate sharing between the texture and depth video sequences.

To this end, Morvan *et al.* in [88] within H.264/MPEG-4 AVC propose combining the texture and depth RD curves to obtain a single RD surface that allows the optimization of a joint texture/depth bitrate allocation problem. However the proposed joint bitrate allocation problem is in relation to the obtained rendering quality, which requires the *a priori* knowledge of the synthesis view at the encoder, which can not always hold. An attempt to overcome this issue was proposed by Liu *et al.* in [69] passing through a view synthesis distortion model. The proposed distortion model is an additive model that accounts for the texture-coding-induced distortion and the depth-quantization-induced distortion. On the other hand, Maitre *et al.* propose in [72] a wavelet-based RD framework to jointly and automatically optimize the bitrate allocation by using a dynamic programming along the tree of the wavelet coefficients.

Since the ratio between the texture and the depth bitrate is still an open question, we propose a bit allocation algorithm, based on wavelet subband bit allocation [120], resulting in taking into account the ratio of the variances of the pictures in the video-plus-depth sequence. For the predictive P-pictures and the bi-directional B-pictures, this variance is computed for the displaced frame difference (DFD), defined as:

$$\Delta I_t(x,y) = I_t(x,y) - I_{t-1}(x+v_x, y+v_y), \tag{5.4}$$

with $\mathbf{v}(v_x, v_y)$ being the MV which minimizes the MSE measure defined in (5.3). The variance of this DFD is given by:

$$\sigma^2 = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\left[\Delta I_t(x,y) - \overline{\Delta I_t}\right]^2 \tag{5.5}$$

where $\overline{\Delta I_t}$ denotes its average value, *i.e.*

$$\overline{\Delta I_t} = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\Delta I_t(x,y). \tag{5.6}$$

5.2.1 Log-variance approach

Let us first recall that according to the rate-control algorithm Test Model 5 (TM5) [54] recommended in MPEG-2 standard, the number of bits T allocated in a GOP is shared between the pictures inside the GOP such that:

$$T \approx N_I T_I + N_P T_P + N_B T_B \tag{5.7}$$

where T_I, T_P and T_B are the target-bits (see Section 2.2.2) for respectively the I-pictures, P-pictures, B-pictures, and N_I, N_P and N_B are respectively the number of I-pictures, P-pictures, B-pictures inside the GOP.

As shorthand, we will use the notation T to denote one of T_I, T_P and T_B when they is non risk for confusion. In the following we present the proposed target-bit allocation strategy that shares out the global number of bits T between the texture and depth data.

Statement Let us consider the problem of jointly coding a texture-plus-depth picture at a maximum total target-bits T and with a minimum overall reconstruction error D. The goal of the joint target-bit allocation is to estimate the optimal target-budgets $(\widehat{T}_t, \widehat{T}_d)$ for encoding the texture and depth pictures, such that the overall reconstruction error D of the texture and the depth pictures is minimized. Consequently, the optimization problem can be formulated as follows:

$$\left(\widehat{T}_t, \widehat{T}_d\right) = \underset{T_t, T_d}{\arg\min}\left\{D(T_t, T_d)\right\}, \tag{5.8}$$

where T_t and T_d are respectively the target-bits associated with the texture and depth pictures.

We use the following RD model at high resolution [110]:

$$D(T) = a\sigma^2 2^{-2T}, \tag{5.9}$$

where a is a constant depending on the distribution of the source. Therefore, one can write the joint distortion as:

$$\begin{aligned} D(T_t, T_d) &= D_t(T_t) + D_d(T_d) \\ &= a_t\sigma_t^2 2^{-2T_t} + a_d\sigma_d^2 2^{-2T_d} \end{aligned} \tag{5.10}$$

where a_t, a_d are constants associated with the distribution of the texture and depth pictures, σ_t^2 and σ_d^2 are respectively the variance of the texture and depth pictures.

Lagrangian multipliers We propose using the Lagrangian multiplier techniques for the optimization problem of minimization of the overall reconstruction error $D(T_t, T_d)$ subject to the following constraint of a given total budget target-bits:

$$T = T_t + T_d, \tag{5.11}$$

Hence, finding the optimal rate allocation between the texture and the depth videos can be formulated by minimizing the criterion J expressed as:

$$J\big(T_t, T_d, \lambda\big) = \big(a_t\sigma_t^2 2^{-2T_t} + a_d\sigma_d^2 2^{-2T_d}\big) + \lambda \cdot \big(T_t + T_d - T\big), \tag{5.12}$$

The simultaneous equations for the solution are:

$$\frac{\partial J}{\partial T_t} = 0 = -2a_t\sigma_t^2 2^{-2T_t}\ln(2) + \lambda \tag{5.13a}$$

$$\frac{\partial J}{\partial T_d} = 0 = -2a_d\sigma_d^2 2^{-2T_d}\ln(2) + \lambda \tag{5.13b}$$

$$\frac{\partial J}{\partial \lambda} = 0 = T_t + T_d - T \tag{5.13c}$$

After solving Eqs. (5.13a) and Eq. (5.13b), the solution verifies:

$$\widehat{T}_t - \widehat{T}_d = \frac{1}{2}\log_2\left(\frac{a_t\sigma_t^2}{a_d\sigma_d^2}\right). \tag{5.14}$$

To complete the solution with the global bitrate T, we substitute Eq. (5.14) into the constraint in Eq. (5.13c), to get the following analytical formulations:

$$\widehat{T}_t = \frac{T}{2} + \frac{1}{4}\log_2\left(\frac{a_t\sigma_t^2}{a_d\sigma_d^2}\right), \tag{5.15a}$$

$$\widehat{T}_d = \frac{T}{2} - \frac{1}{4}\log_2\left(\frac{a_t\sigma_t^2}{a_d\sigma_d^2}\right), \tag{5.15b}$$

$$\lambda = a_d\sigma_d^2\sqrt{\frac{a_t\sigma_t^2}{a_d\sigma_d^2}}\ln(2)2^{1-T}. \tag{5.15c}$$

The needed bitrate to encode each stream is function of the global bitrate R and the variance of the composing streams, texture and depth video sequences. Thus, with the variance of a picture defined in Eq. (5.5), we can estimate the average number of bits allocated for each stream composing the global video-plus-depth stream.

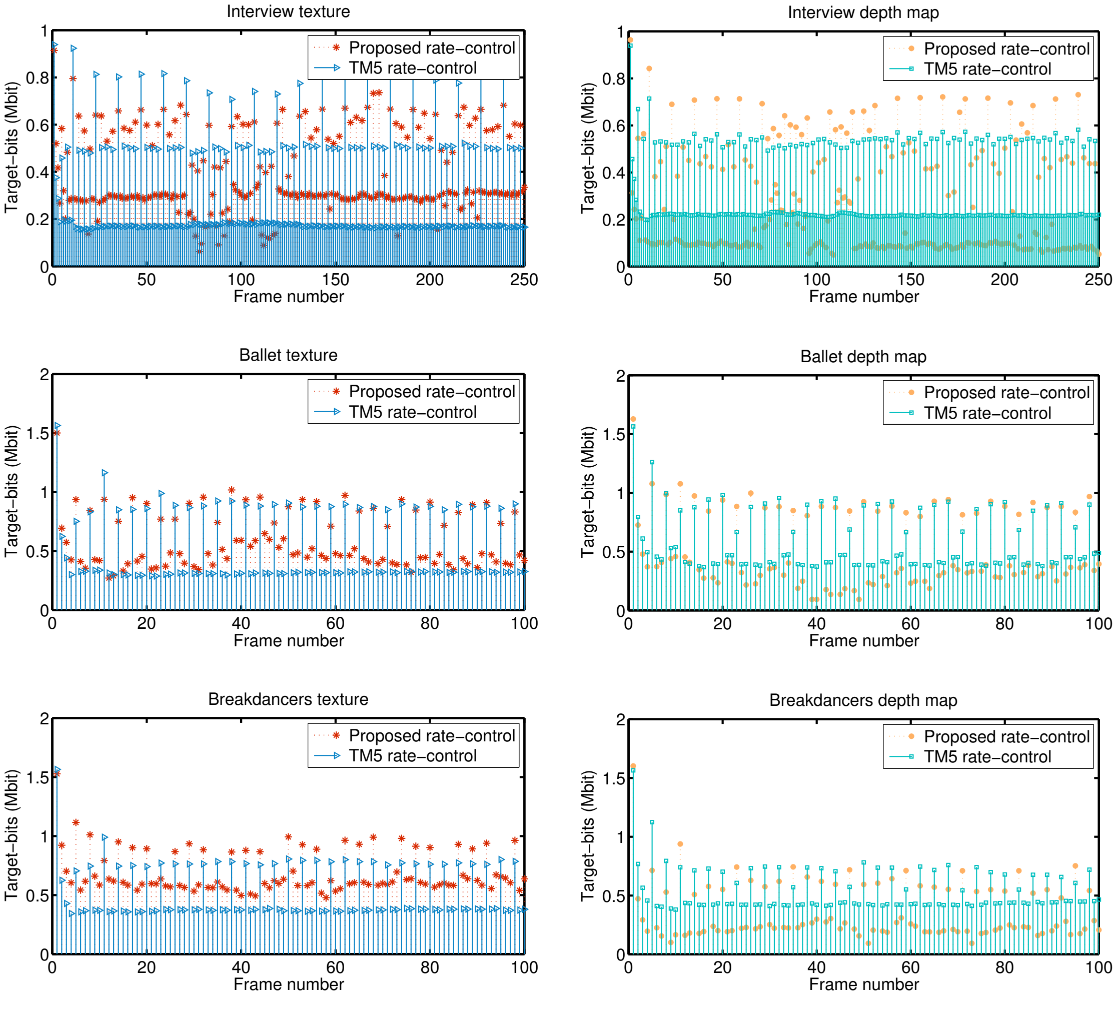
5.2.2 Target-bit analysis

As defined in Eqs. (5.15), Table 5.1 shows the average variance ratio between the texture and depth video sequences for each type of picture in a GOP. The results highlight a higher average variance of the texture, corresponding to a higher motion activity, which leads to allocate more bits to the texture stream than to the depth stream for predictive P-pictures and B-pictures, and the inverse in the case of I-pictures.

By applying Eqs. (5.15), the proposed distribution of the available bits is then applied between the texture and the depth pictures as illustrated in Fig. 5.7, according to the motion in the scene.

Table 5.1: Average variance ratio between the texture and depth videos.

Sequence	σ_t^2/σ_d^2		
	I	P	B
"Interview"	0.713	3.072	3.735
"Ballet"	0.660	1.045	3.828
"Breakdancers"	0.672	3.926	4.269



(a) Texture

(b) Depth map

Figure 5.7: Target-bit re-allocation with our strategy compared with the TM5 rate-algorithm at target bitrate of 8Mbps for the texture and the depth videos.

Notice that in Eq. (5.15), the term $\frac{T}{2}$ represents the average target-budget in bits for a texture-plus-depth picture, where T is computed using the distribution of bits inside a GOP as defined in Eq. (5.7). We propose starting from the low time consuming TM5 target-budget, which relies on the number of pictures of a certain type and their relative complexity measure in a GOP as described in Section 2.2.2, and to adjust it according to the log-variance term, leading to the new distribution shown in Fig. 5.8.

Hence, Eq. (5.15) is updated as follows:

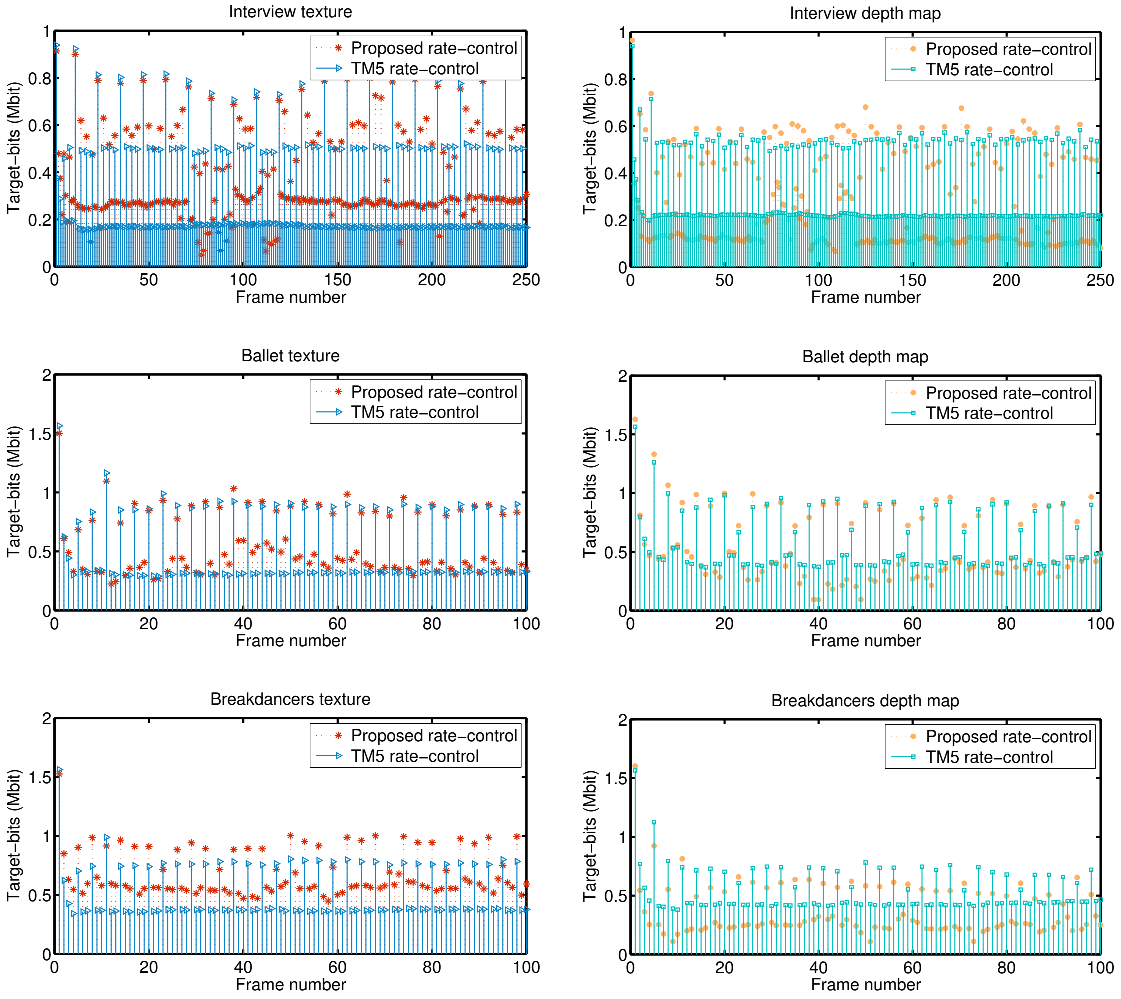
$$\widehat{T}_t = T_t^{\mathrm{TM5}} + \frac{1}{4}\log_2\left(\frac{a_t\sigma_t^2}{a_d\sigma_d^2}\right), \tag{5.16a}$$

$$\widehat{T}_d = T_d^{\mathrm{TM5}} - \frac{1}{4}\log_2\left(\frac{a_t\sigma_t^2}{a_d\sigma_d^2}\right), \tag{5.16b}$$

$$\tag{5.16c}$$

where T_t^{TM5} and T_d^{TM5} are the TM5 target-bits used as initialization respectively for the texture and depth pictures.

Hence, using the TM5 target-bits as initialization, allows us to take into the difference between $(\frac{T}{2}, \frac{T}{2})$ and $(T_t^{\mathrm{TM5}}, T_d^{\mathrm{TM5}})$. Although for DFD these values are similar, for *intra* encoded I-pictures the difference becomes larger.



(a) Texture (b) Depth map

Figure 5.8: Target-bit re-allocation with our strategy compared with the TM5 rate-algorithm at target bitrate of 8Mbps for the texture and the depth videos, using TM5 target-bits as initialization.

5.3 Experimental results and discussion

Our experiments evaluate the proposed motion estimation and bitrate allocation methods on three different video-plus-depth sequences. One of them, "Interview" (720×576, 25fps), is from the ATTEST project [38], and the two others, "Ballet"and "Breakdancers" (1024×768, 15fps), were single views chosen from the multiview datasets produced by Microsoft Research [7]. The depth videos of the ATTEST sequences had lower contrast but more detail than the Microsoft sequences.

The depth videos provided by Microsoft Research have been computed using a stereo matching algorithm [143]. The ATTEST ones have been captured directly from the so-called Zcam$^{\text{TM}}$ camera [53].

According to the MPEG-C Part 3 specifications, and under constraints that the same encoder is used as well for the texture and depth videos, the experiments have been done with the MPEG-2 reference software. An IBBP GOP of 12 pictures was used for the configuration of the codec.

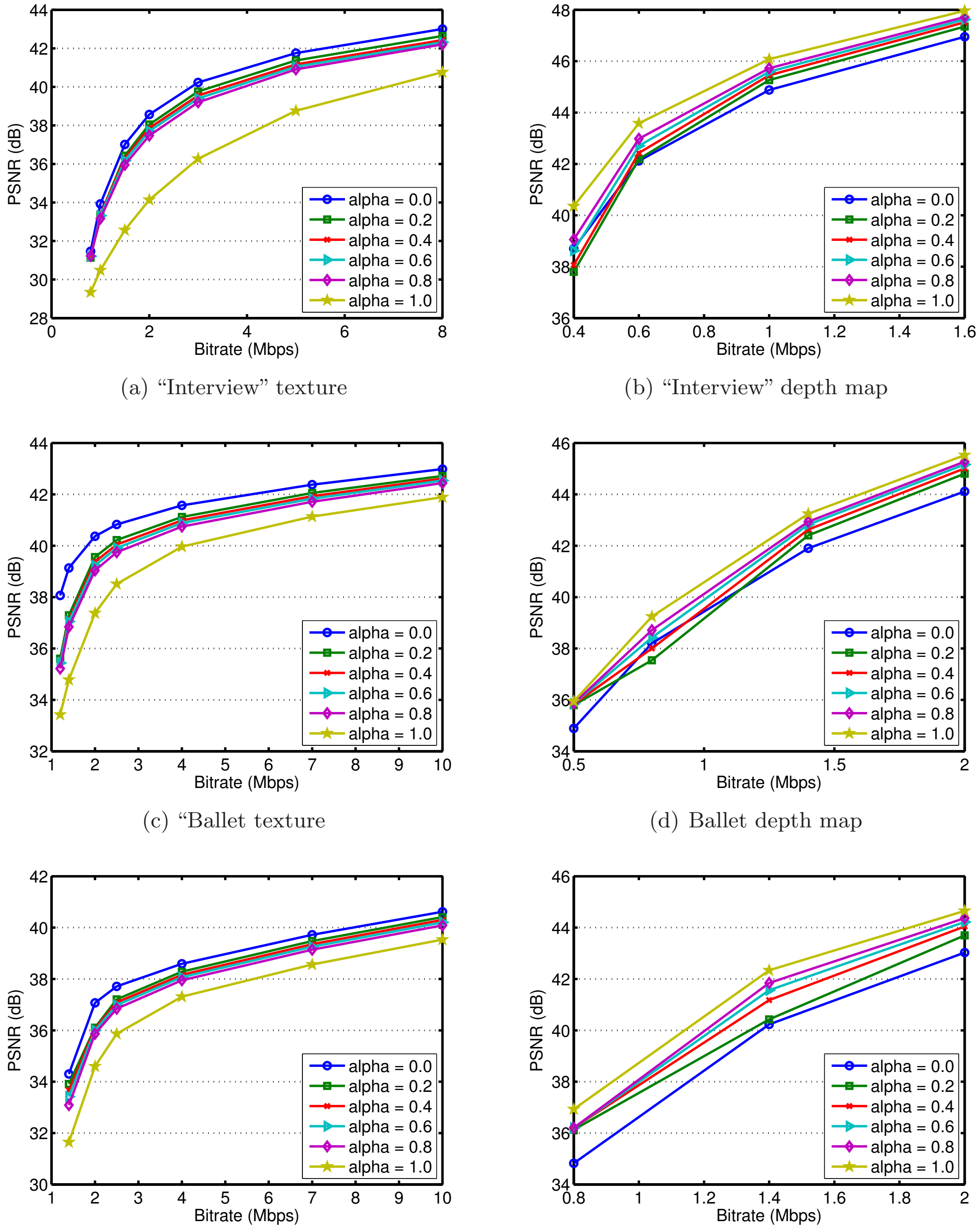
One of the various MPEG-2 industrial applications can be the storage on DVD support or the transmission over the digital broadcast using the DVB standard. The used bitrate has to satisfy at least the quality and the resolution of the picture for which an average viewer does not perceive any compression artifacts. Firstly in the DVD case, considering an SD resolution (720×576) at 25 fps, the bitrate is between 4 Mbps and 8 Mbps, that is, 0.39 bpp and 0.77 bpp. Still in SD resolution, the digital television channels are transmitted using mostly a bitrate between 2 Mbps and 8 Mbps, that is, 0.19 bpp and 0.77 bpp [5]. According to these values, the test sequences are encoded, according to their own resolution and frame rate, in respect of the bitrate range used in digital content industry.

Fig. 5.9 shows the PSNR of the texture and depth videos when the parameter α varies between 0 (when the depth data is not considered for motion estimation) and 1 (motion estimation is based only on the depth data). One can remark a sensible improvement of the depth video reconstruction (more than 1dB), for a small reduction in the texture video quality (between 0.4-0.8 dB), when using the joint estimation criterion.

In order to find the optimal value of α for each test sequence, we tune the parameter and provide PSNR analysis of the reconstructed (virtual) sequence as illustrated in Fig. 5.10. The depth video bitrate is arbitrarily fixed to 20% to the texture bitrate. The curves highlight a value close to $\alpha = 0.2$, $\alpha = 0.0$ and $\alpha = 0.6$ as the best value for respectively the video sequence "Ballet", "Breakdancers" and "Interview". This shows that estimating the MVs only on the texture video does not lead to the best reconstruction of the virtual sequence, and the proposed trade-off can largely improve the results.

Considering the depth video bitrate equal to 20% of the texture video bitrate, the Fig. 5.11 shows the PSNR of the resulting "virtual" sequence. The joint motion estimation has been coupled with the new bitrate allocation. The results show better performance at high bitrate (between 0.5-1.5 dB) for a small reduction at low bitrate (between 0.2-1 dB).

Since 3D perception depends heavily on the stereoscopic vision of two sequences, the transmitted texture video and the reconstructed "virtual" sequence, it is difficult to evaluate the 3D perceived quality only by means of an objective evaluation model like the PSNR. Thus an additional validation is proposed through an subjective evaluation. For this, the perceived quality and the depth perception are conducted using the Double



(a) “Interview” texture

(b) “Interview” depth map

(c) “Ballet texture

(d) Ballet depth map

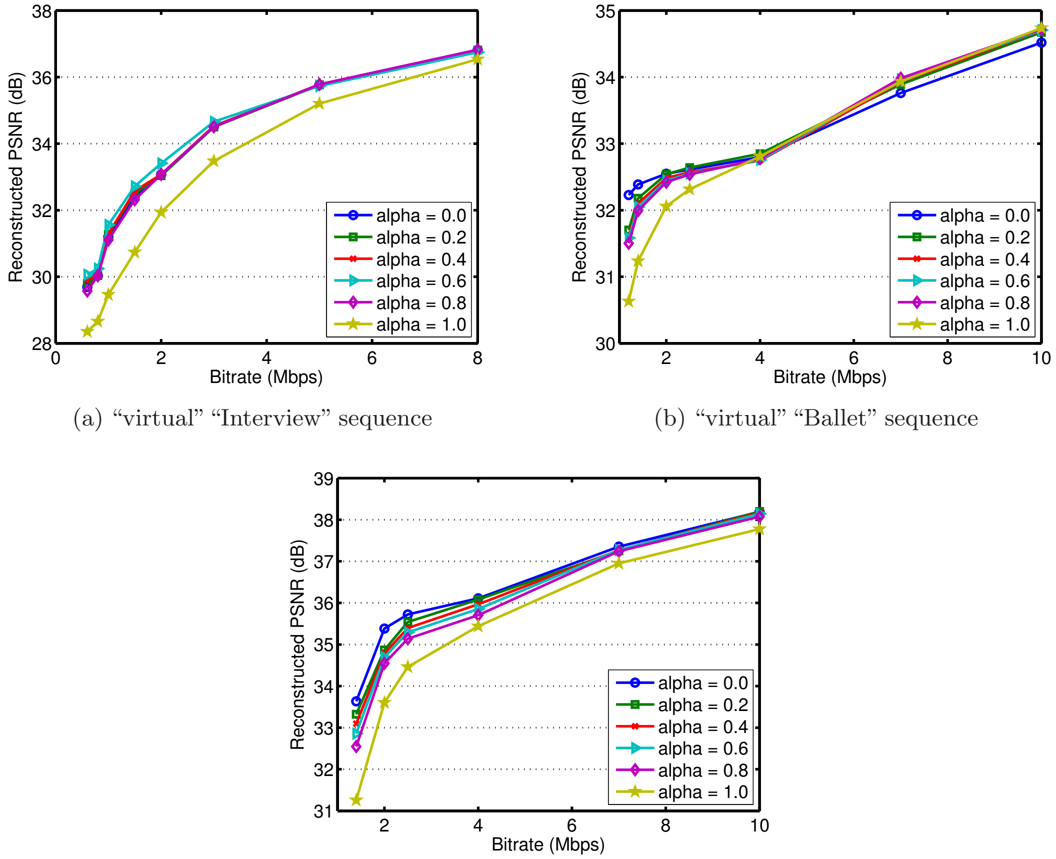
(e) “Breakdancers texture

(f) “Breakdancers” depth map

Figure 5.9: PSNR comparison with a joint MSE, for a variable parameter $\alpha \in [0, 1]$.

Stimulus Continuous Quality Scale Method (DSCQS) test methodology [1]. Non-experts

[1] The DSCQS method is especially useful when it is not possible to provide test stimulus that exhibit the full range of quality. Thus, it has been widely applied for evaluating high-quality TV pictures.The subjects are presented with a series of pairs of pictures or sequences in random order and with random impairments covering all required combinations, each from the same source, but one via the process under examination, and the other one directly from the source. For instance, the DSCQS method is claimed to



(a) "virtual" "Interview" sequence

(b) "virtual" "Ballet" sequence

(c) "virtual" "Breakdancers sequence

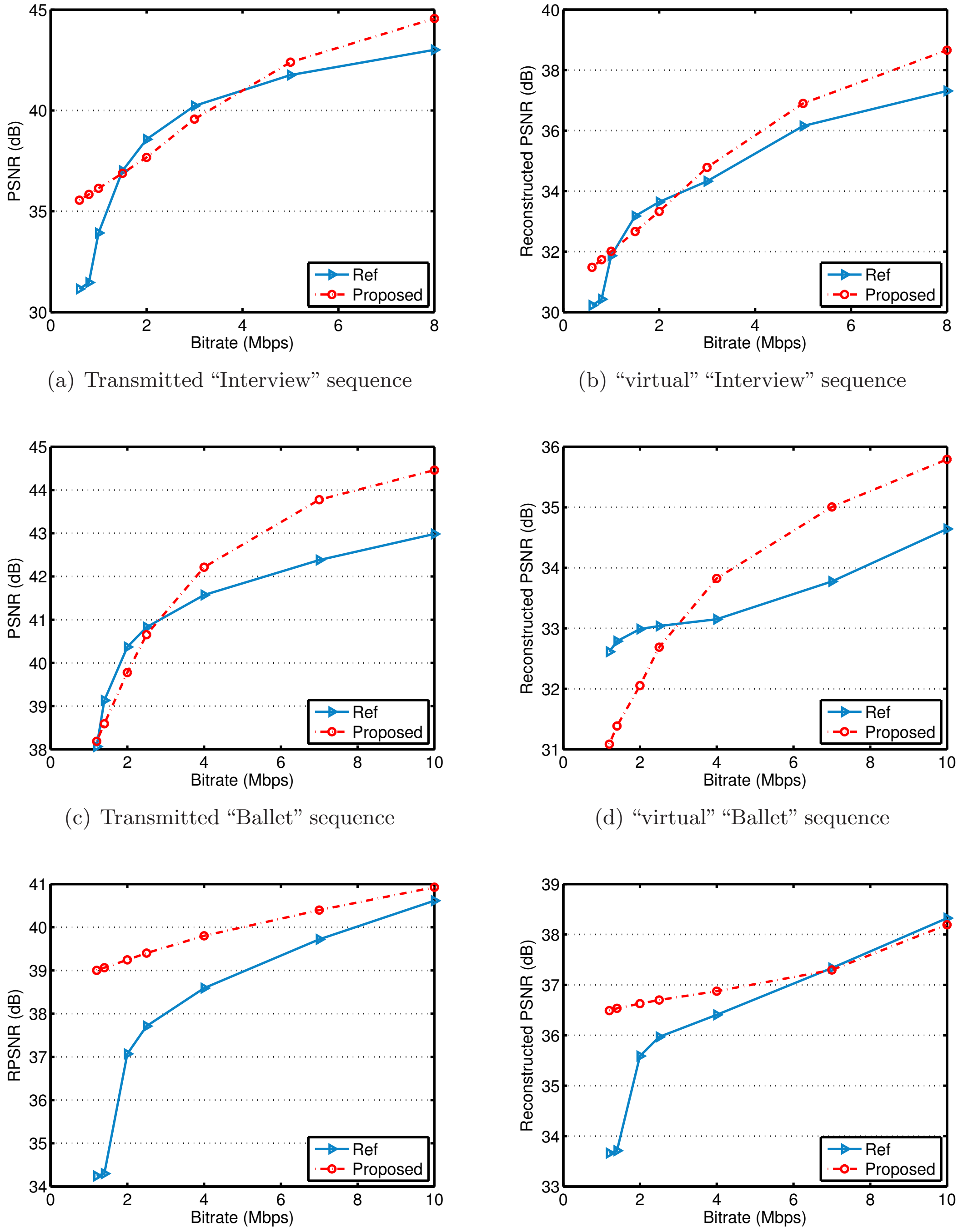
Figure 5.10: Search of the optimal α by comparison of the reconstructed "virtual" video PSNR. The depth video bitrate is equal to 20% of the texture video bitrate.

and inexperienced accessors have given their opinion of the video quality and the video depth perception. The experiment gathers 15 accessors using an autostereoscopic Sharp LL-151-3D LCD Monitor. Average Mean Opinion Score (MOS), according to the ITU-R Recommendation BT.500-10 [56], is given in Table 5.2. The obtained results confirm the objective results. It is shown an overall amelioration with the proposed method compared to the conventional MPEG-2 bitrate allocation with an advantageous amelioration at high bitrate.

5.4 Conclusion

In this chapter, we presented a novel method for the coding the video-plus-depth data by means of a joint estimation of the MV fields for the texture and depth video sequences in order to send only one MV field for both the texture and depth streams. According to the MPEG-C Part 3 specifications, the joint MV field is placed in the MPEG-2 texture stream for backward-compatibility purposes. The experimental results point out that optimizing only the texture MVs does not lead necessarily to the best view synthesis.

be less sensitive to context (i.e., subjective ratings are less influenced by the severity and ordering of the impairments within the test session).



(a) Transmitted “Interview” sequence

(b) “virtual” “Interview” sequence

(c) Transmitted “Ballet” sequence

(d) “virtual” “Ballet” sequence

(e) Transmitted “Breakdancers” sequence

(f) “virtual” “Breakdancers” sequence

Figure 5.11: Resulting reconstructed PSNR of the “virtual” video using the new bitrate allocation with the PSNR of the other stereo view. Depth video bitrate equals 20% of the texture video bitrate.

Moreover, the benefit of having a correlation between the MVs transmitted in texture and depth streams, can be utilize for error resilient transmission of a video-plus-depth sequence, where in case of lost informations in one stream, one can extract them from the

Table 5.2: Average MOS provided numerical indication of the perceived quality. MOS value is expresses between 1 to 5, where 1 refers to lowest quality, and 5 to a highest quality.

Sequence	**reference method**	**proposed method**
“Interview” (1.5 Mbps)	**1.4**	1.3
“Interview” (5 Mbps)	3.8	**3.9**
“Ballet” (2.5 Mbps)	**2.7**	2.4
“Ballet” (7 Mbps)	3.6	**4.2**
“Breakdancers” (2.5 Mbps)	2.8	**3.4**
“Breakdancers” (7 Mbps)	4.1	**4.2**

other.

On the other hand, we developed a joint bit allocation strategy between the texture and depth streams based on a high bitrate distortion model. The proposed rate control algorithm takes into account the motion activity of each texture and depth pictures by calculating the log-variance ratio. The new bitrate allocation allows a better distribution and largely improves the results.

Despite a low complexity in our bit allocation strategy that is function of the pictures variances, this approach suffers from the lack of the distortion model. It is well known that there is a mismatch between the theoretical formula and the actual RD curve at low bitrates, which is visible in our experimental results. Instead of using an explicit distortion analytical function, one can pursue the optimal solution through using the generalized Breiman, Freidman, Olshen, and Stone (BFOS) algorithm firstly introduced in the context of classification and regression trees [15] and later in source coding [25]. The algorithm begins with an initial tree involving assigning a maximum of bits to each source, then deallocating the bits in order to reach a given RD trade-off.

Chapter 6

H.264-based dense motion/disparity estimation in MVC

Contents

In this chapter, we propose a dense motion/disparity estimation algorithm, designed to replace the classical temporal/inter-view unit within the multiview video coding (MVC) extension of H.264/MPEG-4 AVC, which uses a block-based motion/disparity estimation. We focus on the sequential prediction structure of the employed MVC encoder scheme [8].

Initially introduced by Combettes [28] in image denoising and image restoration, and after resumed by Miled [82] in the problem of stereo matching and its application to road obstacle detection from a stereo vision system, the dense disparity estimation problem has been formulated as a convex programming problem within a global variational framework. Numerical studies have shown that variational-based disparity estimation methods are among the most powerful techniques meanwhile preserving the depth discontinuities. A quantitative comparison with results from other stereo algorithms available at the Middlebury website[1] shows that this approach is competitive with state-of-the-art methods, such as graph cuts [14, 63] and belief propagation based methods [111]. This naturally motivates our choice to integrate this global convex variational framework, along with the motion and disparity estimation, within a multiview video coder.

In the first part, we provide details about the retained variational convex optimization approach that generates smooth displacement vector fields with ideally infinite precision. The objective function to optimize is however different, taking into account the structure and coding order in a multiview codec. The estimation problem is solved through the minimization of a global objective function, which is the sum of Displaced Frame Differences (DFD), under various convex constraints.

In the second part, we study the influence of the parameters used in the convex programming problem. Then, we address the problem of coding the resulting *dense* motion and disparity vectors which is a challenging issue because of the high bitrate needed to transmit such fields. We propose reducing the bitrate needed for the coding of the dense displacement fields by performing an RD segmentation and coding. This is achieved by optimizing a Lagrangian cost function which takes into account the accuracy and the coding cost of the displacement field. The dense estimation framework followed by the segmentation step, replace therefore the block-based motion/disparity estimation stage in the MVC extension.

In the third part, we introduce in the non-base views, the dense disparity prediction of the temporal *intra* pictures. Since disparity fields vary smoothly in homogeneous regions and change abruptly around object boundaries, we use edge preserving regularizing constraints based on the Total Variation measure, which has already proved to be very useful in image recovery and denoising problems [100].

In the fourth part, the concept is extended to the temporal *inter* pictures, which use both temporal and inter-view reference pictures. The motion and disparity fields are therefore jointly and simultaneously derived using the *stereo-motion consistency* constraint in the same proposed set theoretic convex optimization framework.

6.1 A variational convex optimization approach

The retained variational approach belongs to the global variational methods which are based on the minimization of a global energy functional $E(\mathbf{u})$ on the whole image, which consists of a data term and a regularization term:

$$E(\mathbf{u}) = E_{\text{data}}(\mathbf{u}) + \lambda \cdot E_{\text{smooth}}(\mathbf{u}) \tag{6.1}$$

[1] http://vision.middlebury.edu/stereo/

where $\mathbf{u}$ denotes the displacement (*i.e.* motion or disparity) vector fields to be estimated and λ is the Lagrange parameter that weights the smoothness term relatively to the first data term.

Based on a set theoretic framework, this motion/disparity approach may incorporate various convex constraints corresponding to *a priori* information such as the range of displacement vectors or the total variation regularization constraint which assures smooth disparity fields while preserving discontinuities, as we shall address next.

6.1.1 Problem statement

Let us first introduce some notations that we use in the remainder of this chapter for the representation of:

- a pixel intensity: $I(s) = I(x, y)$,
- a compensated pixel intensity: $I(s + \mathbf{u}(s)) = I(x + u_x(x, y), y + u_y(x, y))$,
- a compensated picture: $I(\mathbf{u}) = \sum_{s \in \mathcal{D}} I(s + \mathbf{u}(s))$,

where s represents the pixel location (x, y), and $\mathbf{u}(u_x, u_y)$ is a displacement (motion or disparity) vector field. Note that in the rest of this chapter, we will not always make explicit that $\mathbf{u}(s)$ or its components $\big(u_x(s), u_y(s)\big)$ are functions of s for notation concision.

Consider I_{curr} and I_{ref} being two temporal (or inter-view) consecutive pictures taken respectively by the n^{th} camera at time t and $t - 1$ (or the n^{th} and $(n-1)^{\text{th}}$ cameras at time t separated by a fixed baseline).

Dense estimation methods attempt to determine, for each pixel in the current picture I_{curr}, the best corresponding pixel in the reference picture I_{ref}. Generally, the estimation is obtained by minimizing a given cost functional J, formulated in terms of the sum of squared differences (SSD):

$$\begin{aligned} \hat{\mathbf{u}} = \big(\widehat{u}_x, \widehat{u}_y\big) &= \arg\min_{\mathbf{u} \in \Omega^2} J(\mathbf{u}) \\ \text{with} \quad J(\mathbf{u}) &= \sum_{s \in \mathcal{D}} \big[I_{\text{curr}}(s) - I_{\text{ref}}(s + \mathbf{u})\big]^2 \end{aligned} \tag{6.2}$$

where $\mathcal{D} \subset \mathbb{N}^2$ is the picture support and Ω^2 is the range of candidate vectors. The pixel displacement is denoted by the vector $\mathbf{u}(u_x, u_y)$. The cost functional J may differ according to the problem statement as we will address in the next sections.

Generally, an initial estimate $\bar{\mathbf{u}}(\overline{u}_x, \overline{u}_y)$ of $\mathbf{u}(u_x, u_y)$ is available, for example using a dense correlation-based method. In practice, the initial displacement field $\bar{\mathbf{u}}$ can be estimated by local approaches. In this work we utilize a dense block matching method using the SSD function to estimate $\bar{\mathbf{u}}$.

Assuming that the magnitude difference of both fields is relatively small, the compensated reference picture is approximated around $\bar{\mathbf{u}}$ by a Taylor expansion:

$$I_{\text{ref}}(s + \mathbf{u}) \simeq I_{\text{ref}}(s + \bar{\mathbf{u}}) + (\mathbf{u} - \bar{\mathbf{u}}) \cdot \nabla I_{\text{ref}}(s + \bar{\mathbf{u}}) \tag{6.3}$$

where ∇ is the gradient operator applied on the compensated reference picture $I_{\text{ref}}(\bar{\mathbf{u}})$. Using the linearization in Eq. (6.3), the criterion J in Eq. (6.2) can be approximated by a quadratic convex functional $\widetilde{J}$ in $\mathbf{u}$ such that:

$$\widetilde{J}(\mathbf{u}) = \sum_{s \in \mathcal{D}} \left[\mathbf{L}(s) \cdot \mathbf{u}^{\top} - \mathbf{r}(s)\right]^2, \tag{6.4}$$

where

$$\mathbf{L}(s) = \nabla I_{\text{ref}}(s + \bar{\mathbf{u}}) \quad \text{and} \quad \mathbf{r}(s) = I_{\text{curr}}(s) - I_{\text{ref}}(s + \bar{\mathbf{u}}) + \mathbf{L}(s) \cdot \bar{\mathbf{u}}^{\top}.$$

Since $L(s)$ may be zero, in the criterion $\widetilde{J}$ Eq. (6.4) an additive term has been introduced to make $\widetilde{J}$ strictly convex, in compliance with the assumption required to guarantee the convergence of the algorithm. Therefore, $\widetilde{J}$ becomes:

$$\widetilde{J}(\mathbf{u}) = \sum_{s \in \mathcal{D}} \left[\mathbf{L}(s) \cdot \mathbf{u}^{\top} - \mathbf{r}(s)\right]^2 + \alpha \cdot \sum_{s \in \mathcal{D}} \left[\mathbf{u} - \bar{\mathbf{u}}\right]^2 \tag{6.5}$$

where α is a positive real number: when it is large, it favors the regularization term and the final solution $\hat{\mathbf{u}}$ tends to be close to the initialization; on the contrary, when α is small, the data attachment term becomes dominant, and the solution can diverge from the initialization.

The minimization of this quadratic functional is an ill-posed problem, as the components of L may locally vanish. Thus, to convert this problem to a well-posed one, additional constraints have been incorporated, which reflect the prior knowledge about the displacement vector fields.

In this work, we address the problem through a set theoretic framework [81]. Firstly, each constraint is represented by a closed convex set S_m with $m \in \{1, \ldots, M\}$, in a Hilbert space $\mathcal{H}$. The intersection S of all the M sets S_m constitutes the family of possible solutions. Therefore, the constrained problem amounts to finding the solution in S which minimizes the functional $\widetilde{J}$ expressed as:

$$\text{Find } \mathbf{u} \in S = \bigcap_{m=1}^{M} S_m \quad \text{such that} \quad \widetilde{J}(\mathbf{u}) = \min_{\mathbf{u} \in S} \widetilde{J}(\mathbf{u}). \tag{6.6}$$

The constraint sets are modeled as level sets:

$$\forall m \in \{1, \ldots, M\}, \quad S_m = \{\mathbf{u} \in \mathcal{H} \mid f_m(\mathbf{u}) \leq \delta_m\} \tag{6.7}$$

where $f_m : \mathcal{H} \to \mathbb{R}$ is a continuous convex function for all $m \in \{1, \ldots, M\}$ and $(\delta_m)_{1 \leq m \leq M}$ are real-valued parameters such that $S = \bigcap_{m=1}^{M} S_m \neq \emptyset$.

Hence, it is required to define the convex sets S_m to proceed to the motion/disparity estimation algorithm within the set theoretic framework. At this level, it is important to emphasize the great flexibility in incorporating any set of arbitrary convex constraints. In what follows, we will focus on $M = 4$ constraints. The two first ones consist of restricting the variation of the vector components (u_x, u_y) within a specified range $[u_{x\min}, u_{x\max}]$ and $[u_{y_{\min}}, u_{y_{\max}}]$. This *a priori* can be expressed by the following constraint sets:

$$\begin{aligned} S_1 &= \{u_x \in \mathcal{H} \mid u_{x\min} \leq u_x \leq u_{x\max}\}, \\ S_2 &= \{u_y \in \mathcal{H} \mid u_{y_{\min}} \leq u_y \leq u_{y_{\max}}\}. \end{aligned} \tag{6.8}$$

Most importantly, a constraint can be incorporated in order to strengthen the smoothness of the vector fields in the homogeneous areas while preserving edges. Indeed, neighboring pixels belonging to the same object should have similar values. This can be achieved by considering the total variation $\mathsf{tv}(u_x)$ and $\mathsf{tv}(u_y)$ which can be defined as the sum over $\mathcal{D}$ of the norm of the spatial gradient of each direction u_x and u_y [100]. The total variation of the discrete horizontal displacement image $u_x = [{u_x}^{i,j}]$ is given by:

$$\begin{aligned}\mathsf{tv}(u_x) = &\sum_{i=1}^{W-1}\sum_{j=1}^{H-1}\sqrt{|{u_x}^{i+1,j} - {u_x}^{i,j}|^2 + |{u_x}^{i,j+1} - {u_x}^{i,j}|^2}\\ &+ \sum_{i=1}^{W-1}\sqrt{|{u_x}^{i+1,H} - {u_x}^{i,H}|}\\ &+ \sum_{j=1}^{H-1}\sqrt{|{u_x}^{W,j+1} - {u_x}^{W,j}|}\end{aligned} \tag{6.9}$$

where $W \times H$ is the support of the displacement image. And *vice versa*, for the total variation of the discrete vertical displacement image $u_y = [{u_y}^{i,j}]$.

Hence, a total variation based regularization constraint amounts to impose upper bounds τ_x and τ_y on the tv of the image, in each direction, leading to the following constraint sets:

$$\begin{aligned}S_3 &= \{u_x \in \mathcal{H} \mid \mathsf{tv}(u_x) \leq \tau_x\},\\ S_4 &= \{u_y \in \mathcal{H} \mid \mathsf{tv}(u_y) \leq \tau_y\}.\end{aligned} \tag{6.10}$$

It is worth pointing out that the positive constants τ_x and τ_y can be estimated for example through a learning procedure on video databases [29]. In our case, we choose the values of these thresholds such that to maximize the quality of the reference compensated picture, as shown in Section 6.2.1.

Finally, the motion/disparity estimation problem is formulated by minimizing the quadratic objective function $\widetilde{J}$ in Eq. (6.4) under the mentioned constraint sets. To solve this problem, we employ a parallel block-iterative algorithm using subgradient projections on the convex constraint sets and based on recently developed convex analysis tools [28].

6.1.2 Optimization algorithm

Here, we first recall some essential facts on convex analysis, which are necessary for our minimization problem. More details can be found in [28]. The image space is the real Hilbert space $\mathcal{H}$ with scalar product $\langle . \mid . \rangle$ and norm $\| \, . \, \|$. Let S_m be the nonempty closed and convex subset of $\mathcal{H}$ given by Eq. (6.7), where f_m is a continuous and convex function. For every $\mathbf{u} \in \mathcal{H}$, f_m possesses at least one subgradient at $\mathbf{u}$, *i.e.*, a vector $\mathbf{g}_m \in \mathcal{H}$ such that

$$\forall \mathbf{z} \in \mathcal{H}, \quad \langle \mathbf{z} - \mathbf{u} \mid \mathbf{g}_m \rangle + f_m(\mathbf{u}) \leq f_m(\mathbf{z}). \tag{6.11}$$

The set of all subgradients of f_m at $\mathbf{u}$ is the subdifferential of f_m at $\mathbf{u}$ and it is denoted by $\partial f_m(\mathbf{u})$. If f_m is differentiable at $\mathbf{u}$, then $\partial f_m(\mathbf{u}) = \{\nabla f_m(\mathbf{u})\}$. Now, fix $\mathbf{u} \in \mathcal{H}$ and a subgradient $g_m \in \partial f_m(\mathbf{u})$. The subgradient projection $G_m\mathbf{u}$ of $\mathbf{u}$ onto S_m is given by:

$$G_m\mathbf{u} = \begin{cases} \mathbf{u} - \dfrac{f_m(\mathbf{u}) - \delta_m}{\|\mathbf{g}_m\|^2}\mathbf{g}_m & \text{, if} f_m(\mathbf{u}) > \delta_m \\ \mathbf{u} & \text{, otherwise.} \end{cases} \tag{6.12}$$

The proposed algorithm activates the constraints by means of subgradient projections rather than exact projections. The former are much easier to compute than the latter, as they require only the availability of a subgradient (the gradient in the differentiable case). However, when the projection is simple to compute, one can use it as a subgradient projection. In our case, exact projections onto $(S_m)_{1\leq m\leq 2}$ are straightforwardly obtained, whereas for the constraint sets $(S_m)_{3\leq m\leq 4}$, the expression of subgradient projections are given in [29].

We now proceed with the description of the proposed algorithm to estimate the displacement vector field $\mathbf{u}$. This algorithm starts from an initial point $\mathbf{u}_0$ and iteratively constructs a sequence $(\mathbf{u}_n)_{n\in\mathbb{N}}$, converging to the optimal solution $\hat{\mathbf{u}}$, as follows.

Initialization

① Take a nonempty index set $\mathbb{K} \subseteq \{1,\ldots,m\}$, where $\mathbb{K}$ defines the constraints.

② Set $\mathbf{u}_0 = (\mathbf{L}\cdot\mathbf{r}^\top + \alpha\cdot\bar{\mathbf{u}})\cdot(\mathbf{L}^2+\alpha\cdot\boldsymbol{I})^{-1}$ index.

Iteration n, for $n > 0$

③ For every index $i \in \mathbb{K}$, set $\mathbf{a}_{i,n} = \mathbf{P}_{i,n} - \mathbf{u}_n$, where $\mathbf{P}_{i,n}$ is a subgradient projection of $\mathbf{u}_n$ onto the solution S_i as in [28].

④ Set:

- $\mathbf{z}_n = \frac{1}{m}\cdot\sum_{i\in\mathbb{K}} \mathbf{a}_{i,n}$,
- $\kappa_n = \frac{1}{m}\sum_{i\in\mathbb{K}} \|\mathbf{a}_{i,n}\|^2$,

where m denotes the number of elements in $\mathbb{K}$.

⑤ If $\kappa_n = 0$, exit iteration. Otherwise set:

- $\mathbf{b}_n = \mathbf{u}_0 - \mathbf{u}_n$,
- $\mathbf{c}_n = (\mathbf{L}^2+\alpha\cdot\boldsymbol{I})\cdot\mathbf{b}_n$,
- $\mathbf{d}_n = (\mathbf{L}^2+\alpha\cdot\boldsymbol{I})^{-1}\cdot\mathbf{z}_n$,
- $\lambda_n = \kappa_n/\langle\mathbf{d}_n,\mathbf{z}_n\rangle$.

⑥ Set:

- $\tilde{\mathbf{d}}_n = \lambda_n\cdot\mathbf{d}_n$,
- $\pi_n = -\langle\mathbf{c}_n,\tilde{\mathbf{d}}_n\rangle$,
- $\mu_n = -\langle\mathbf{b}_n,\mathbf{c}_n\rangle$,
- $\nu_n = -\lambda_n\cdot\langle\tilde{\mathbf{d}}_n,\mathbf{z}_n\rangle$,
- $\rho_n = \mu_n\cdot\nu_n - \pi_n^2$.

⑦ Set:

$$u_{n+1} = \begin{cases} \mathbf{u}_n + \tilde{\mathbf{d}}_n, & \text{if} \quad \rho_n = 0, \quad \pi_n \geq 0, \\ \mathbf{u}_0 + (1+\pi_n/\nu_n)\cdot\tilde{\mathbf{d}}_n & \text{if} \quad \rho_n > 0, \quad \pi_n\nu_n \geq \rho_n, \\ \mathbf{u}_n + \frac{\nu_n}{\rho_n}\cdot(\pi_n\mathbf{b}_n + \mu_n\tilde{\mathbf{d}}_n) & \text{if} \quad \rho_n > 0, \quad \pi_n\nu_n < \rho_n. \end{cases}$$

⑧ Set $n = n+1$ and go to step ③.

As stop criterion, a difference magnitude observation of the current solution and the solution at the previous iteration may be used. Otherwise, a maximum number of iterations can be specified.

Note that, as proved in [81], if there exists a positive integer K such that

$$\forall n \in \mathbb{N}, \qquad \bigcup_{l=n}^{n+K-1} \mathbb{K}_l = \{1, \dots, M\}, \tag{6.13}$$

then every sequence $(\mathbf{u}_n)_{n \in \mathbb{N}}$ generated by Algorithm 1 converges to the unique solution of (6.6). Some comments about this convergence result and Algorithm 1 can be done at this point:

- Algorithm 1 allows to easily incorporate additional convex constraints if these are available. Its ability to use approximate (subgradient) projections onto the constraint sets makes it possible to handle a wide range of complex convex constraints.

- Due to its block iterative structure, this algorithm offers a lot of flexibility in terms of parallel implementation. Indeed, the set $\mathbb{K}_n$ defines the constraints to be activated at iteration n and, according to Eq. (6.13), different blocks of constraint sets may be used. Therefore, Algorithm 1 can be efficiently implemented on parallel computing architectures by adapting the number of elements in $\mathbb{K}_n$ to the number of available parallel processors.

- Concerning the computational complexity of this algorithm, the main computation at one iteration is the calculation of a subgradient for to the total variation constraint which can be estimated in $O(WH + W + H)$, where W and H are respectively the width and height of the image. Concerning the global computational cost, complexity can be estimated in $O(n(WH+W+H))$ where n is the number of iterations required for the convergence of the optimization algorithm.

In the next section, we shall be interested in representing and efficiently encoding the dense field issued from this algorithm.

6.2 Integration in the H.264 framework

6.2.1 Influence of the parameters

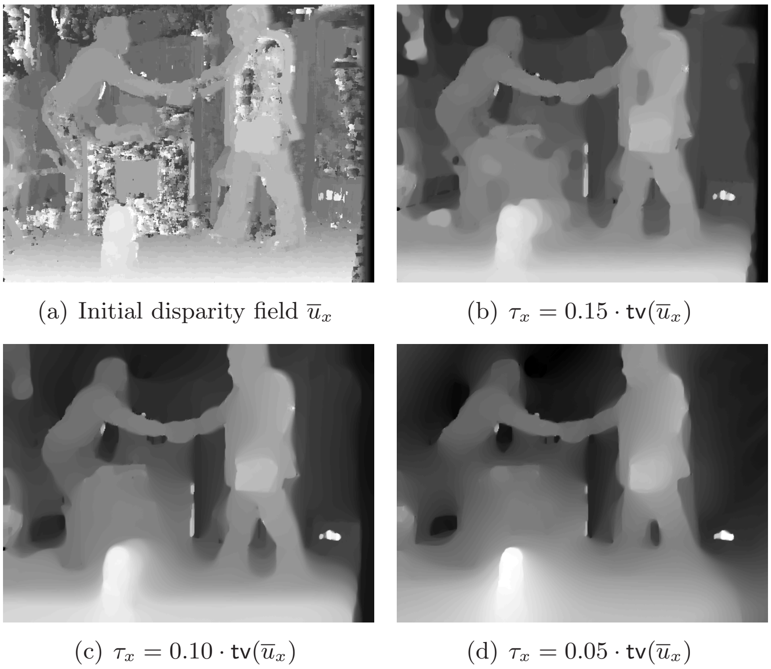
In practice, the optimal value of the parameters $[u_{x\text{min}}, u_{x\text{max}}, u_{y\text{min}}, u_{y\text{max}}]$, τ_x, τ_y and α may not be known exactly and it is, therefore, important to evaluate their impact in terms of coding rate and PSNR of the compensated picture.

In the remainder of this section, we consider only the inter-view displacement, and assume that the cameras are rectified, so that the vectors are restricted to the horizontal component.

Regularization constraint

The upper bound τ_x used to enforce the smoothness of the estimated vector component fields may be estimated from a scale value of the total variation of the initial fields $\overline{u}_x$, as shown in Fig. 6.1. A low scale value results in smoothing more the field, and so, reducing the number of bits required for the transmission.

The parameter α introduced in Eq. (6.5) has to be fixed in order to weight the influence of the additive term introduced to make $\widetilde{J}$ strictly convex. By choosing a high value of α, the criterion $\widetilde{J}$ simply reduces to the distance between $\mathbf{u}$ and $\bar{\mathbf{u}}$. Conversely, assigning a small value privileges the first data term.


(a) Initial disparity field $\overline{u}_x$

(b) $\tau_x = 0.15 \cdot \mathsf{tv}(\overline{u}_x)$

(c) $\tau_x = 0.10 \cdot \mathsf{tv}(\overline{u}_x)$

(d) $\tau_x = 0.05 \cdot \mathsf{tv}(\overline{u}_x)$

Figure 6.1: Example of dense disparity fields at different values of the upper bound τ_x parameter (from "Book arrival" sequence, picture 47, camera 3 and 5).

Table 6.1 and Table 6.2 show the impact of the parameters τ_x and α on the coding rate of the field and on the quality of the disparity compensated picture, evaluated as PSNR between the original view and the disparity-compensated estimation. First, an arbitrary fixed value of α is used to determine the parameter τ_x. Then, the optimal value of α is determined. The value of both parameters is selected according on the highest PSNR value of the disparity compensated picture.

Table 6.1: Example of the influence of the parameter τ_x with $\alpha = 6$ on the bitrate and the PSNR of the dense disparity compensated picture (from "Book arrival" sequence, picture 48).

	bitrate (H.264 intra, QP=0)	PSNR
$\tau_x = 50000$	0.6416 bpp	37.13 dB
$\tau_x = 40000$	0.5222 bpp	37.23 dB
$\tau_x = 30000$	0.4036 bpp	37.24 dB
$\tau_x = 20000$	0.3051 bpp	36.48 dB
$\tau_x = 10000$	0.2878 bpp	33.44 dB

Note that, in Table 6.1 and Table 6.2, the bitrate of the dense disparity field has been computed using the dense disparity field as a picture quantized on 8 bits with H.264/MPEG-4 AVC in *intra* mode at a QP value of 0.

Table 6.2: Example of the influence of the parameter α with $\tau_x = 30000$ on the bitrate and the PSNR of the dense disparity compensated picture (from "Book arrival" sequence, picture 48).

	bitrate (H.264 intra, QP=0)	PSNR
$\alpha = 0.1$	0.4667 bpp	35.79 dB
$\alpha = 6$	0.4036 bpp	37.24 dB
$\alpha = 10$	0.3943 bpp	37.20 dB
$\alpha = 50$	0.3905 bpp	36.50 dB
$\alpha = 100$	0.3837 bpp	35.78 dB

Range values

The choice of the ranges $[u_{x\min}, u_{x\max}]$ and $[u_{y_{\min}}, u_{y_{\max}}]$ can be accurately found by matching certain points of interest selected manually in two adjacent pictures.

When considering stationary cameras, the disparity ranges can be considered as fixed for every picture pair. For motion case, a generalization over the sequence is a difficult task. However, we can consider the idea that the inter-picture motion has low values.

Summary

For each multiview video sequence provided in [39] we have determined heuristically a set of parameters, and presented in Table 6.3.

However the parameters can be adjusted for each different sequence, by specifying their value in the Sequence Parameter Set (SPS). The increase in bitrate related to this side information is very small, such that we neglect this contribution when reporting experimental data about the coding rate.

Table 6.3: Parameter settings.

	disparity range	τ_x	α
"Book arrival"	[14, 35]	30000	6
"Door flowers"	[17, 34]	30000	6
"Outdoor"	[0, 8]	10000	10

6.2.2 Rate-distortion-based segmentation

However, coding the resulting dense motion and disparity vectors is a challenging issue because of the high bitrate needed to transmit such vectors. To overcome this problem, we propose reducing the coding cost by operating an RD-driven segmentation on the field. This is achieved by optimizing a Lagrangian cost function which takes into account the accuracy and the coding cost of the displacement field.

Moreover, the segmentation allows to adapt the obtained dense displacement fields into a block-based representation, as defined in the H.264/MPEG-4 AVC standard, and thus, to provide a bitstream compliant with MVC.

As illustrated in Fig. 6.2, we propose within the segmentation process an RD selection of the best displacement vector by partition inside a MB among all the reference pictures. In this work, reference pictures fall in two categories: the temporal ones involving the inter-picture neighboring for the motion prediction, and the inter-view ones, including the pictures from adjacent views for the disparity prediction.

The proposed scheme is summarized in Fig. 6.2: we replace the block-based displacement estimation stage by a dense displacement estimation one, and then, we apply an RD-based segmentation to the generated displacement vector fields. This is performed by optimizing a Lagrangian cost function which takes into account the accuracy and the coding cost of the displacement field.

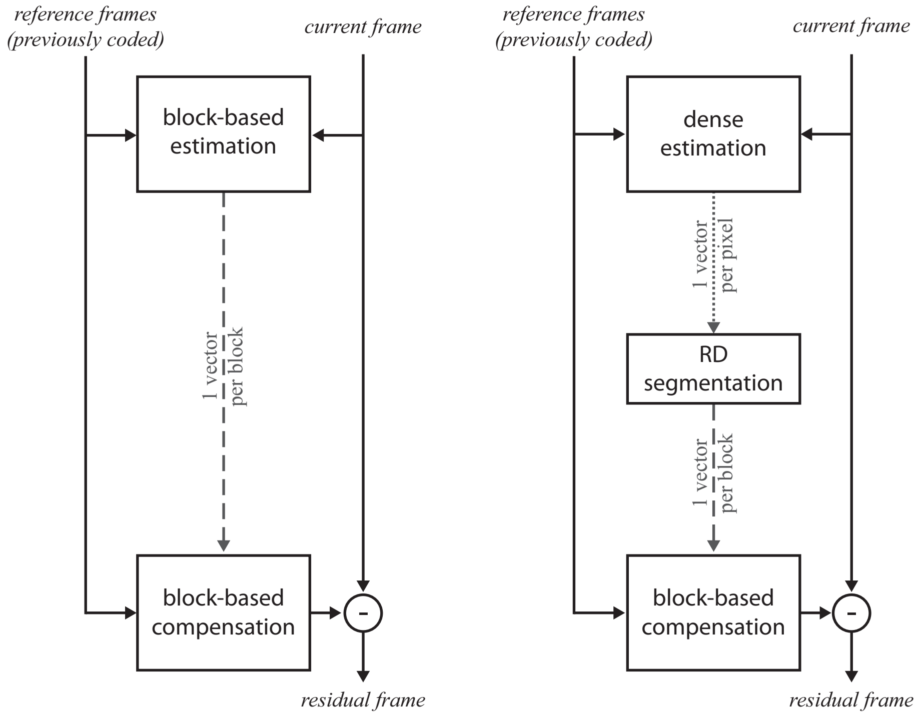


Figure 6.2: Motion/disparity prediction (left) in MVC, (right) in the proposed framework.

Block-based representation

Let us first recall that, as defined in the H.264/MPEG-4 AVC standard (see Section 2.3.2), a MB can support different partitions corresponding to 4 prediction modes (16×16, 8×16, 16×8, 8×8), which can have a different vector each. Finally in the 8×8 mode, the block can be split into 8×4, 4×8 and 4×4 sub-MB , which in turn have a single vector.

Partition-based segmentation

As mentioned earlier, the dense motion/disparity estimation method generates a field with real valued displacement vectors. A first approximation consists in truncating the precision at a quarter-pixel accuracy as in the H.264/MPEG-4 AVC standard. The accuracy conversion is performed by rounding the motion vector to the nearest quarter-pixel position. Then, one other approximation has to be made from a dense MB B_k, containing 256 displacement vectors (one vector per pixel), to an approximated MB $\widetilde{B}_k$, where each partition $\widetilde{B}_k^p$ contains only one displacement vector among the different reference pictures (*i.e.*, the temporal reference picture or the inter-view reference picture). For example we have to choose one vector for the 16×16 partition, 2 for each of the 16×8 and 8×16 partitions and so on. At first, for each partition we consider a set of 6 candidates (for each reference picture), namely the average vector, the median vector and the four vectors whose norm is closest to the median one. Finally, from this set, the vector leading to the smallest RD cost is selected for each partition. Note that different reference pictures can be chosen for different partitions in the same MB.

Concerning the complexity of estimating simultaneously the temporal and inter-view displacements, we reduce the number of displacement vectors to be estimated and we therefore reduce the estimation complexity compared with the block-based MVC that estimate sequentially the temporal vectors, and afterwards the disparity vectors.

Rate-distortion optimization

Once the candidates are pre-selected for each partition, an RD selection is performed by minimizing a Lagrangian function cost. For the p^{th} partition B_k^p of the k^{th} macroblock B_k, the best displacement vector $\hat{\mathbf{u}}^p$ is derived by minimizing the following criterion:

$$\begin{aligned} \hat{\mathbf{u}}^p &= \arg\min_{\mathbf{u}^p \in \Omega} J_{\mathbf{u}^p}(B_k^p|Q) \qquad (6.14)\\ \text{with} \quad J_{\mathbf{u}^p}(B_k^p|Q) &= SSD(B_k^p, \mathbf{u}^p|Q) + \lambda \cdot R(B_k^p, \mathbf{u}^p|Q) \end{aligned}$$

where Ω is the set of candidate vectors, Q is the quantization parameter. Here SSD represents the sum of squared differences being the distortion measure, and R is the number of bits to be transmitted for the predictive displacement vector error. λ is the Lagrangian multiplier described in Section 2.3.1.
Finally, the Lagrangian mode decision for a macroblock B_k proceeds by minimizing

$$J_{\text{MODE}}(B_k|Q) = SSD(B_k, \text{MODE}|Q) + \lambda \cdot R(B_k, \text{MODE}|Q). \qquad (6.15)$$

Note that the RD cost of a MB in a specified prediction mode is the sum of the RD cost of all the partitions as follows:

$$\begin{aligned} J_{\text{MODE}}(B_k|Q) &= \sum_p J_{\mathbf{u}^p}(B_k^p|Q)\\ \text{with} \quad SSD(B_k, \text{MODE}|Q) &= \sum_p SSD(B_k^p, \text{MODE}|Q)\\ \text{and} \quad R(B_k, \text{MODE}|Q) &= \sum_p R(B_k^p, \text{MODE}|Q) \end{aligned}$$

The main relevance of the segmentation of the dense displacement field which finally ends in a block-based representation is to make good use of the smoothness of the dense displacement field. Furthermore, unlike H.264/MPEG-4 AVC in which the motion/disparity estimation is causal and local, our proposed estimation has a global approach which favors the regularization of the displacement field. As a consequence, more MBs will be coded in the SKIP mode, which is particularly efficient when the vector field is regular, since it consists in sending neither side information nor residual: the vector is computed as the median of neighbors, and the block is copied from the compensated position of the original picture. The proposed method takes advantage from the augmented effectiveness of the SKIP mode which will be selected more often, resulting in a remarkable rate reduction (see next section).

An alternative to the segmentation process for encoding the dense displacement fields, would be to consider these dense vector fields as component pictures, and thus, utilizing some near-lossless image encoders like H.264/MPEG-4 AVC in *intra* mode, lossless algorithms like JPEG 2000, or other wavelet-based approaches designed to use the similarities between the color video and its associated depth data such as those investigated in Chapter 4.

For the moment, these approaches lead to a much higher amount of bits for encoding even the smooth fields, and therefore, in coding multiview video sequences we prefer the current approach, based on quantizing and segmenting the dense fields.

6.3 Disparity prediction on temporal key pictures

Merkle *et al.* have shown in [80] that most of the additional coding gain of MVC compared to the simulcast solution, comes from the inter-view prediction of the temporal *intra* picture, while for the *inter* pictures the temporal prediction is the most efficient prediction mode. As a consequence, limiting the inter-view prediction to the *intra* pictures is commonly reputed as a reasonable complexity/efficiency trade-off.

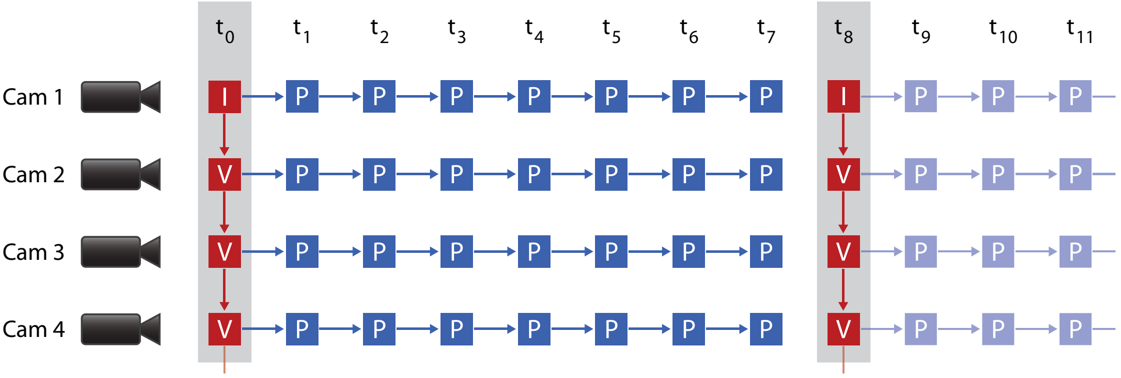


Figure 6.3: P-picture prediction structure for MVC using inter-view prediction on key pictures.

Hence, in this section we will study the case of encoding in the non-base views temporal key pictures, also known as V-pictures. We recall that V-pictures may only be inter-view predicted (see Section 2.4.3). Fig. 6.3 shows the prediction structure employed in this section.

Two main approaches, block-based and dense, have been used to estimate disparity vector fields. A survey of the different techniques proposed in the literature can be found in [101]. The MVC extension employs a variable block-based disparity estimation, assuming that within each partition of the current MB the disparity vector is constant. However, this assumption does not always hold, especially around depth discontinuities. Dense pixel-based approaches attempt to overcome this drawback by assigning one disparity vector to each pixel. Of course this means that the disparity field would require a very high bitrate to be encoded. Thus, we operate on it the RD-driven segmentation described in Section 6.2.2.

In particular, we propose improving the disparity prediction unit in the MVC extension by using the dense estimation approach described in 6.1 followed by the segmentation step.

In the followings we simplify the minimization problem presented in 6.1 to the case of a disparity estimation between two rectified video sequences. Finally, we give experimental results confirming the effectiveness of the proposed method.

6.3.1 Convex optimization framework

Let in the convex optimization framework $I_{\text{curr}} = I_t^r$ and $I_{\text{ref}} = I_t^l$ be a pair of stereo pictures taken respectively by the right and left cameras separated by a fixed baseline at time t. We assume that cameras are rectified, such that the disparity vector $\mathbf{d} = (d_x, 0)$ has only one component. The cost functional in Eq. (6.2) becomes:

$$J(d_x) = \sum_{(x,y)\in\mathcal{D}} \left[I_t^r(x,y) - I_t^l(x + d_x, y)\right]^2 \tag{6.16}$$

where $\mathcal{D} \subset \mathbb{N}^2$ is the picture support and Ω is the range of candidate disparity values. The left view being the base view, and the right view the non-base view being inter-view predicted.

To circumvent the non-convexity of the criterion in Eq. (6.16), the compensated reference picture is approximated around $\overline{d}_x$ by a Taylor expansion:

$$I_t^l(x + d_x, y) \simeq I_t^l(x + \overline{d}_x, y) + (d_x - \overline{d}_x) \cdot \nabla_x I_t^l(x + \overline{d}_x, y) \tag{6.17}$$

where ∇_x is the horizontal gradient operator.

The quadratic convex functional $\widetilde{J}$ in d_x becomes:

$$\widetilde{J}(d_x) = \sum_{(x,y)\in\mathcal{D}} \left[L(x,y) \cdot d_x - r(x,y)\right]^2 \tag{6.18}$$

where

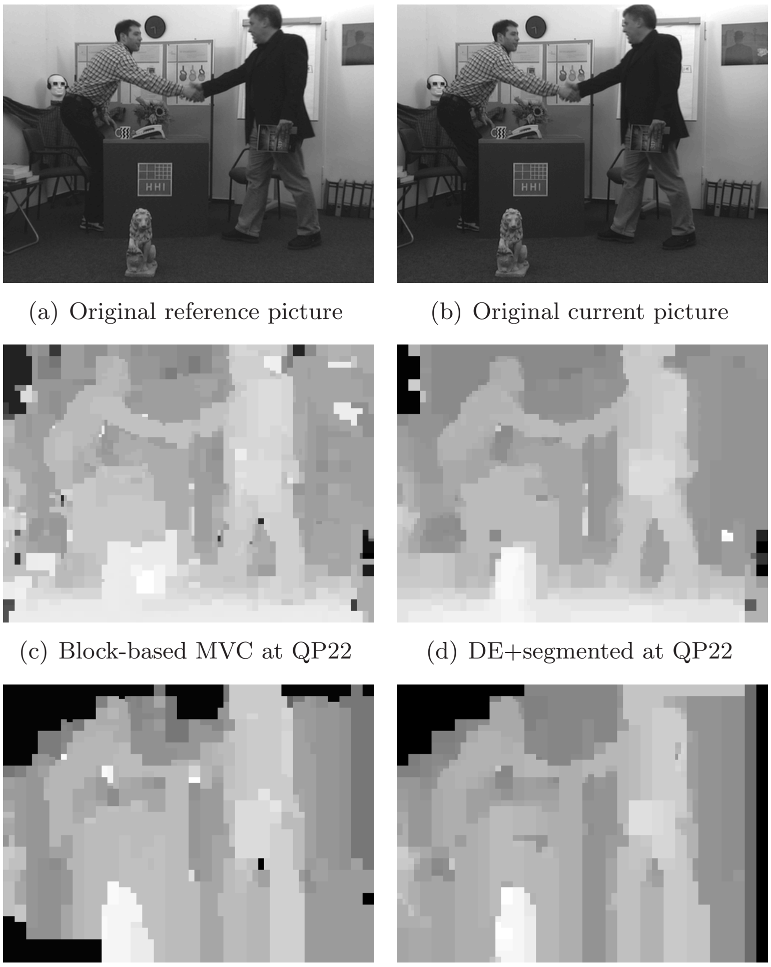
$$\begin{aligned} L(x,y) &= \nabla_x I_t^l(x + \overline{d}_x, y) \\ r(x,y) &= I_t^r(x,y) - I_t^l(x + \overline{d}_x, y) + L(x,y) \cdot \overline{d}_x \end{aligned}$$

Finally, the quadratic convex functional $\widetilde{J}$ is minimized using the algorithm described in Section 6.1.2.

6.3.2 Experimental results

In this section, we provide some simulation results to evaluate the RD performance of the proposed structure. The experiments were run on three multiview video sequences provided by Fraunhofer HHI [39]: "Book arrival", "Door flowers" and "Outdoor". For all the video sequences, we use four views with a spatial resolution reduced to 512×384. The multiview video sequences are all rectified, which implies in the following a null vertical component of the disparity vector field. Considering a GOP length of 12, experimental results are reported on V-pictures that do use disparity prediction only. We use the JMVM 8.0 software [129].

Within the H.264/MPEG-4 AVC framework, the RD estimation of the disparity vector generates different disparity fields at different QP values (Fig. 6.4). Disparity fields are usually smoother at low bitrate (high QP value) which favors the selection of the SKIP mode. At high bitrate (low QP value), the distortion is privileged against the cost of the predictive disparity error which reduces the number of SKIP MBs. We present a comparison in Fig. 6.5 at two QP points: 22 and 42. We can see in black the SKIP MBs. Our method has the benefit to generate a smooth block-based representation of the disparity vectors field at high bitrate, which reduces the predictive disparity error, and subsequently uses more SKIP MBs. Especially at high bitrate, when our method is used, the number of SKIP MBs increases, with a beneficial effect on the required coding rate.


(a) Original reference picture (b) Original current picture

(c) Block-based MVC at QP22 (d) DE+segmented at QP22

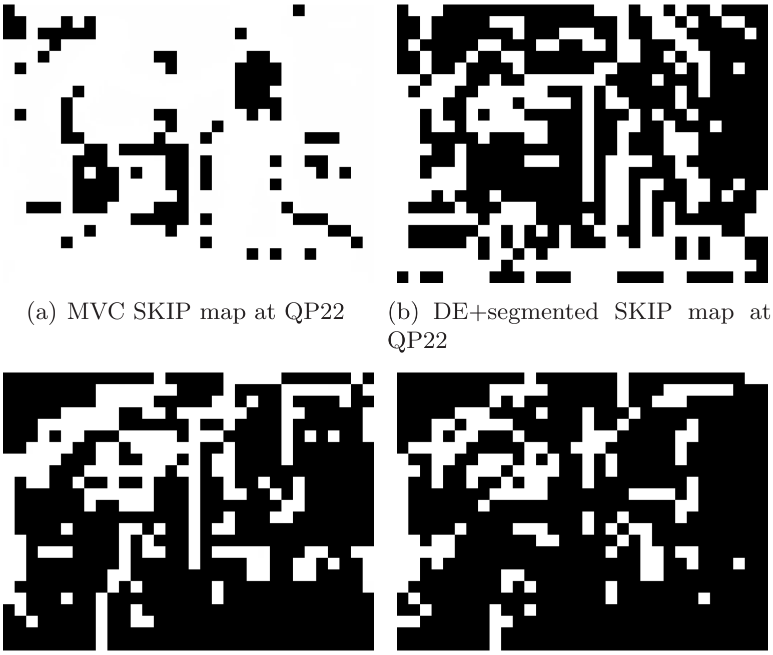
(e) Block-based MVC at QP42 (f) DE+segmented at QP42

Figure 6.4: Example of block-based fields issued from the classical H.264-based estimation and the proposed dense estimation (DE) (from “Book arrival” sequence).

For example at QP=22 on the multiview video sequence “Book arrival” (Fig. 6.5), with the proposed method 58% of MB are coded in the SKIP mode, with respect to a mere 16% for the original encoder. At QP=42, we obtained a percentage of 81% against 72% as shown in Fig. 6.11.

Fig. 6.10 shows the results in terms of RD performance. Comparing the dense disparity estimation to the block-based reference H.264/MPEG-4 AVC estimation clearly indicates the benefits of a dense estimation followed by a segmentation optimized for RD efficiency, especially for the “Outdoor” sequence, where a coding gain of 1.5 dB is achieved. The curve consists of 5 QP points which are 22, 27, 32, 37, 42.

In addition, to measure the relative gain we used the Bjontegaard metric [13]. The results are shown in Table 6.4 for low bitrate and high bitrate corresponding respectively to the four QP points 27, 32, 37, 42 and 22, 27, 32, 37. We can see that our method works especially well on the sequence “Outdoor” (in which the disparity range is small, $[d_{\min}, d_{\max}] = [0, 8]$).



(a) MVC SKIP map at QP22

(b) DE+segmented SKIP map at QP22

(c) MVC SKIP map at QP42

(d) DE+segmented SKIP map at QP42

Figure 6.5: Example of SKIP map corresponding to the disparity vector fields in Fig. 6.4. In black there are the SKIP MBs and in white the inter-MBs.

Table 6.4: Calculation of average PSNR differences and the bitrate saving. “low” represents the low bitrate range (QP points 27,...,42) and “high” the high bitrate range (QP points 22,...,37).

	bitrate saving		**PSNR gain**	
	low	**high**	**low**	**high**
Book arrival	-1.49 %	-2.86 %	0.04 dB	0.10 dB
Door flowers	-12.83 %	-10.86 %	0.58 dB	0.52 dB
Outdoor	-60.03 %	-45.58 %	1.93 dB	1.59 dB

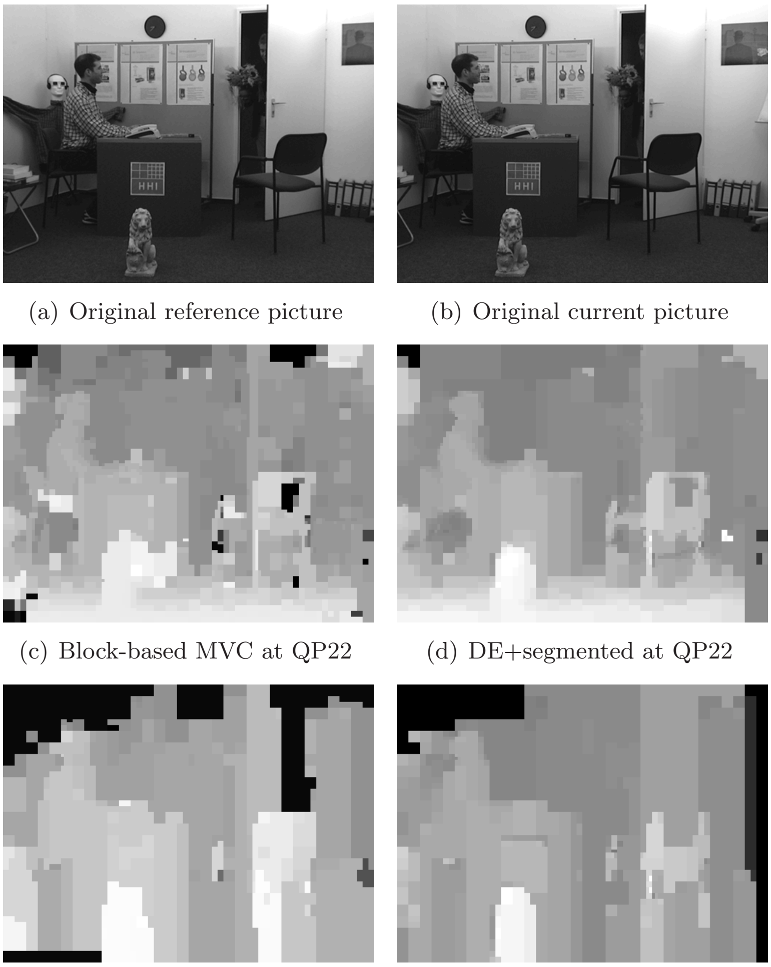
6.3.3 Conclusion

In this section, we have presented the benefits of using a dense disparity estimation followed by a block-based segmentation and coding of the disparity field in MVC. As expected, a dense disparity estimation produces a smooth disparity field with an ideally infinite precision. This field is then presented with a quarter pixel precision and segmented based on an RD-optimized fashion. The smooth property of the estimated disparity vector fields increase the SKIP prediction, and therefore, the algorithm can achieve a better coding gain.

6.4 Joint motion/disparity prediction on non-key pictures

In this section, we extend the primary work of the previous section on dense estimation of *intra* pictures to temporal *inter* pictures.

In the case of temporal *inter* pictures in MVC, the temporal plus the inter-view corre-


(a) Original reference picture (b) Original current picture

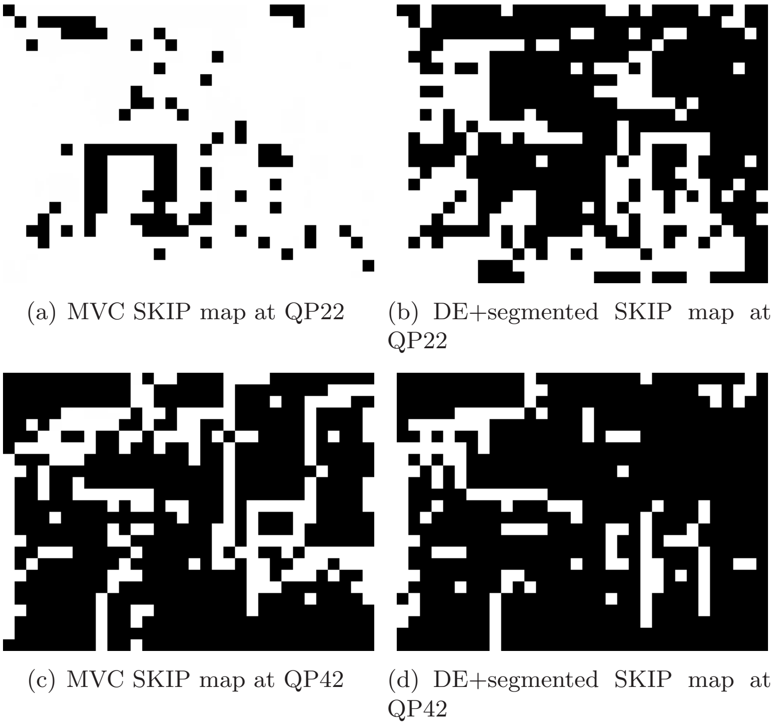
(c) Block-based MVC at QP22 (d) DE+segmented at QP22

(e) Block-based MVC at QP42 (f) DE+segmented at QP42

Figure 6.6: Example of block-based fields issued from the classical H.264-based estimation and the proposed dense estimation (DE) (from "Door flowers" sequence).

lations are exploited by combining motion/disparity compensated prediction, where temporal *inter* pictures are not only predicted from temporally neighboring pictures but also from corresponding pictures in the adjacent views as illustrated in Fig. 6.12. This involves the estimation of motion and disparity fields and may lead to a high computational cost.

One way to overcome these problems is to jointly estimate disparity and motion fields, by fully exploiting the relations between both vector fields, while incorporating in the estimation additional constraints on the smoothness and bounds for motion and disparity vectors. Using the epipolar constraint, an RD optimized framework has been proposed in [138], in which two prediction techniques based on DIBR and disparity compensation are jointly optimized. In the case of no *a priori* knowledge about the cameras several approaches have been proposed for combining inter-view and motion analysis within stereoscopic video sequences [137, 68, 116]. In [131], the joint estimation was performed on a multi-resolution pyramid of images using an anisotropic diffusion regularization to preserve image boundaries. In [95], the authors proposed a multi-scale iterative relaxation algorithm to first calculate the disparity field of the first stereoscopic pair. The left and right motion fields, involved in the two consecutive stereo pictures, are then simultaneously



(a) MVC SKIP map at QP22

(b) DE+segmented SKIP map at QP22

(c) MVC SKIP map at QP42

(d) DE+segmented SKIP map at QP42

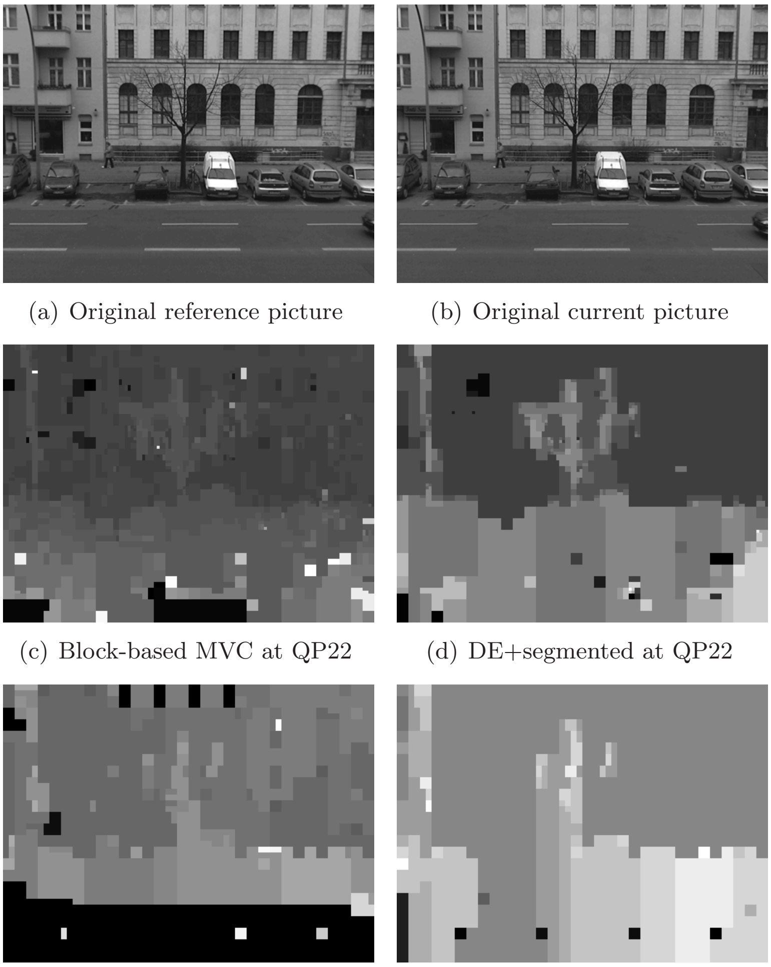
Figure 6.7: Example of SKIP map corresponding to the disparity vector fields in Fig. 6.6. In black the SKIP MBs and in white the inter-MBs.

estimated. Using the computed vector fields and the *stereo-motion consistency* constraint, the current disparity field is implicitly constructed, and refined later using the same multiscale relaxation algorithm. In [84], an edge-preserving regularization algorithm that simultaneously calculates dense disparity and motion fields is proposed. The authors use the Euler-Lagrange equations within a variational framework to minimize a global edge-preserving energy function. Although interesting results were reported, the discretization of the partial differential equation, using a finite difference method, is a difficult and numerically unstable task. Furthermore, all previously mentioned approaches were proposed in the case of stereo video sequences and no extension to the case of multiview video or adaptation to MVC exists.

In the following, we first describe the relationship between disparity and motion vectors and then describe the convex variational algorithm, we proposed simultaneously to estimate motion and disparity vectors.

6.4.1 Joint estimation model

We take the case of a stereoscopic video sequence as an example to describe the proposed joint estimation model. With reference to Fig. 6.13, let us consider two consecutive picture pairs, denoted by I^l_{t-1}, I^r_{t-1}, I^l_t and I^r_t, which are, respectively, the left and right views of the previous and current pictures of a stereo video sequence. Let $\mathbf{v}^l = (v^l_x, v^l_y)$ and $\mathbf{v}^r = (v^r_x, v^r_y)$ be the left and the right motion fields, and let $\mathbf{d}_{t-1} = (d_{t-1_x}, d_{t-1_y})$ and $\mathbf{d}_t = (d_{t_x}, d_{t_y})$ designate the disparity vector fields of the stereo picture pair at time $t-1$ and t. Note that there is no hypothesis of camera alignment, the disparity fields involving bi-dimensional vectors. We assume that the temporal motion for the left view $\mathbf{v}^l$, and the



(a) Original reference picture (b) Original current picture

(c) Block-based MVC at QP22 (d) DE+segmented at QP22

(e) Block-based MVC at QP42 (f) DE+segmented at QP42

Figure 6.8: Example of block-based fields issued from the classical H.264-based estimation and the proposed dense estimation (DE) (from “Outdoor” sequence).

disparity at time $t-1$, $\mathbf{d}_{t-1}$, are already known and we aim at estimating the temporal motion for the right view $\mathbf{v}^r$ along with the disparity at time t, $\mathbf{d}_t$. If the four vector fields relate to the projections of the same physical point in the scene, the following constraint, illustrated in Fig. 6.13, must hold:

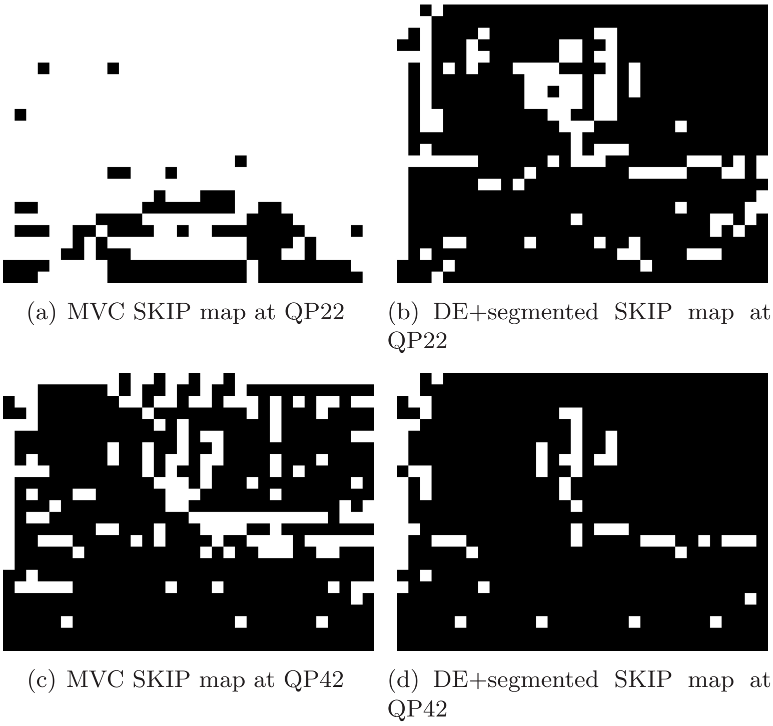
$$\mathbf{d}_{t-1} + \mathbf{v}^r - \mathbf{d}_t - \mathbf{v}^l = \mathbf{0}_2 \tag{6.19}$$

where $\mathbf{0}_2 = (0, 0)$ is the bi-dimensional null vector.

Assuming that the spatial point is projected to the pixel $s = (x, y)$ on frame I_t^r, the previous consistency constraint can be expressed as follows:

$$\mathbf{d}_{t-1}\big(s + \mathbf{v}^r(s)\big) + \mathbf{v}^r\big(s\big) - \mathbf{d}_t\big(s\big) - \mathbf{v}^l\big(s + \mathbf{d}_t(s)\big) = \mathbf{0}_2 \tag{6.20}$$

In what follows, we use this constraint, which establishes the relationship between motion vectors and disparity vectors, to calculate two displacement fields from the two other fields within a joint optimization framework.



(a) MVC SKIP map at QP22

(b) DE+segmented SKIP map at QP22

(c) MVC SKIP map at QP42

(d) DE+segmented SKIP map at QP42

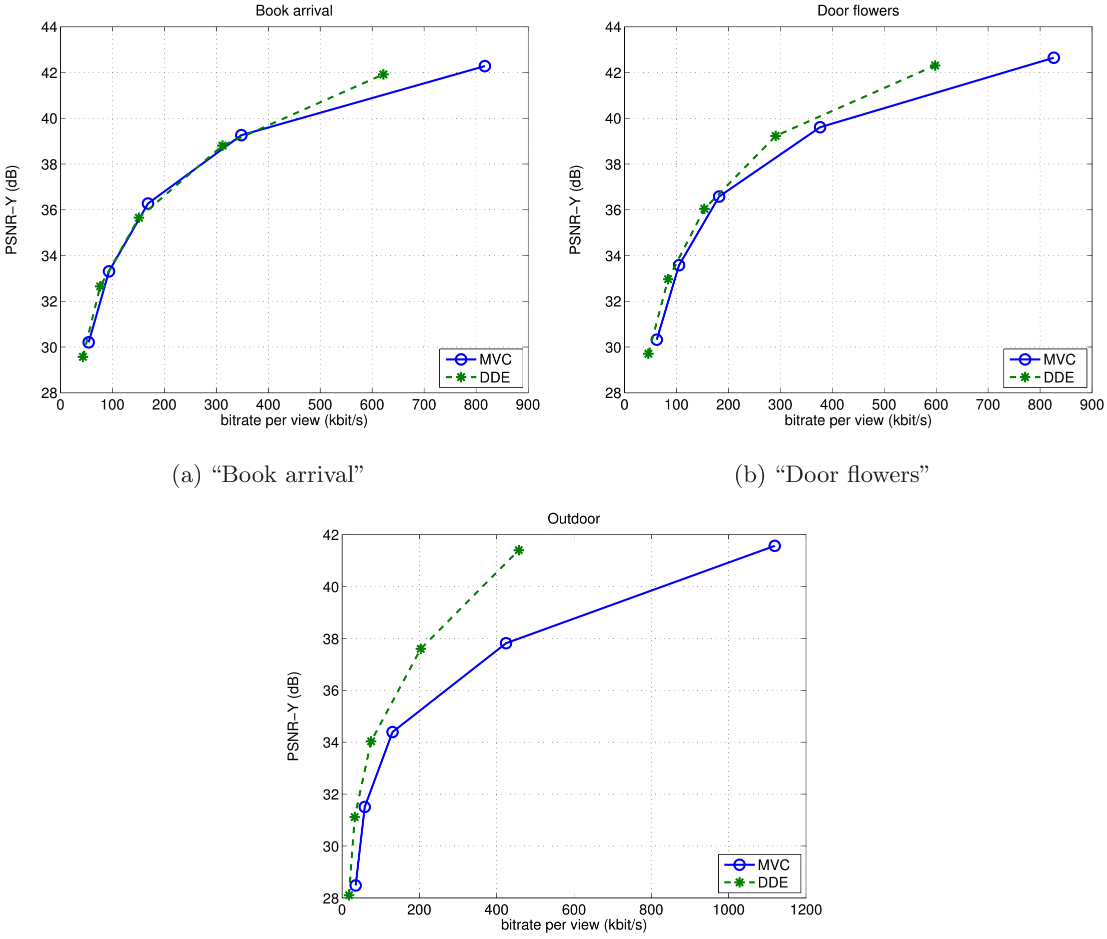
Figure 6.9: Example of SKIP map corresponding to the disparity vector fields in the Fig. 6.8. In black the SKIP MBs and in white the inter-MBs.

6.4.2 Convex optimization framework

According to the constraint in Eq. (6.20), the disparity field $\mathbf{d}_{t-1}$ obtained at time $t-1$ and the left motion field $\mathbf{v}^l$ can be used to simultaneously estimate the disparity field $\mathbf{d}_t$ obtained at time t and the right motion field $\mathbf{v}^r$. This constraint, that forms a loop from the four vectors, may be exploited to estimate the dense vector fields (one motion or disparity vector per pixel). Looking for corresponding pixels in the two pairs of images, the estimation of the dense fields can be performed by minimizing the following objective function:

$$\begin{aligned} J(\mathbf{v}^r, \mathbf{d}_t) = & \sum_{s \in \mathcal{D}} \left[I_t^r(s) - I_{t-1}^r(s + \mathbf{v}^r) \right]^2 \\ & + \sum_{s \in \mathcal{D}} [I_t^r(s) - I_t^l(s + \mathbf{d}_t)]^2 \\ & + \sum_{s \in \mathcal{D}} \left[I_{t-1}^l(s + \mathbf{v}^r + \mathbf{d}_{t-1}(s + \mathbf{v}^r)) - I_{t-1}^l(s + \mathbf{d}_t + \mathbf{v}^l(s + \mathbf{d}_t)) \right]^2 \end{aligned} \qquad (6.21)$$

where $\mathcal{D} \subset \mathbb{N}^2$ is the picture support. Note that, for notation concision, we have not made explicit in the equation above that $\mathbf{v}^r$ and $\mathbf{d}_t$ are also functions of s. The cost function in Eq. (6.21) consists of three error terms: the first two for right motion and current disparity fields and the latter one implicitly expresses the loop constraint by minimizing the distance between the two projections of s (following the two paths of the loop in Fig. 6.13) onto the reference picture I_{t-1}^l. The search for the vector combination $(\mathbf{v}^r, \mathbf{d}_t)$ that minimizes the objective function J is formulated, similarly to [83], as a convex optimization problem. This requires the approximation of the cost function in Eq. (6.21) by a quadratic convex



(a) “Book arrival”

(b) “Door flowers”

(c) “Outdoor”

Figure 6.10: Rate-distortion coding results.

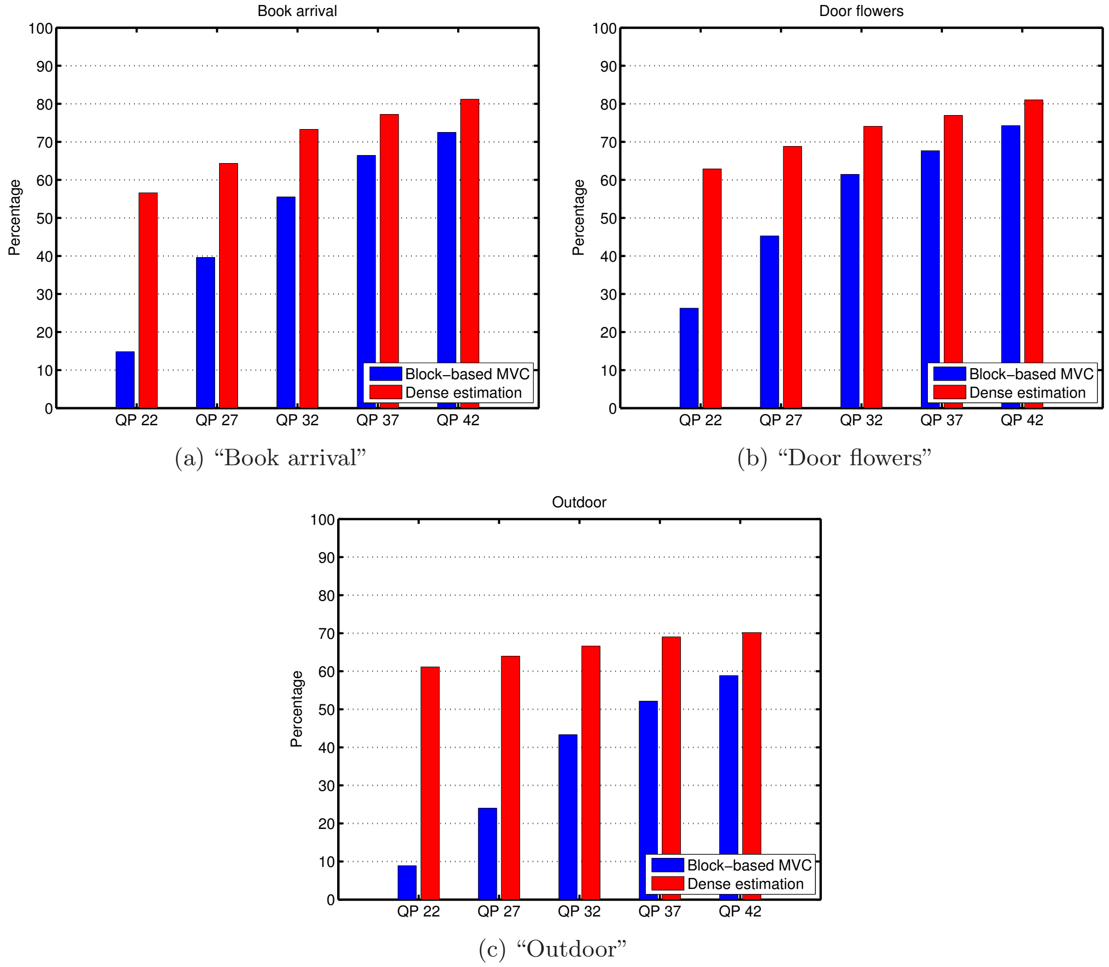
objective function, by considering a Taylor expansion of the non-linear terms around the initial estimates $\bar{\mathbf{v}}^r$ and $\bar{\mathbf{d}}_t$, as follows:

$$\begin{aligned} I^r_{t-1}(s+\mathbf{v}^r) &\simeq I^r_{t-1}(s+\bar{\mathbf{v}}^r) + (\mathbf{v}^r - \bar{\mathbf{v}}^r)\cdot \nabla I^r_{t-1}(s+\bar{\mathbf{v}}^r) \\ I^l_t(s+\mathbf{d}_t) &\simeq I^l_t(s+\bar{\mathbf{d}}_t) + (\mathbf{d}_t - \bar{\mathbf{d}}_t)\cdot \nabla I^l_t(s+\bar{\mathbf{d}}_t) \\ I^l_{t-1}(s+\mathbf{v}^r+\mathbf{d}_{t-1}) &\simeq I^l_{t-1}(s+\bar{\mathbf{v}}^r+\mathbf{d}_{t-1}) + (\mathbf{v}^r - \bar{\mathbf{v}}^r)\cdot \nabla I^l_{t-1}(s+\bar{\mathbf{v}}^r+\mathbf{d}_{t-1}) \\ I^l_{t-1}(s+\mathbf{d}_t+\mathbf{v}^l) &\simeq I^l_{t-1}(s+\bar{\mathbf{d}}_t+\mathbf{v}^l) + (\mathbf{d}_t - \bar{\mathbf{d}}_t)\cdot \nabla I^l_{t-1}(s+\bar{\mathbf{d}}_t+\mathbf{v}^l) \end{aligned}$$

The initial estimates $\bar{\mathbf{v}}^r$ and $\bar{\mathbf{d}}_t$ may be obtained from a correlation based method or from previous iterations within an iterative process.

Now, consider the vector parameters $\mathbf{w} = [\mathbf{v}^r, \mathbf{d}_t]^\top$. The joint estimation problem may be rewritten using the previous approximations by minimizing the following objective function with respect to $\mathbf{w}$:

$$J(\mathbf{w}) = \sum_{i=1}^{3}\sum_{s\in\mathcal{D}} [\mathbf{L}_i(s)\mathbf{w}(s) - \mathbf{r}_i(s)]^2 \tag{6.22}$$



(a) "Book arrival"

(b) "Door flowers"

(c) "Outdoor"

Figure 6.11: Percentage of disparity MBs coded in SKIP mode.

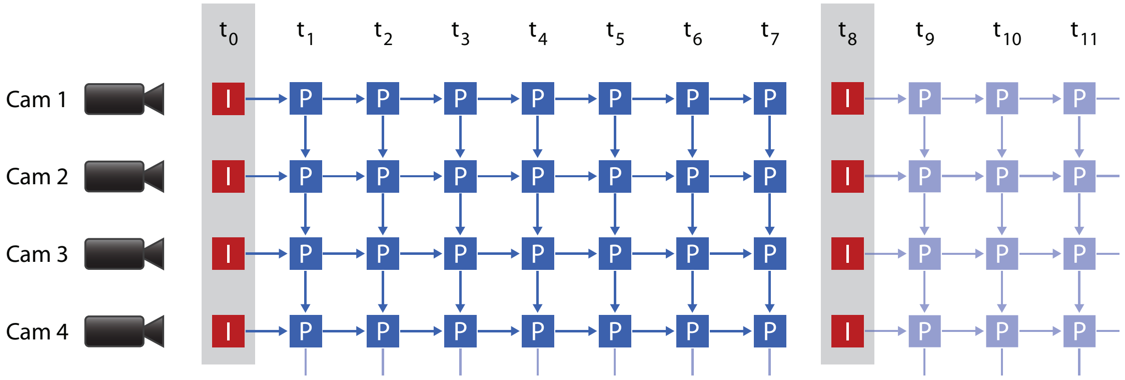


Figure 6.12: P-picture prediction structure for MVC using inter-view prediction on non-key pictures.

where

$$\begin{cases} \mathbf{L}_1(s) = \left[\nabla I^r_{t-1}(s+\bar{\mathbf{v}}^r), \mathbf{0}_2\right] \\ \mathbf{L}_2(s) = \left[\mathbf{0}_2, \nabla I^l_t(s+\bar{\mathbf{d}}_t)\right] \\ \mathbf{L}_3(s) = \left[\nabla I^l_{t-1}(s+\bar{\mathbf{v}}^r+\mathbf{d}_{t-1}), -\nabla I^l_{t-1}(s+\bar{\mathbf{d}}_t+\mathbf{v}^l)\right] \end{cases}$$

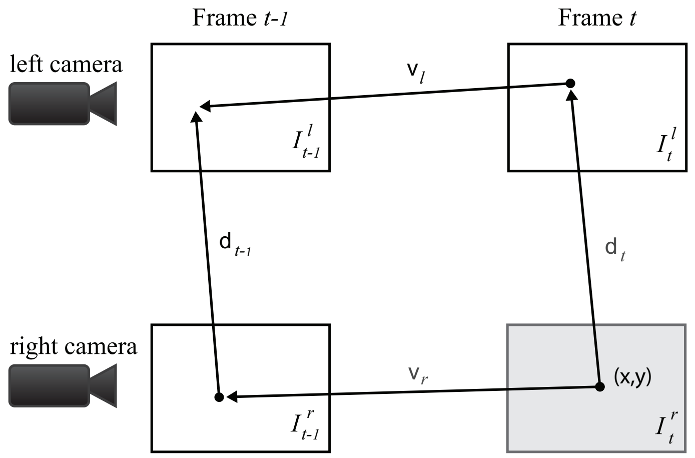


Figure 6.13: The geometric consistency constraint.

and

$$\begin{cases} \mathbf{r}_1(s) = I^r_t - I^r_{t-1}(s + \bar{\mathbf{v}}^r) + \mathbf{L}_1(s)\bar{\mathbf{w}}(s) \\ \mathbf{r}_2(s) = I^r_t - I^l_t(s + \bar{\mathbf{d}}_t) + \mathbf{L}_2(s)\bar{\mathbf{w}}(s) \\ \mathbf{r}_3(s) = I^l_{t-1}(s + \bar{\mathbf{v}}^r + \mathbf{d}_{t-1}) - I^l_{t-1}(s + \bar{\mathbf{d}}_t + \mathbf{v}^l) + \mathbf{L}_3(s)\bar{\mathbf{w}}(s) \end{cases}$$

Finally the quadratic convex functional $\widetilde{J}$ is minimized using the algorithm described in Section 6.1.2.

6.4.3 Experimental results

In this section, we provide some simulation results to evaluate the RD performance of the proposed structure. The experiments were run on three multiview video sequences : "Book arrival", "Door flowers" and "Outdoor" [39]. For all these video sequences, we use a spatial resolution reduced to 512×384. The multiview video sequences are all rectified, which implies in the following a null vertical component of the disparity vector field. Considering a GOP length of 12, experimental results are reported in the non-base views on the temporal *inter* pictures that do use motion/disparity prediction (since for *intra* pictures all the compared structures perform identically). We use the JMVM 8.0 software [129].

The derivation of the joint estimation of the motion/disparity displacement field at a time t requires the knowledge of the motion vector field in the reference view and the disparity vector field at the previous picture taken at time $t-1$. In the case of coding the picture next to the key picture (in temporal term), the previous disparity vector field (*i.e.* the disparity vector field of the key-picture) is obtained with the method described in [31]. Otherwise, due to the coding scheme order, both the motion vector in the reference view and the disparity vector fields are known. The two remaining displacement vector fields are thus jointly estimated as presented in Section 6.4.2. Our method has the benefit of deriving dense smooth fields while preserving the discontinuities around the objects with an ideally infinite precision as illustrated in Fig. 6.14.

Once the dense displacement vector field is obtained with the algorithm described in Section 6.4.2, the RD segmentation is performed in order to obtain the block-based vector field, which is compatible with the block-based representation of motion vectors in

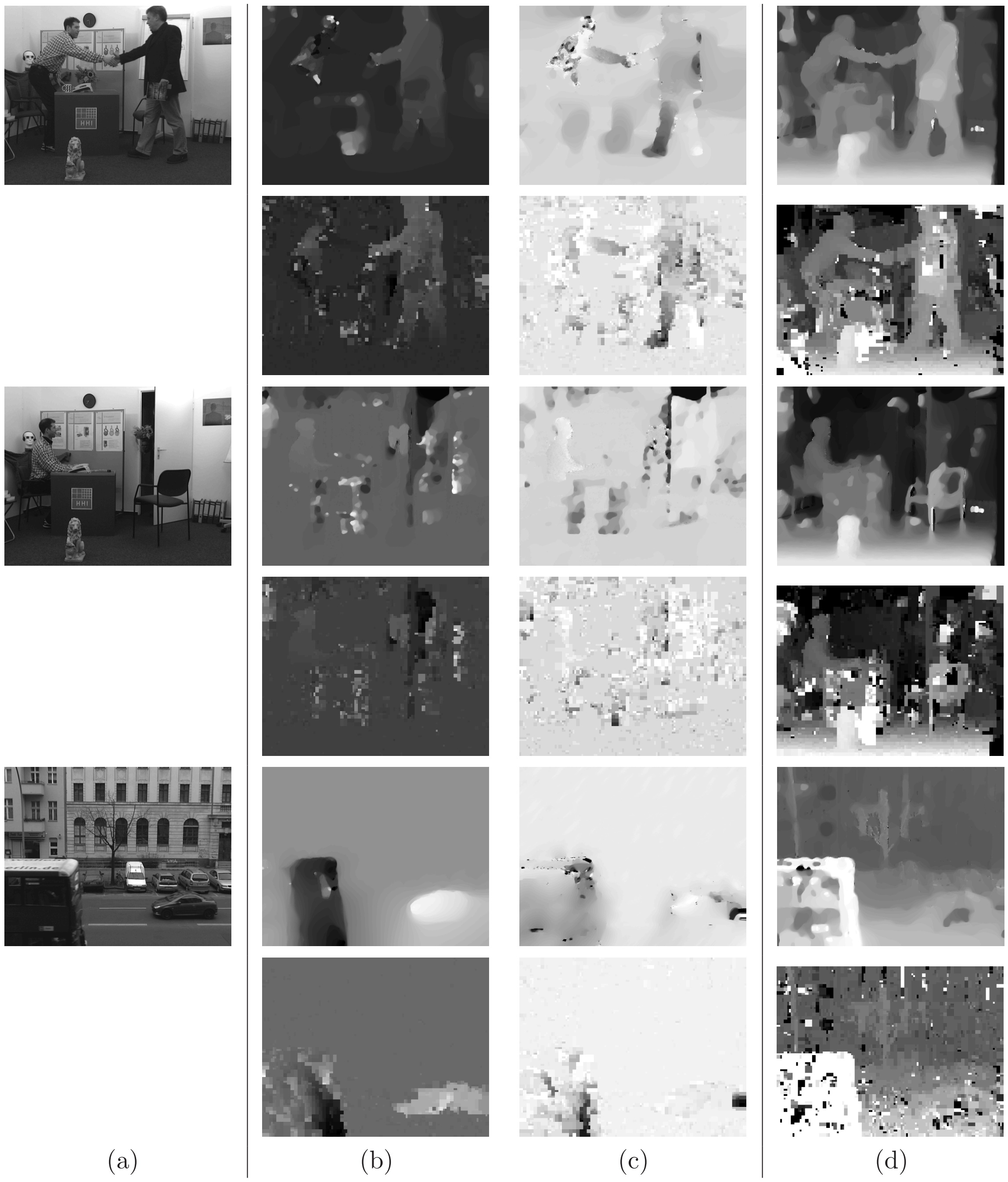


Figure 6.14: Example of jointly estimated (first row of each sequence) and block-based estimated (second row for each sequence) displacement fields (from "Book arrival" picture 35 (up), "Door flower" picture 35 (middle) and "Outdoor" picture 41 (bottom)) : (a) current picture, (b) horizontal motion component, (c) vertical motion component, (d) horizontal disparity component.

H.264/MPEG-4 AVC. Furthermore, the dense motion vector field and the dense disparity vector field are jointly segmented and combined, as described in Section 6.2.2, in order to provide a bitstream compliant with MVC. As result, after the RD segmentation part, the obtained segmented vector field is a combination of the previous two dense displacement vector fields where for each partition, the displacement vector with the smallest RD cost has been chosen. In practice, motion vectors are chosen more often than disparity vectors as shown in Table 6.5. This can be explained by the small value of the motion vectors compared to the disparity vectors value in most multiview video sequences.

Table 6.5: Average percentage representing the distribution of MBs using motion prediction or disparity prediction in the proposed framework using a dense estimation (DE) and in MVC for the three test sequences.

Sequence	**motion prediction**		**disparity prediction**	
	DE	MVC	DE	MVC
Book arrival	76 %	87 %	24 %	13 %
Door flowers	82 %	93 %	18 %	7 %
Outdoor	80 %	79 %	20 %	21 %

For illustration, we present in Fig. 6.15 examples of reference maps, in which for each MBs is represented the reference picture type (*i.e.* temporal or inter-view) which is used. We can see a repartition of the reference picture type between the background with small motion, and the objects or characters in motion. The MBs belonging to the background use for most of them the temporal prediction, whereas, the MB with high motion use more often the disparity prediction.

As seen in the Section 6.3, due to the RD design of the segmentation process, the segmented vector field is different at different QP values. A high QP value favors a smooth and regular vector field, that privileges a low rate of the displacement vector field. As a consequence, more MBs are coded in the SKIP mode. At high bitrate, the Lagrangian parameters λ (for motion and disparity) decrease and the distortion term gets more importance in the computation of the RD cost criterion J, which reduces the number of SKIP MBs. As illustrated in Fig. 6.14, our method has the benefit of generating a smooth block-based representation of the displacement vector field. As a result, the total number of combined motion/disparity MB coded in SKIP mode increases with our method as illustrated in Fig. 6.16. In addition, we propose as well a comparison with a separated dense estimation where each displacement vector field, motion and disparity, are estimated independently using the method described in Section 6.1.

In our experiments, the benefit in terms of computational load of using the joint estimation model is the reduction of the execution time by about 30 to 40 percent compared to a separate estimation.

Fig. 6.17 and Fig. 6.18 show the results in terms of RD performance respectively for high and low bitrate. Comparing the dense estimation to the block-based reference H.264/MPEG-4 AVC estimation clearly indicates the benefits of a dense estimation followed by a segmentation optimized for RD coding efficiency. In addition we propose comparing our proposed framework with a separate dense estimation based on a local matching cost computation [101] with pixel accuracy, that we refer to as separated correlation estimation. We can also observe an improvement of the joint dense estimation over the separated dense estimation. The curves consist of 4 QP points which are 22, 27, 32, 37 for high bitrate and 32, 37, 42, 47 for low bitrate. As expected, an increased selection of the SKIP mode implies a bitrate reduction with a small reduction of quality of the reconstructed picture at small QP in the test set.

In addition, to measure the relative gain we used the Bjontegaard metric [13] as recommended by VCEG. The results are shown in Table 6.6 and Table 6.7, corresponding at

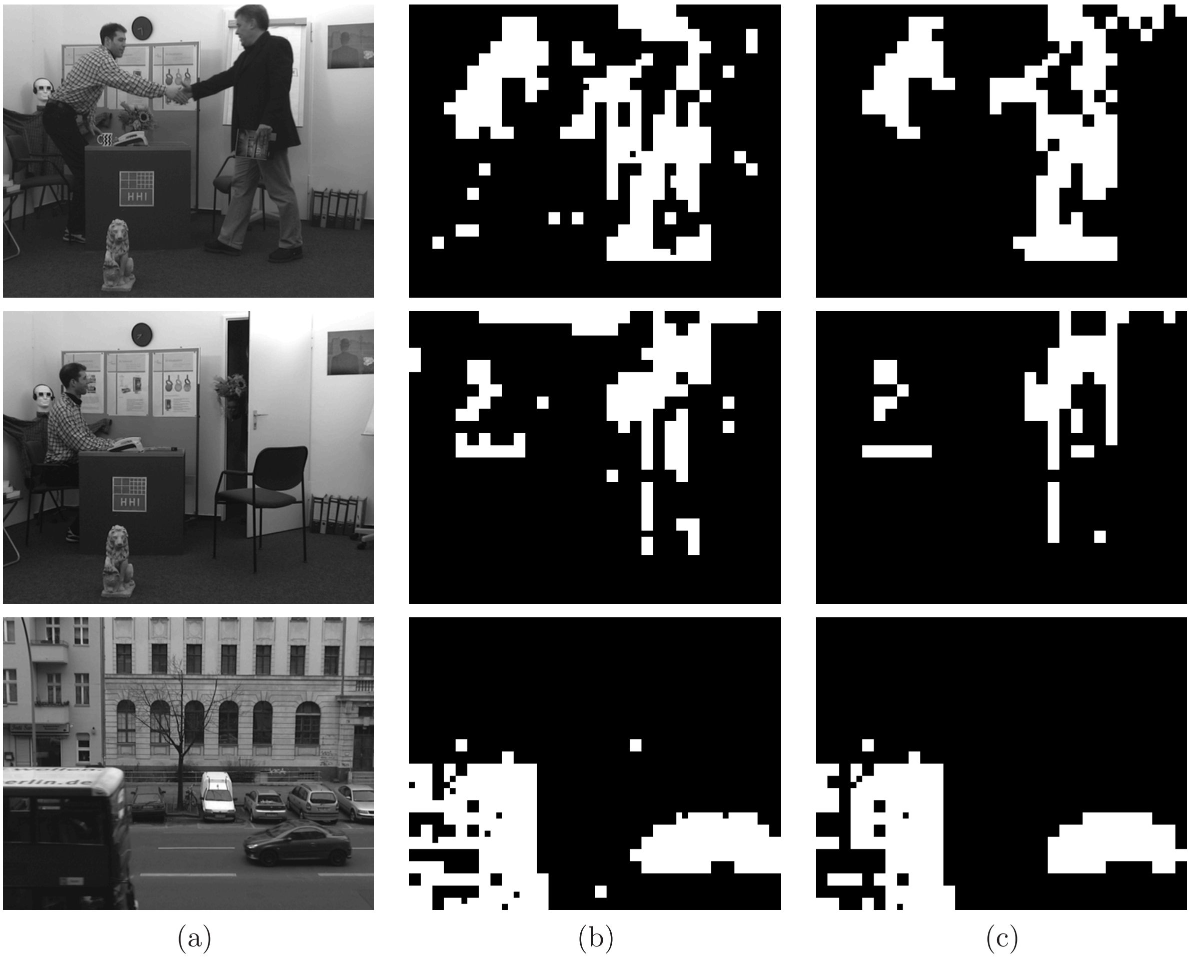
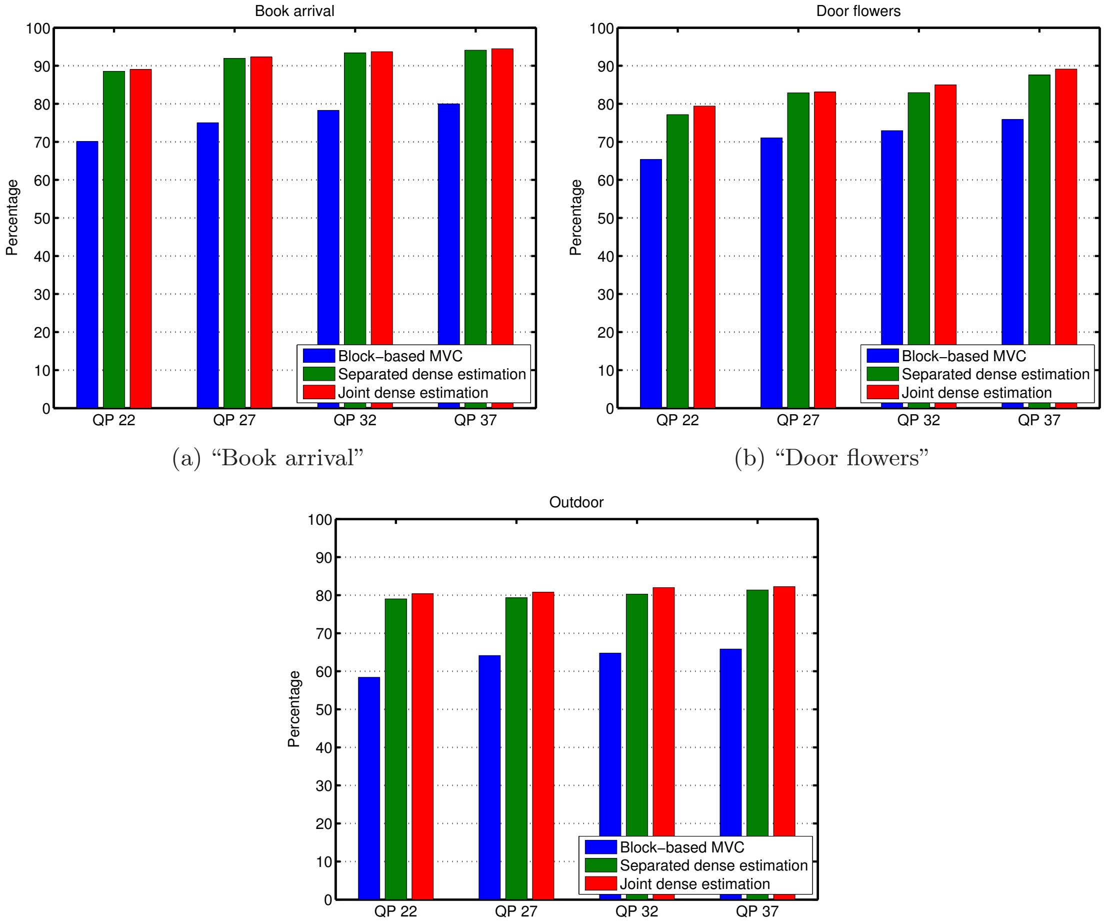


Figure 6.15: Example of reference map in which the black MBs used motion prediction, and the white MBs used disparity prediction in the proposed framework (from "Book arrival" picture 35 (up), "Door flower" picture 35 (middle) and "Outdoor" picture 41 (bottom)) : (a) current picture, (b) reference map at QP 22, (c) reference map at QP 37.

high bitrate to the four QP points 22, 27, 32, 37, and at low bitrate to the four QP points 32, 37, 42, 47. Note that a bitrate saving compared with a reference method corresponds to negative values. Table 6.6 represents the comparison between the joint dense estimation and MVC, and Table 6.7 the comparison between the joint dense estimation and the separated dense estimation. The gains are consistent and systematic, for all the tested sequences, and all the bitrates.

Table 6.6: Average PSNR gains and the corresponding bitrate savings (joint dense estimation vs MVC) for the three test sequences.

Sequence	**bitrate saving**		PSNR **gain**	
	low	**high**	**low**	**high**
Book arrival	-14.07 %	-19.52 %	0.64 dB	0.87 dB
Door flowers	-14.91 %	-10.74 %	1.21 dB	0.52 dB
Outdoor	-10.42 %	-16.34 %	1.13 dB	0.95 dB



(a) "Book arrival"

(b) "Door flowers"

(c) "Outdoor"

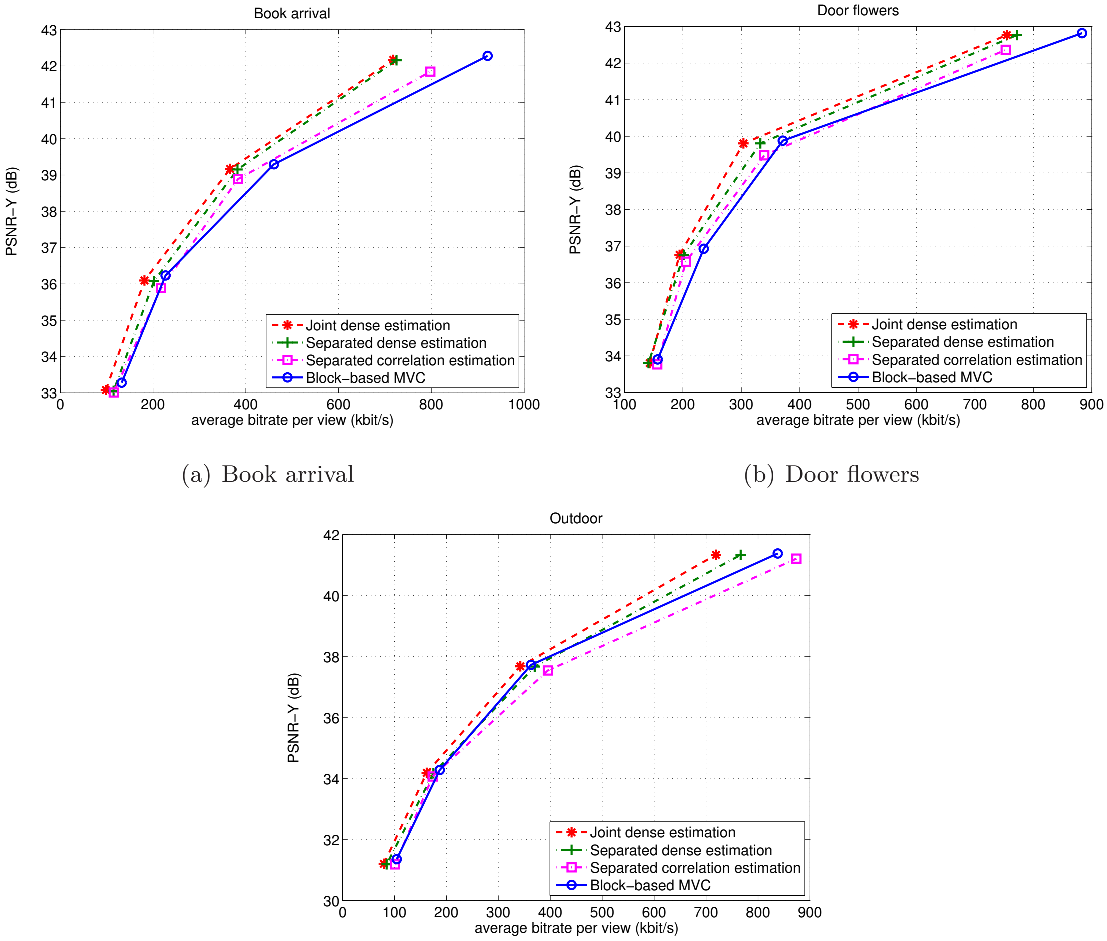
Figure 6.16: Percentage of combined motion/disparity MBs coded in SKIP mode.

Table 6.7: Average PSNR gains and the corresponding bitrate savings (joint dense estimation vs separate dense estimation) for the three test sequences.

Sequence	**bitrate saving**		PSNR **gain**	
	low	**high**	**low**	**high**
Book arrival	-5.57 %	-6.44 %	0.18 dB	0.29 dB
Door flowers	-5.07 %	-4.42 %	0.44 dB	0.23 dB
Outdoor	-3.60 %	-4.42 %	0.68 dB	0.30 dB

6.4.4 Conclusion

In this section, we introduced the consistency constraint on the displacement vectors in a stereo/multiview sequence. Based on the observed relationship, we modeled joint motion/disparity estimation, and solved the problem using convex optimization with subgradient projection. Then, each obtained dense motion/disparity field is partitioned into variable block sizes, which are supported in H.264-based codec, while minimizing RD cost. By fully exploiting the relations between both vector fields, more coherent estimations are derived allowing a reduction the bitrate cost of the motion/disparity with a better accuracy.



(a) Book arrival

(b) Door flowers

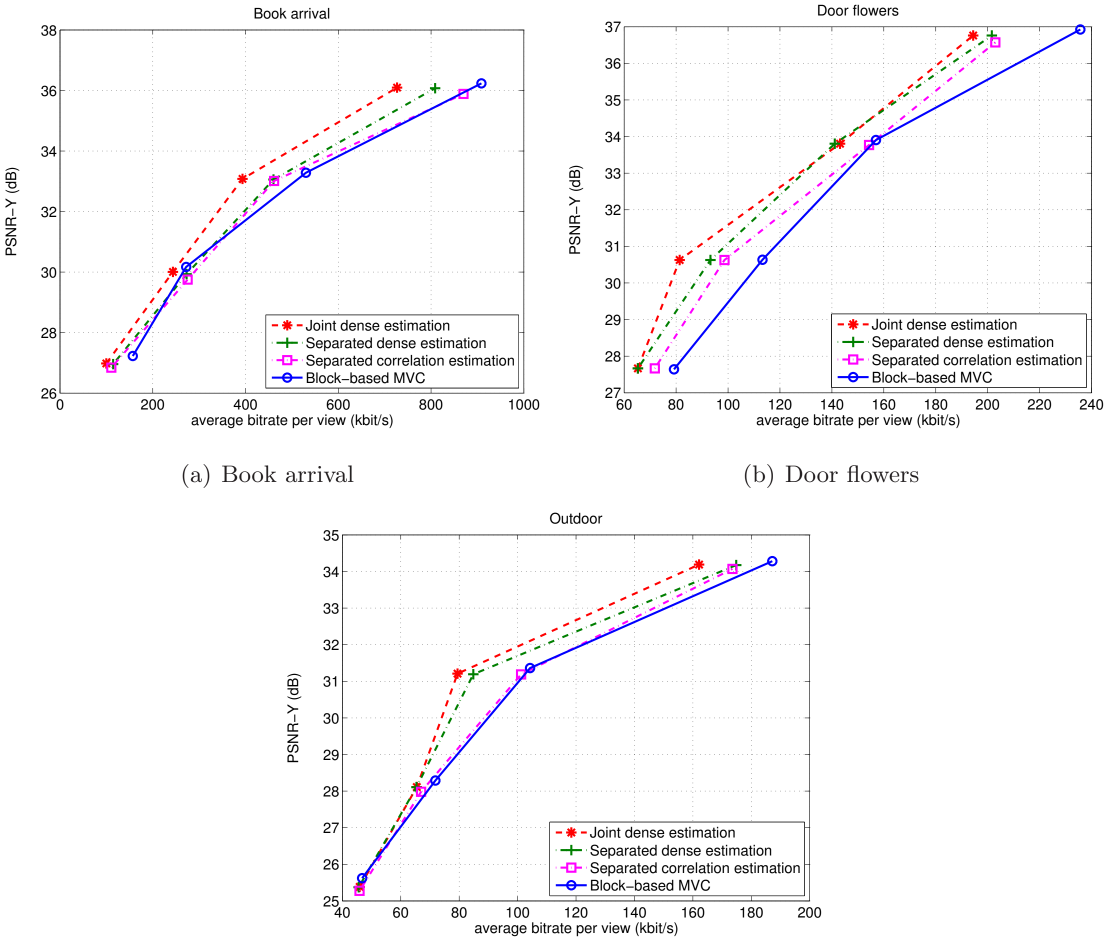
(c) Outdoor

Figure 6.17: Rate-distortion coding results for the QP set [22, 27, 32, 37].

6.5 Conclusion and future work

In this chapter, we addressed the problem of reducing temporal/inter-view redundancies of key pictures and non-key pictures with disparity estimation and joint motion/disparity estimation. We proposed a dense motion/disparity estimation framework followed by an RD-driven segmentation and coding designed to replace the block-based motion/disparity estimation stage in MVC extension, which lead to improve overall RD performances.

Considering an alternative of segmenting the dense vector fields, this requires future work on the improvement of the coding efficiency in terms of bitrate and the quality of the reconstructed picture, by using for example an appropriate quantification method. We intend to pursue the work initiated in [49], in which a comparison of the effect of two depth map quantification methods has been investigated.



(a) Book arrival

(b) Door flowers

(c) Outdoor

Figure 6.18: Rate-distortion coding results for the QP set [32, 37, 42, 47].

Conclusion and perspectives

This thesis proposed several contributions to the development of an advanced 3D video codec. More specifically, our research focused on the development of a hole-filling strategy for view synthesis, the investigation of the depth video compression by different wavelet filter banks and its impact on the quality of the view synthesis, the development of an MPEG-2-based coding scheme of a video-plus-depth sequence, and, last but not least, the construction and optimization of a dense estimation framework for an H.264-based coding a MVV sequence. Below we summarize the contributions of the thesis work, and then propose some directions for future research.

Synthesis of thesis contributions

I – Hole-filling for novel view synthesis

DIBR technique has been recognized as a promising tool which can synthesize some new "virtual" views from the so-called video-plus-depth data representation. The most important problem in the DIBR process while creating "virtual" views is to deal with the newly exposed areas appearing in the "virtual" images.

Pre-processing of the depth video

We have thus, in the case of a small baseline, proposed pre-processing the depth video before the DIBR process. In order to reduce or completely remove the newly exposed areas an efficient smoothing is necessary for the sharp depth changes near object boundaries. In the meantime it is not necessary to smooth the non-disoccluded areas. Our solution has been based on a weighted Gaussian filter taking into account the distance to the contours. In this way, the geometric distortions and the computation time have been reduced compared to an uniform filtering of the depth video. Experimental results have illustrated the high efficiency of the proposed method.

Depth-aided inpainting

We have also addressed the problem of larger disocclusions by proposing to post-process the warped image based on inpainting techniques, well-known for their abilities to propagate texture and structure along contours of "holes". The proposed algorithm has inherited the Crimini's algorithm, where the depth information has been added in the priority computation and the patch matching. Thus, the proposed method has been developed to be relied on the texture and structure propagation, and in the meantime taking into account the depth information by distinguishing foreground and background parts of the image.

Experimental results show that the visual quality of the inpainted image is improved, specially in preserving the foreground contours.

There are many open issues that warrant future research. For one, we feel that these two contributions can be combined to form a complete framework, where the pre-processing of the depth will reduce the disocclusion area size; followed by a depth-aided inpainting to retrieve the remained disocclusions.

II – Wavelet-based coding of the depth video

3D video transmission is an emerging application raising the problem of efficiently encoding of the depth video, in addition to classical texture video. We have thus investigated the depth video coding via an adaptive wavelet lifting scheme. Long filters in homogeneous areas and short filters over the edges of the depth video are decided based on the contours detected in the texture video. The method took thus into consideration the correlation existing between the edges in the texture and depth videos, leading to an improved encoding of the latter one. Adaptativity has been introduced through the use of the texture contours to switch filters in the depth map decomposition. By applying shorter filters over edges, the energy of the detail coefficients has been reduced and the location of the edges better preserved in the reconstructed depth map. Experimental results illustrated the efficiency of the proposed adaptive lifting to achieve better preservation of the depth edges in the coding process, which also led to a quality improvement of the view synthesis.

III – MPEG-2-based coding of video-plus-depth sequence

Joint motion estimation

The compression efficiency is usually higher for smooth gray level data representing the depth map than for classical video texture. However, improvements of the coding efficiency are still possible, taking into account the fact that the color video and the depth map sequences are strongly correlated. We have thus proposed reducing the amount of information for describing the motion of the texture video and of the depth map sequences by sharing one common motion vector field. According to the MPEG-C Part 3 specifications, the joint motion field has been placed in the MPEG-2 texture stream for backward-compatibility purposes. The experimental results pointed out that optimizing only the texture motion vectors does not lead necessarily to the best view synthesis. Our approach offers a better tradeoff.

Joint bit allocation strategy

Furthermore, in the literature, the bitrate control scheme generally fixes for the depth map sequence a fixed percentage of 10% to 20% of the texture stream bitrate. However, this fixed percentage should depend of the content of each sequence. We have thus proposed a new bitrate allocation strategy between the texture and its associated per-pixel depth information. The proposed rate control algorithm takes into account the motion activity of each texture and depth pictures by calculating the log-variance ratio. The new bitrate allocation allows a better distribution and largely improves the results. The experimental results had shown an overall amelioration of the proposed method compared to the conventional MPEG-2 bitrate allocation with an advantageous amelioration at high bitrate.

IV – Dense motion/disparity estimation in H.264/MVC

We have then addressed the problem of MVV coding and described a joint method for estimating disparity and motion fields involved in the MVV sequence. We have addressed the problem of reducing temporal/inter-view redundancies of key pictures and non-key pictures with disparity estimation and joint motion/disparity estimation. We have thus proposed a dense motion/disparity estimation framework followed by a rate-distortion-driven segmentation and coding designed to replace the block-based motion/disparity estimation stage in MVC extension. This led to an overall improvement of the rate-distortion performances. In order to reduce computational complexity and improve the estimation accuracy, a joint estimation technique has been proposed and addressed in a variational framework, by using an iterative algorithm based on recently developed convex optimization tools. Experimental results involving real sequences indicated the feasibility and robustness of our approach both in terms of reconstruction and consistency of estimated displacement fields.

Moreover, due to the *stereo-motion consistency* constraint, the four displacement vector fields are highly correlated, which make them well suited for inter-fields compression, *e.g.*, with one vector field predicting the others, similar to the technique introduced in the inter-view DIRECT mode [46].

Perspectives

In this thesis work, we have studied and proposed several multiview-based video compression schemes able to exploit the inherent correlation inside an MVV system. Nevertheless, a number of topics can be identified that still require further investigation, and may lead to even better compression performance for the 3D class of video coding algorithms. These include:

Asymmetric view coding

3D video perception requires a pair of views, the left view and the right view, to be presented to the left eye and the right eye of users. The two views can be coded independently, requiring twice the bandwidth of the traditional TV or interview coding techniques that allow prediction across view can be employed to improve compression efficiency. The bitrate required for 3D video services can be substantially reduced if the human visual system is properly exploited. Human visual system has the remarkable ability to compensate for the loss of information in one of the views and still present a very good 3D video perception. This is essentially the case where a person with perfect vision in one eye and a slightly blurry vision in the other eye is able to see the world around him normally. This ability of the human visual system can be exploited to reduce the compression of 3D video services by applying asymmetric view coding. In asymmetric view coding the left and the right eye views are encoded with different qualities without degrading the 3D experience. The goal of this study will be to better understand the bounds of asymmetric coding between the visual perception and 3D quality of asymmetrically coded video, and to understand of the effects of the coder coding artifacts.

View synthesis distortion

Additionally, an efficiency improvement of the 3D video codec can be proposed by studying a distortion model to characterize the view synthesis quality based on subjective quality. 3D perception quality can be regarded as one of the most important issues to be taken into consideration. Not taking the visual system into account is probably one of the serious drawbacks of the above mentioned measures. In the 3D ATTEST chain one can distinguish three categories of technology variables, related to the content generation, to the 3D coding algorithms and to the display system and viewing situation.

The goal of this study would be to propose a distortion model to characterize the view synthesis quality. Three kinds of distortions are mainly concerned, namely, the video-coding-induced distortion, the depth-quantization-induced distortion, and the inherent geometry distortion.

Appendix A

GPU-based photometric reconstruction from screen light

This annexe address the problem of obtaining from one web camera and a computer display, a facial reconstruction of the user within online applications like skype, msn, *etc*, to name but a few of the existing messaging applications. We present a 3D shape recovery in real time based on the photometric information of a set of four images under varying illumination conditions. This work was supported by the Japan Society for the Promotion of Science (JSPS) Fellowship Program and has been realized in collaboration with Keio University.

A.1 Introduction

Enjoy 3D entertainment at home is intended to be one of the new promising communication services. The development of digital TV and autostereoscopic displays allows to easily insert stereoscopic technologies at home. We expect in a near future that every household will be equipped with such equipments. In that way we designed a 3D reconstruction application based on cheap and accessible devices that allows users to communicate via online applications like skype, msn with a 3D perception of their interlocutor, as shown in Fig. A.1 by only means of a computer screen and a web camera.

A.1.1 3D reconstruction

There exist various techniques to perform 3D reconstruction from videos. Some of them can work in real-time and most of them require several calibrated cameras. Depth from stereo methods [101] compute a disparity map from point correspondences. The visual hulls [70] method extracts the silhouette of the main object of the scene on the images from every camera. The 3D shape of this object is then approximated by the intersection of the projected silhouettes. The plane-sweep algorithm can compute a 3D reconstruction of the scene in real-time [90] using a discretization of the scene with parallel planes.

Some other methods can perform a 3D reconstruction from a single camera. Optical flow methods [61] analyze the motion of the objects in the scene to recover the 3D information. Finally, 3D reconstruction can be performed by radiometric techniques. These methods require several images of the same scene under different lighting conditions to extract the 3D shape of the scene. We propose focusing on the latter family, and expose the related works in the following.

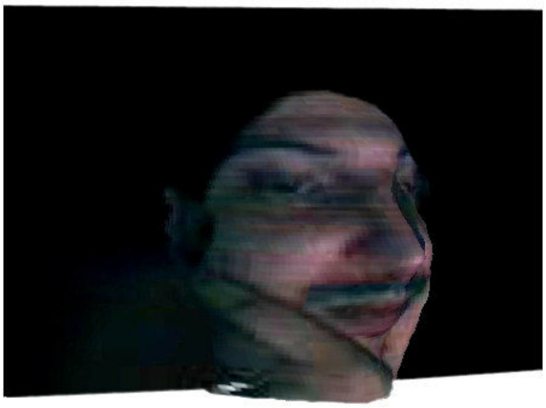


Figure A.1: 3D facial reconstruction.

A.1.2 Previous work

In the past decades, intense interest in photometric stereo problem has produced many excellent works for establishing the theoretical part. The idea of photometric stereo, first introduced by Woodham [134], is to determine the surface orientation at each point by varying the direction of incident illumination while holding the same point of view.

The main and difficult part is to find a way to map the of RGB intensity to a normal map. To overcome this issue, experimental methods have been investigated by Christensen and Shapiro [26] and Hertzmann and Seitz [51]. They build for each material surface a look-up table with general reflectance properties. The main drawback of these methods concern the assumption that objects have homogeneous surfaces, which is not workable for complex objects. They proposed, however, a full segmentation into different materials.

More recently, Hernandez and al. [50] have presented work on using spatially separated red, green and blue light sources to estimate a dense depth map from a untextured non-rigid surface. By using a calibration tool, a mapping RGB intensity to normal map is provided, and thus, the depth map is obtained by integration of the normal map.

From the small baseline multi-flash camera used in [40] is possible to compute first the depth edge. In fact the flash illumination allows to measure the cast shadow width. Based on this measurement, a gradient map field is provided, which is integrated by using a Poisson equation.

As previously seen, the knowledge of the lighting conditions are commonly necessary. Hallinan [47] overcomes the issue of not knowing the lighting conditions by proposing a low-dimensional illumination representation of human face under arbitrary light conditions. Given a set face image, the lighting conditions are estimated by using principal component analysis in an image basis.

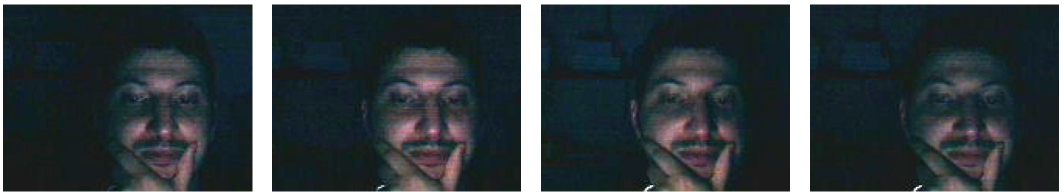
Finally, for biochemistry purposes, a simple system has been proposed by Filippini *et al.* [41] consisting of using a computer screen as a programmable light source, working with a web camera which captures the visible absorption features of samples as chemical image.

Next, we present each step of the proposed framework for 3D geometry recovery from input images are discussed, followed by a description of the real time implementation on a GPU.

A.2 Our Approach

In this section we present in detail the recovery of the 3D structure of a human face by photometric means. By using four light sources via the computer screen, as shows the input image in Fig. A.2, the normal surface map is estimated by using an image basis issued from the lighting conditions. Then, the depth map is computed from the normal surface map integration.

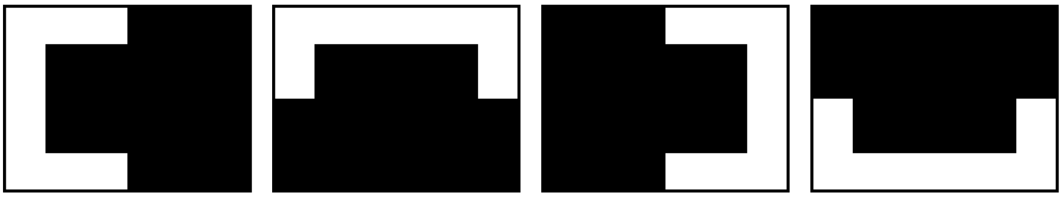
A.2.1 Lighting from the screen



(a) Left (b) Top (c) Right (d) Bottom

Figure A.2: Input images under varying illumination.

The computer screen is used as a large programmable light source area and provides various lighting conditions for the photometric reconstruction. The illuminated scene is captured by a camera attached on the screen and a reconstruction is performed for every new captured frame. Indeed, the new image is transfered to the system and added to the latest captured image set. To optimize the quality of the reconstruction, every image of the set should be taken under different lighting conditions and the number of images should be as big as possible. However, using too many images with a dynamic scene will lead to a latency on the reconstruction process since every image should correspond to the same scene geometry. Moreover, using too many images may increase the computation time and prevent from real-time rendering. Considering these constraints, using four input images and hence four lighting conditions seems to be a good compromise. To ensure enough lighting, the light source should not be punctual and since the light direction should be roughly known, we choose to illuminate alternatively every top, right, bottom and left half part of the screen, as shown on Fig. A.3.



(a) Left (b) Top (c) Right (d) Bottom

Figure A.3: Different lighting conditions on the computer screen.

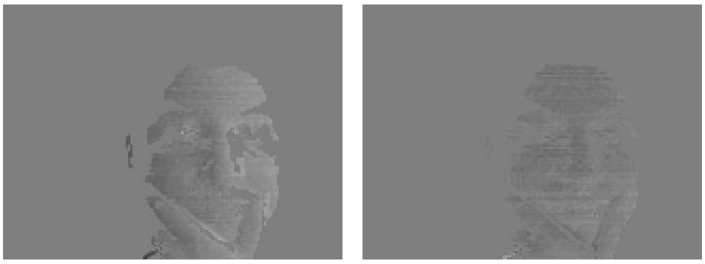
As mentioned above, the system does not have to wait for four new views between two consecutive reconstructions. The latest captured image will update the image set by replacing the existing image under the same lighting condition. Naturally, this approach involves synchronization between the screen and the camera.

Finally, the ambient light of the scene should be reduced to the minimum to maximize the screen light contribution.

A.2.2 Surface normal map

Given an image of a scene, a surface normal map associates the surface normal vectors to every pixel of the image. As presented by Yuille *et al.* in [139], the Singular Value Decomposition (SVD) can be used to compute a surface normal map from a set of N images I_j ($j = 1...N$) of dimension $width \times height$. The N input images are converted to grey scales images and considered as 1 dimensional arrays. A matrix $\boldsymbol{A}$ with N rows and $width \times height$ columns is defined such that every row of A corresponds to an input image. The SVD of $\boldsymbol{A}\boldsymbol{A}^\top$ provides a set of eigen vectors that determines for every pixel the contribution coefficient of the input images to create the normal map. According to the big size of $\boldsymbol{A}\boldsymbol{A}^\top$, the computation time required for the SVD may prevent from real-time rendering. As presented in [139], an alternative to this approach is to compute the SVD of $\boldsymbol{A}^\top\boldsymbol{A}$ which is an $N \times N$ matrix. This approach is much faster, however the eigen vector information is common for every pixel of the image.

Since every input image is different, the SVD of $\boldsymbol{A}^\top\boldsymbol{A}$ will provide N orthogonal eigen vectors. The eigen vector associated to the biggest eigen value corresponds to the z-axis. The two next biggest eigen values correspond to two other orthogonal directions. Since we arranged our light system to be oriented only in vertical and horizontal directions, these two eigen values will correspond to the x and y axis. We can identify which of the two eigen vectors corresponds to the x and y axis by comparing the provided coefficient for every input image. The x axis eigen vector will provide a high coefficient for the images associated with the left and right lighting while the y axis eigen vector will give high values for the top and bottom lighting. To check the identification, the i^{th} component of the x, y and z eigen vectors should correspond to the light orientation of the image of the i^{th} line of $\boldsymbol{A}$.



(a) X component (b) Y component

Figure A.4: Example of normal surface map.

Finally, for every pixel of the camera image, we compute a normal vector. The x component (respectively the y and z component) is given by the dot product of the x axis eigen vector (respectively the y and z vector) with the column of $\boldsymbol{A}$ corresponding to the current pixel as shown in Fig. A.4. For the depth map recovery, the normal vectors should be normalized. However, since the eigen vectors of $\boldsymbol{A}^\top\boldsymbol{A}$ are common for all the pixels, some resulting normal vectors may be null. These vectors should be detected to avoid mistakes during the depth map recovery process.

A.2.3 Depth map computation

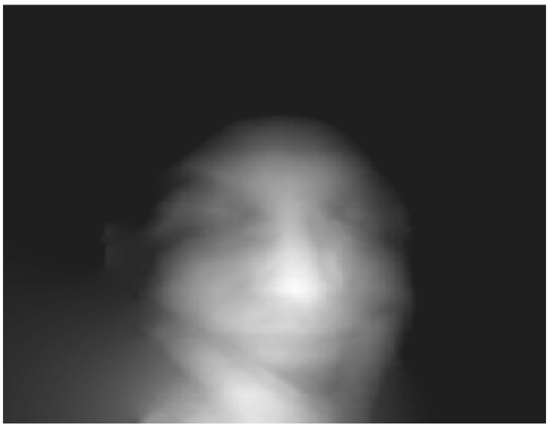


Figure A.5: Depth map.

Let us first introduce the relationship between the normal surface map and the depth map. The depth value z of a object at the position (x, y) can be expressed by a depth map with the following equation:

$$z = f(x, y) \tag{A.1}$$

By using this notation, it is possible to define for each surface position the normal vector $\mathbf{n}(n_x, n_y, -n_z)$, defined by the gradient of $f(x, y)$ as follows:

$$\frac{n_x}{n_z} = \frac{\partial f(x, y)}{\partial x} \tag{A.2a}$$

$$\frac{n_y}{n_z} = \frac{\partial f(x, y)}{\partial y} \tag{A.2b}$$

Let us define the quantity (p, q) as $p = \frac{n_x}{n_z}$ and $q = \frac{n_y}{n_z}$. Thus, the recovery of the surface can be expressed as the minimization of the following expression:

$$\sum_i \sum_j \left(z_{i+1,j} - z_{i,j} - p_{i,j}\right)^2 + \left(z_{i,j+1} - z_{i,j} - q_{i,j}\right)^2 \tag{A.3}$$

which leads to the following iterative scheme:

$$z_{i,j}^{k+1} = \frac{1}{4}\left[z_{i+1,j}^k + z_{i-1,j}^k + z_{i,j+1}^k + z_{i,j-1}^k - p_{i,j} + p_{i-1,j} - q_{i,j} + q_{i,j-1}\right]. \tag{A.4}$$

As suggested in [102], we choose to solve this equation using Gauss-Seidel method. Indeed, even if an iterative method does not guarantee the best accuracy, it presents the advantage to accept as an initial solution the depth map of the previous frame, which leads to a fast convergence. Moreover, Gauss-Seidel relaxation is very well suited for a GPU implementation since all the pixels can be processed simultaneously. Finally, to ensure a constant reference depth for all the reconstructions, the relaxation process is not applied on the border of the depth map.

A.3 Implementation

For each new reconstruction, a new black and white pattern is displayed on the screen. Then, the scene is captured by the camera. The camera-screen synchronization is a critical issue to solve, especially with webcams using a streaming mode. In our system, we used Video for Linux and used a read method that waits until it receives the queried image. This approach is not slower than the streaming method and contributes to an accurate synchronization.

The latest grabbed image is converted in grey scale and inserted in the matrix A. Since the camera and the screen are synchronized, we know the image lighting conditions and thus can insert this image on the corresponding line of A. Hence, this method prevents from inserting in A two images with the same light conditions.

The $A^\top A$ matrix does not require a full computation to be updated. Actually, only one row and one column should be updated and since $A^\top A$ is symmetric, these row and column can be updated simultaneously.

The eigen vectors and surface normal maps are computed on the CPU and transfered to the GPU for the relaxation step. The Gauss-Seidel iterations are performed off-screen by the GPU using frame buffer object (FBO). Two textures are used alternatively to contain the depth map of the previous iteration or to be attached to the FBO for the current iteration rendering. The relaxation process uses the equation presented in Section A.2.3.

Finally, a dense flat mesh is projected on the screen. Every vertex depth is modified according to the depth map using a vertex program. Then the meshes are textured with the input images with a fragment program.

This method does not require any transfer of the depth map between the GPU and the main memory.

A.4 Experimental results

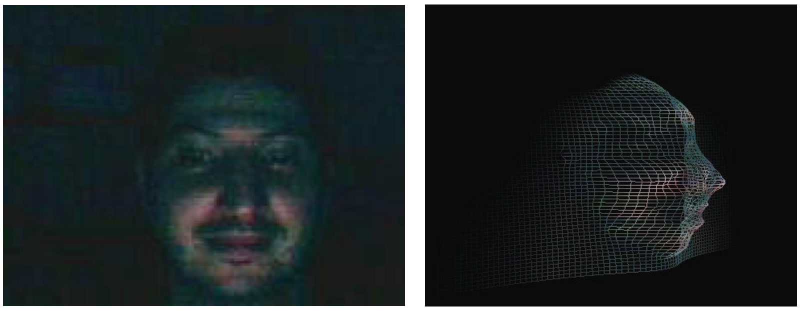
We have implemented our system on a PC Intel core duo 1.86 GHz with an nVidia GeForce 7900 GTX. The video acquisition is performed by one USB Logitech fusion web camera connected to the computer. With a 320×240 resolution, the acquisition frame rate reaches 15 frames per second. All the computations are made within the image size of 320×240 pixels.

The frame rate of the web camera is high enough to satisfy the requirements of the photometric fixed viewpoint restriction of the subject for every four images, corresponding to the light rotation period.

The GPU parallelism computation allows for the depth map recovery about 100 iterations to obtain a convergent solution with Gauss-Seidel. Actually, the fps limitations is only due the web camera hardware limitations.

For more accuracy, the subject should be far from the camera to reproduce the orthographic projection. A distance trade-off is made such that the subject receives enough light from the screen.

The photometric stereo method of Section A.2 was applied. The results in Fig. A.6 show the facial reconstruction of a subject at a distance from the screen of about 45cm.



(a)

(b)

Figure A.6: 3D geometry surface recovery: (left) input image, (right) 3D surface.

A.5 Conclusion

In this chapter we presented a real-time implementation on GPU of a 3D facial reconstruction destined to home use public application by simply using a standard web camera and the computer screen. By knowing the lighting conditions, it is possible to reconstruct the 3D geometrical surface with only one webcam.

Thanks to our GPU implementation of the relaxation step and the multi-texturing blending, this method can reach real-time rendering. However a limitation of this method concerns the screen light contribution that must be predominant over the ambient light.

Publications

International Journal papers

1. Ismaël Daribo, Wided Miled, and Béatrice Pesquet-Popescu. “Joint depth-motion dense estimation for multiview video coding”. in Journal on Visual Communication and Image Representation Special Issue on Multi-Camera Imaging, Coding and Innovative Display: Techniques and Systems, 2010.

2. Ismaël Daribo, Christophe Tillier, and Béatrice Pesquet-Popescu. “Motion vector sharing and bit-rate allocation for 3D video-plus-depth coding”. *EURASIP Journal on Applied Signal Processing*, 2009:13 pages, 2009. Special Issue on 3DTV. Article ID 258920.

International Conference papers

1. Ismaël Daribo, Mounir Kaaniche, Wided Miled, Marco Cagnazzo, and Béatrice Pesquet-Popescu. “Dense disparity estimation in multiview video coding”. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, Rio de Janeiro, Brazil, October 2009.

2. Vincent Nozick, Ismaël Daribo, and Hideo Saito. “GPU-based photometric reconstruction from screen light”. In *Proc. of the 18th International Conference on Artificial Reality and Telexistence (ICAT)*, pages 242–245, Yokohama, Japan, December 2008.

3. Ismaël Daribo, Christophe Tillier, and Béatrice Pesquet-Popescu. “Adaptive wavelet coding of the depth map for stereoscopic view synthesis”. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 413–417, Cairns, Queensland, Australia, 2008.

4. Ismaël Daribo, Christophe Tillier, and Béatrice Pesquet-Popescu. “Distance dependent depth filtering in 3D warping for 3DTV”. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 312–315, Crete, Greece, October 2007.

Domestique Conference papers

1. Wided Miled, Ismaël Daribo, and Béatrice Pesquet-Popescu. “Estimation conjointe disparité mouvement pour le codage de séquences vidéo multi-vues”. In *22éme Colloque GRETSI*, Dijon, France, September 2009.

2. Ismaël Daribo, Christophe Tillier, and Béatrice Pesquet-Popescu. “Filtrage de la profondeur pour le plaquage de texture dans la télévision 3D”. In *COmpression et REprésentation des Signaux Audiovisuels (CORESA)*, 2007.

List of Figures

List of Tables

Bibliography

[1] 3D Consortium. [Online] Availaible: http://www.3dc.gr.jp/english.

[2] 3D4YOU. [Online] Availaible: http://www.3d4you.eu.

[3] 3DPHONE. [Online] Availaible: http://www.3dphone.org.

[4] 3DTV network of excellence. [Online] Availaible: http://www.3dtv-research.org.

[5] Digital bitrate. [Online] Availaible: http://www.digitalbitrate.com.

[6] Generic coding of moving pictures and associated audio information – part 2: Video. ITU-T and ISO/IEC JTC 1, November 1994. Recommendation H.222.0 and ISO/IEC 13 818-2 (MPEG-2 Video).

[7] Sequence microsoft ballet and breakdancers, 2004. [Online] Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/.

[8] Survey of algorithms used for multi-view video coding (MVC). N6909 doc., Hong Kong, China, January 2005.

[9] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, April 1992.

[10] Raphaèle Balter, Patrick Gioia, and Luce Morin. Scalable and efficient coding using 3D modeling. *IEEE Transactions on Multimedia*, 8(6):1147–1155, 2006.

[11] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–355–I–362 vol.1, 2001.

[12] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, New Orleans, USA, July 2000.

[13] G. Bjontegaard. Calculation of average PSNR differences between RD curves, April 2001. ITU SC16/Q6, 13th VCEG Meeting, Austin, Texas, USA, VCEG-M33 doc.

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (11):1222–1239, 2001.

[15] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984. new edition.

[16] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans peter Seidel. Free-viewpoint video of human actors. In *Proc. of the ACM Transactions on Graphics (TOG)*, volume 22, pages 569–577, New York, NY, USA, 2003.

[17] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding, and Liang-Gee Chen. Efficient depth image based rendering with edge dependent depth filter and interpolation. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1314–1317, 6-8 2005.

[18] Ying Chen, Ye-Kui Wang, and Miska M. Hannuksela. On MVC reference picture list construction. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-V043 doc., Marrakech, Morocco, January 2007.

[19] Ying Chen, Ye-Kui Wang, and Miska M. Hannuksela. View instantaneous decoding refresh (V-IDR) picture. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-X029 doc., Geneva, Switzerland, July 2007. Comments to MVC JD 3.0.

[20] Ying Chen, Ye-Kui Wang, Miska M. Hannuksela, Shujie Liu, and Houqiang Li. On MVC reference picture marking. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-V044 doc., Marrakech, Morocco, January 2007.

[21] Ying Chen, Ye-Kui Wang, and M.M. Hannuksela. Single-loop decoding and motion skip study in JMVM. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-Y053 doc., Shenzhen, China, October 2006.

[22] Ying Chen, Ye-Kui Wang, Kemal Ugur, MiskaM. Hannuksela, Jani Lainema, and Moncef Gabbouj. The emerging MVC standard for 3D video services. *EURASIP Journal on Advances in Signal Processing*, 2009, 2009. Article ID 786015, 13 pages.

[23] Chia-Ming Cheng, Shu-Jyuan Lin, Shang-Hong Lai, and Jinn-Cherng Yang. Improved novel view synthesis from depth image with large baseline. In *Proc. of the 19th International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[24] Sukhee Cho, Kugjin Yun, Byungjun Bae, and Youngkow Hahm. Disparity-compensated coding using MAC for stereoscopic video. In *Proc. of the IEEE International Conference on Consumer Electronics (ICCE)*, pages 170–171, 17-19 June 2003.

[25] P.A. Chou, T. Lookabaugh, and R.M. Gray. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Transactions on Information Theory*, 35(2):299–315, March 1989.

[26] Per H. Christensen and Linda G. Shapiro. Three-dimensional shape from color photometric stereo. *International Journal of Computer Vision*, 13(2):213–227, 1994.

[27] A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–500, 1992.

[28] P.L. Combettes. A block-iterative surrogate constraint splitting method for quadratic signal recovery. *IEEE Transactions on Signal Processing*, 51(7):1771–1782, 2003.

[29] P.L. Combettes and J.-C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9):1213–1222, 2004.

[30] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9): 1200–1212, 2004.

[31] Ismaël Daribo, Mounir Kaaniche, Wided Miled, Marco Cagnazzo, and Béatrice Pesquet-Popescu. Dense disparity estimation in multiview video coding. In *Proc. of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, Rio de Janeiro, Brazil, October 2009.

[32] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4:247–269, 1998.

[33] Frederic Devernay and Olivier D. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[34] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of the 7th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038 vol.2, 1999.

[35] Christoph Fehn. A 3D-TV approach using depth-image-based rendering (DIBR). In *Proceedings of VIIP*, Benalmadena, Spain, September 2003.

[36] Christoph Fehn. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *Proc. of the SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pages 93–104, San Jose, CA, USA, January 2004.

[37] Christoph Fehn, Eddie Cooke, Oliver Schreer, and Peter Kauff. 3D analysis and image-based rendering for immersive TV applications. *Signal Processing: Image Communication*, 17(9):705 – 715, 2002.

[38] Christoph Fehn, Klaas Schüür, Ingo Feldmann, Peter Kauff, and Aljoscha Smolic. Distribution of ATTEST test sequences for EE4 in MPEG 3DAV. ISO/IEC JTC1/SC29/WG11, M9219 doc., December 2002.

[39] I. Feldmann, M. Müller, F. Zilly, R. Tanger, K. Muller, A. Smolic, P. Kauff, and T. Wiegand. HHI test material for 3D video. , M15413 doc., May 2008.

[40] Rogerio Feris, Ramesh Raskar, Longbin Chen, Karhan Tan, and Matthew Turk. Discontinuity preserving stereo with small baseline multi-flash illumination. In *IEEE International Conference in Computer Vision (ICCV'05)*, Beijing, China, 2005.

[41] Filippini, D. Svensson, and I. S. P. S. Lundstrom. Computer screen as a programmable light source for visible absorption characterization of (bio)chemical assays. *CHEMICAL COMMUNICATIONS- ROYAL SOCIETY OF CHEMISTRY*, 2: 240–241, 2003.

[42] Borko Furht. *Encyclopedia of Multimedia*. Springer, second edition, 2008.

[43] O.P. Gangwal and R.-P. Berretty. Depth map post-processing for 3D-TV. In *Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, pages 1–2, January 2009.

[44] Yaorong Ge and J.M. Fitzpatrick. On the generation of skeletons from discrete euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1055–1066, 1996.

[45] S. Grewatsch and E. Müller. Sharing of motion vectors in 3D video coding. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 3271–3274, Singapore, 24-27 October 2004.

[46] Xun Guo, Yan Lu, Feng Wu, and Wen Gao. Inter-view direct mode for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16 (12):1527–1532, 2006.

[47] P.W. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pages 995–999, June 1994.

[48] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[49] Renlong He, Mei Yu, You Yang, and Gangyi Jiang. Comparison of the depth quantification method in terms of coding and synthesizing capacity in 3DTV system. In *Proc. of the 9th International Conference on Signal Processing (ICSP)*, pages 1279–1282, Leipzig, Germany, May 2008.

[50] C. Hernandez, G. Vogiatzis, G.J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. IEEE 11th International Conference on Computer Vision ICCV 2007*, pages 1–8, 2007.

[51] A. Hertzmann and S.M. Seitz. Shape and materials by example: a photometric stereo approach. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–533–I–540 vol.1, 2003.

[52] Yo-Sung Ho, Kwan-Jung Oh, and Cheon Lee. Regional disparity derivation for motion skip mode. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-Z030 doc., Antalya, Turkey, January 2008.

[53] G.J. Iddan and G. Yahav. 3D imaging in the studio (and elsewhere...). *SPIE : Videometrics and Optical Methods for 3D Shape Measurements*, 4298:48–55, 2001.

[54] ISO/IEC JTC1/SC29/WG11. Test model 5, April 1993. MPEG 93/457 doc.

[55] ISO/IEC JTC1/SC29/WG11. MPEG-4 animation framework extension (AFX) VM 10.0. , N5393 doc., December 2002.

[56] ITU. Methodology for the subjective assessment of the quality of television pictures. ITU-R Recommendation BT.500-10 doc., 1974-2002.

[57] ITU-T Recommentation H.264 & ISO/IEC 14496-10 AVC. Advanced video coding for generic audio-visual services, 2005. version 3.

[58] V. Jantet, L. Morin, and C. Guillemot. Incremental-LDI for multi-view coding. In *Proc. of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4, Potsdam, Germany, 2009.

[59] Yong-Hwan Kim Ji Ho Park, Jewoo Kim, Byeong-Ho Choi Yatap dong Bundang-gu, and Seongnam si Gyeonggi-do. Motion skip mode with residual prediction. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-Z031 doc., Antalya, Turkey, January 2008.

[60] KwangHee Jung, Young Kyung Park, Joong Kyu Kim, Hyun Lee, K. Yun, N. Hur, and Jinwoong Kim. Depth image based rendering for 3D data service over T-DMB. In *Proc. of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 237–240, 2008.

[61] S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. In *Proc. of the annual Conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 319–325, 2003.

[62] H.A. Karim, S. Worrall, A.H. Sadka, and A.M. Kondoz. 3-D video compression using MPEG4-multiple auxiliary component (MPEG4-MAC). In *Proc. of the Visual Information Engineering (VIE)*, April 2005.

[63] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. of the 8th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515 vol.2, 2001.

[64] Han-Suh Koo, Yong-Joon Jeon, and Byeong-Moon Jeon. MVC motion skip mode. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-W081 doc., San Jose, California, USA, April 2007.

[65] D. Le Gall and A. Tabatabai. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 761–764, 11–14 April 1988.

[66] Yung-Lyul Lee, Jae-Ho Hur, Yung-Ki Lee, Ki-Hun Han, SukHee Cho, NamHo Hur, JinWoong Kim, Jae-Hoon Kim, Po-Lin Lai, Antonio Ortega, Yeping Su, and Peng Yin andCristina Gomila. CE11: Illumination compensation. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-U052 doc., Hangzhou, China, October 2006.

[67] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *Proc. of the 9th IEEE International Conference on Computer Vision*, pages 305–312 vol.1, 2003.

[68] J. Liu and R. Skerjanc. Stereo and motion correspondence in a sequence of stereo images. *Signal Processing: Image communication*, 5:305–318, October 1993.

[69] Yanwei Liu, Qingming Huang, Siwei Ma, Debin Zhao, and Wen Gao. Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model. *Signal Processing: Image Communication*, 24(8):666–681, 2009.

[70] M. A. Magnor. *Video-based Rendering.* A K Peters Ltd, 2005.

[71] Mathieu Maitre, Christine Guillemot, and Luce Morin. 3D scene modeling for distributed video coding. *IEEE Transactions on Image Processing*, 16(5):1246–1257, 2007.

[72] Matthieu Maitre, Yoshihisa Shinagawa, and Minh N. Do. Rate-distortion optimal depth maps in the wavelet domain for free-viewpoint rendering. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 125–128, 2007.

[73] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7): 674–693, 1989.

[74] William Mark. *Post-Rendering 3D Image Warping: Visibility, Reconstruction, and Performance for Depth-Image Warping.* PhD thesis, University of North Carolina at Chapel Hill, NC, USA, April 1999.

[75] William R. Mark, Leonard McMillan, and Gary Bishop. Post-rendering 3D warping. In *Proc. of the Symposium on Interactive 3D Graphics (SI3D)*, pages 7–16, New York, USA, 1997. ACM Press. ISBN 0-89791-884-3.

[76] Detlev Marpe, Thomas Wiegand, and Stephen Gordon. H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages I–593–6, Genoa, Italy, September 2005.

[77] Leonard McMillan, Jr. *An Image-Based Approach to Three-Dimensional Computer Graphics.* PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1997.

[78] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand. The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communication*, 24(1-2):73–88, 2009. Special issue on advances in three-dimensional television and video.

[79] P. Merkle, K. Müller, A. Smolic, and T. Wiegand. Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1717–1720, 9–12 July 2006.

[80] P. Merkle, A. Smolic, K. Müller, and T. Wiegand. Efficient prediction structures for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1461–1473, 2007.

[81] W. Miled, J.-C. Pesquet, and M. Parent. A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4):813–830, 2009.

[82] Wided Miled. *Mise en correspondance stéréoscopiques par approches variationnelles convexes; application à la détection d'obstacles routiers.* PhD thesis, Université de Paris-Est, Marne-la-Vallée, France, 2008.

[83] Wided Miled, Beatrice Pesquet-Popescu, and Wael Cherif. A variational framework for simultaneous motion and disparity estimation in a sequence of stereo images. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 741–744, Taipei, Taiwan, April 2009.

[84] Dongbo Min, Hansung Kim, and Kwanghoon Sohn. Edge-preserving joint motion-disparity estimation in stereo image sequences. *Signal Processing: Image Communication*, 21(3):252–271, 2006.

[85] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto. View generation with 3D warping using depth information for FTV. In *Proc. 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 229–232, 2008.

[86] Y. Morvan, D. Farin, and P. De With. System architecture for free-viewpoint video and 3D-TV. 54(2):925–932, 2008.

[87] Yannick Morvan, Dirk Farin, and Peter H. N. de With. Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 105–108, San Antonio (TX), USA, September 2007.

[88] Yannick Morvan, Dirk Farin, and Peter H. N. de With. Joint depth/texture bit-allocation for multi-view video compression. In *Proc. of the Picture Coding Symposium (PCS)*, Lisboa, Portugal, November 2007.

[89] M. Nalasani and W.D. Pan. Performance evaluation of MPEG-2 codec with accurate motion estimation. In *Proc. of the Thirty-Seventh Southeastern Symposium on System Theory (SSST)*, pages 287–291, Tuskegee, Alabama, March 2005.

[90] Vincent Nozick and Hideo Saito. On-line free-viewpoint video : From single to multiple view rendering. *Journal of Automation and Computing (IJAC)*, 5(3):257–267, 2008.

[91] Committee Draft of ISO/IEC JTC1/SC29/WG11 23002-3. Auxiliary video data representations, April 2006. N8038 doc. Montreux, Switzerland.

[92] Han Oh and Yo-Sung Ho. H.264-based depth map sequence coding using motion information of corresponding texture video. In *Advances in Image and Video Technology*, pages 898–907, 2006.

[93] Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. In *Proc. of the Picture Coding Symposium (PCS)*, pages 1–4, 2009.

[94] Manuel M. Oliveira. *Relief Texture Mapping*. PhD thesis, University of North Carolina at Chapel Hill, NC, USA, 2000.

[95] I. Patras, N. Alvertos, and G. Tziritas. Joint disparity and motion field estimation in stereoscopic image sequences. In *Proc. of the 13th International Conference on Pattern Recognition*, volume 1, pages 359–363, Washington, DC, USA, August 1996.

[96] D.E. Pearson. Developments in model-based video coding. 83(6):892–906, 1995.

[97] F. Pedersini, A. Sarti, and S. Tubaro. Multi-camera systems. *IEEE Signal Processing Magazine*, 16(3):55–65, 1999.

[98] A. Redert, M.O. de Beeck, C. Fehn, W. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, I. Sexton, and P. Surman. Advanced three-dimensional television system technologies. In *Proc. First International Symposium on 3D Data Processing Visualization and Transmission*, pages 313–319, June 2002.

[99] O. Rioul and P. Duhamel. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Transactions on Information Theory*, 38(2):569–586, 1992.

[100] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[101] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc. of the IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*, pages 131–140, 2001.

[102] Grant Schindler. Photometric stereo via computer screen lighting for real-time surface reconstruction. In *Proceedings of 3DPVT - the 4th Internation Symposium on 3D Data Processing, Visualisation and Transmission*, 2008.

[103] Klaas Schüür, Christoph Fehn, Peter Kauff, and Aljoscha Smolic. About the impact of disparity coding on novel view synthesis. , MPEG02/M8676 doc., Klagenfurt, July 2002.

[104] Jonathan W. Shade, Steven J. Gortler, Li-Wei He, and Richard Szeliski. Layered depth images. *Computer Graphics*, 32(Annual Conference Series):231–242, July 1998.

[105] Manson Siu, Yuk-Hee Chan, and Wan-Chi Siu. A robust model generation technique for model-based video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(11):1188–1192, 2001.

[106] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, and A. Koz. Coding algorithms for 3DTV - a survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1606–1621, 2007.

[107] Aljoscha Smolic, Karsten Mueller, Philipp Merkle, Nicole Atzpadin, Christoph Fehn, Markus Mueller, Oliver Schreer, Ralf Tanger, Peter Kauff, and Thomas Wiegand. Multi-view video plus depth (MVD) format for advanced 3D video systems. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-W100 doc., San Jose, California, USA, April 2007.

[108] L. Stelmach and Wa James Tam. Stereoscopic image coding: Effect of disparate image-quality in left- and right-eye views. *Signal Processing: Image Communication*, 14:111–117, 1998.

[109] L. Stelmach, Wa James Tam, D. Meegan, and A. Vincent. Stereo image quality: effects of mixed spatio-temporal resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2):188–193, 2000.

[110] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, November 1998.

[111] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (7):787–800, 2003.

[112] Wim Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In *Proc. of the SPIE, Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 68–79, 1995.

[113] Wim Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

[114] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal of Mathematical Analysis*, 29:511–546, 1997.

[115] Wa James Tam, Guillaume Alain, Liang Zhang, Taali Martin, and Ronald Renaud. Smoothing depth maps for improved steroscopic image quality. In Bahram Javidi and Fumio Okano, editors, *Three-Dimensional TV, Video, and Display III*, volume 5599, pages 162–172, 2004.

[116] A. Tamtaoui and C. Labit. Coherent disparity and motion compensation in 3DTV image sequence coding schemes. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 2845–2848, Toronto, Ont., Canada, April 1991.

[117] M. Tanimoto. Overview of free viewpoint television. *Signal Processing: Image Communication*, 21:454–461, 2006.

[118] Siping Tao, Ying Chen, M.M. Hannuksela, Ye-Kui Wang, M. Gabbouj, and Houqiang Li. Joint texture and depth map video coding based on the scalable extension of H.264/AVC. In *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2353–2356, 2009.

[119] Z. Tauber, Ze-Nian Li, and M.S. Drew. Review and preview: Disocclusion by inpainting for image-based rendering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(4):527–540, July 2007.

[120] D. S. Taubman and M. W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice.* Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[121] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics, GPU and Game Tools*, 9(1):23–34, 2004.

[122] Thorsten Thormählen and Hellward Broszio. Automatic line-based estimation of radial lens distortion. *Integrated Computer-Aided Engineering*, 12:177–190, 2005.

[123] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of th 6th International Conference on Computer Vision (ICCV)*, pages 839–846, 1998.

[124] Alexis Michael Tourapis, Athanasios Leontaris, Karsten Sühring, and Gary Sullivan. H.264/14496-10 AVC reference software manual. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-AA010 doc., London, July 2009.

[125] A.M. Tourapis, Feng Wu, and Shipeng Li. Direct mode coding for bipredictive slices in the H.264 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):119–126, 2005.

[126] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. 3(4):323–344, 1987.

[127] Anthony Vetro. Mvc profile/level definitions for stereo. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-AB037 doc., Hannover, July 2008.

[128] Anthony Vetro, Purvin Pandit, Hideaki Kimata, Aljoscha Smolic, and Ye-Kui Wang. Joint draft 7.0 on multiview video coding. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-AA209 doc., Geneva, April 2008.

[129] Anthony Vetro, Purvin Pandit, Hideaki Kimata, Aljoscha Smolic, and Ye-Kui Wang. Joint multiview video model (JMVM) 8.0. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-AA207 doc., Geneva, April 2008.

[130] Nicholas Wade. *A Natural History of Vision*. MIP Press, Cambridge, Massachusetts, 1998.

[131] H. Weiler, A. Mitiche, and A. Mansouri. Boundary preserving joint estimation of optical flow and disparity in a sequence of stereoscopic images. In *Proc. on Visualisation, Imaging and Image Processing*, pages 102–106, Malaga, Spain, September 2003.

[132] Charles Wheatstone. Contributions to the physiology of vision – part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, 1838.

[133] Charles Wheatstone. Contributions to the physiology of vision – part the second. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 142:1–17, 1852.

[134] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. pages 513–531, 1989.

[135] S. Yaguchi and H. Saito. Arbitrary viewpoint video synthesis from multiple uncalibrated cameras. *IEEE Trans. on Systems, Man and Cybernetics*, 34:430–439, 2004. PartB.

[136] Haitao Yang, Yilin Chang, Junyan Huo, Sixin Lin, Shan Gao, and Lianhuan Xiong. CE1: Fine motion matching for motion skip mode in MVC. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-Z021 doc., Antalya, Turkey, January 2008.

[137] W. Yang, K. Ngan, J. Lim, and K. Sohn. Joint motion and disparity fields estimation for stereoscopic video sequences. *Signal Processing: Image Communication*, 20(3): 265–276, 2005.

[138] Sehoon Yea and A. Vetro. RD-optimized view synthesis prediction for multiview video coding. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, volume 1, San Antonio, Texas, USA, October 2007.

[139] A. Yuille, D. Snow, R. Epstein, and P. Belhumeur. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability. *Intl. J. of Computer Vision*, 35(3):203–222, 1999.

[140] Liu Zhan-wei, An Ping, Liu Su-xing, and Zhang Zhao-yang. Arbitrary view generation based on DIBR. In *Proc. of the International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 168–171, 2007.

[141] Liang Zhang and W.J. Tam. Stereoscopic image generation based on depth images for 3D TV. *IEEE Transactions on Broadcasting*, 51(2):191–199, June 2005.

[142] Gang Zhu, Xiaozhong Xu, Ping Yang, Yun He, Jianhua Zheng, and Xiaozhen Zheng. Inter-view skip mode with depth information. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-Z029 doc., Antalya, Turkey, January 2008.

[143] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM SIGGRAPH and ACM Transaction on Graphics*, 23(3):600–608, August 2004.