3D-TV CONTENT GENERATION AND MULTI-VIEW VIDEO CODING

by

Mahsa Talebpourazad

B.A.Sc. Iran University of Science and Technology, 2000
M.A.Sc., University of Manitoba, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

June 2010

© Mahsa Talebpourazad, 2010

ABSTRACT

The success of the 3D technology and the speed at which it will penetrate the entertainment market will depend on how well the challenges faced by the 3D-broadcasting system are resolved. The three main 3D-broadcasting system components are 3D content generation, 3D video transmission and 3D display. One obvious challenge is the unavailability of a wide variety of 3D content. Thus, besides generating new 3D-format videos, it is equally important to convert existing 2D material to the 3D format. This is because the generation of new 3D content is highly demanding and in most cases, involves post-processing correction algorithms. Another major challenge is that of transmitting a huge amount of data. This problem becomes much more severe in the case of multiview video content.

This thesis addresses three aspects of the 3D-broadcasting system challenges.

Firstly, the problem of converting 2D acquired video to a 3D format is addressed. Two new and efficient methods were proposed, which exploit the existing relationship between the motion of objects and their distance from the camera, to estimate the depth map of the scene in real-time. These methods can be used at the transmitter and receiver-ends. It is especially advantageous to employ them at the receiver-end since they do not increase the transmission bandwidth requirements. Performance evaluations show that our methods outperform the other existing technique by providing better depth approximation and thus a better 3D visual effect.

Secondly, we studied one of the problems caused by unsynchronized zooming in stereo-camera video acquisition. We developed an effective algorithm for correcting unsynchronized zoom in 3D videos. The proposed scheme finds corresponding pairs of pixels between the left and right views and the relationship between them. This relationship is used to estimate the amount of scaling and translation needed to align the views. Experimental results show our method produces videos with negligible scale difference and vertical parallax.

Lastly, the transmission of 3D-content problem is addressed and two schemes for multiview video coding (MVC) are proposed. While both methods outperform the current MVC standard, one of them introduces significantly less random access delay compared to the MVC standard.

TABLE OF CONTENTS

Abstract ii
Table of Contents iv
List of Tables vii
List of Figures viii
Glossary x
Acknowledgements xii
Dedication xv
Co-Authorship Statement xvi
Chapter 1: Introduction and Overview 1
1.1 Introduction 1
1.2 The Human 3D Visual System 4
1.3 3D Content Generation 6
1.3.1 Stereoscopic dual-camera approach 7
1.3.2 3D depth-range camera approach 8
1.3.3 2D to 3D video conversion approach 10
1.3.4 Multiview video camera approach 11
1.4 3D Video Coding 12
1.4.1 Depth-based coding 13
1.4.2 Multiview video coding 13
1.4.3 Multiview video plus depth coding 16
1.5 3D Displays 17
1.5.1 Binocular-with passive glasses 18
1.5.2 Binocular-with active glasses 21
1.5.3 Autostereoscopic displays 22
1.6 Specific Challenges in 3D Technology 24
1.7 Thesis Objectives 27
1.8 Thesis Contributions 27
1.9 Thesis Summary 30
1.10 References 33
Chapter 2: An H.264-based Scheme for 2D to 3D Video Conversion 36
2.1 Introduction 36
2.2 Proposed 2D-TO-3D Conversion Scheme 40
2.2.1 Motion vector estimation 42
2.2.2 Camera motion correction 43

2.2.3 Correction of displacement estimates 44
2.2.4 Displacement correction of object borders 45
2.2.5 Perceptual depth enhancement 46
2.3 Performance Evaluation 47
2.4 Conclusion 52
2.5 References 54

Chapter 3: Generating the Depth Map from the Motion Information of H.264-Encoded 2D Video Sequence 56

3.1 Introduction 56
3.2 Background 61
3.3 Proposed Scheme 64
3.3.1 Motion vector estimation 64
3.3.2 Camera motion correction 67
3.3.3 Correction of false displacement estimates 68
3.3.4 Displacement correction of object-border pixels 71
3.3.5 Displacement correction of object-body pixels 73
3.3.6 Perceptual depth enhancement 75
3.4 Performance Evaluation and Discussion 76
3.5 Conclusion 84
3.6 References 85

Chapter 4: Unsynchronized Zoom Correction in 3D Video 87

4.1 Introduction 87
4.2 Impact of Zoom Mismatch on Subjective 3D Quality 91
4.3 Proposed Zoom Correction Method 94
4.4 Experimental Results 98
4.4.1 Objective results on digitally zoomed videos 98
4.4.2 Results on 3D video with unsynchronized optical zoom 100
4.5 Conclusions 100
4.6 References 102

Chapter 5: Efficient Inter-view Prediction for Multiview Coding 103

5.1 Introduction 103
5.2 Overview of H.264/MVC 106
5.3 Proposed MVC Methods 108
5.3.1 Adaptive MVC method 108
5.3.2 Panorama-based MVC method 116
5.4 Experiments and Discussion 121
5.5 Conclusion 124
5.6 References 126

Chapter 6: Conclusions 127

6.1 Significance of the Research 127
6.2 Potential Applications of the Research Findings 129
6.3 Contributions 130
6.4 Suggestions for Future Research 132
6.4.1 3D content recording 133

6.4.2 3D video quality metrics.....133
6.4.3 2D to 3D video conversion using multiple monocular cues.....134
6.5 References 135
Appendix A: List of Recent Publications.....136

LIST OF TABLES

Table 2.1 Subjective test scores for test streams. 49
Table 2.2 Average PSNR comparison case b and d in Figure 2.4. 51
Table 2.3 Average PSNR comparison case a, c and e in Figure 2.4. 52
Table 3.1 Average PSNR comparison case b and d in Figure 3.11. 82
Table 3.2 Average PSNR comparison case a, c and e in Figure 3.11. 83
Table 4.1 Accuracy of estimated correction parameters. 99
Table 5.1 Exponential Golomb codes. 110
Table 5.2 Test sequences. 121

LIST OF FIGURES

Figure 1.1 Future 3D TV broadcast chain. 3
Figure 1.2 3D visual depth perception (http://www.strabismus.org). 4
Figure 1.3 Stereoscopic camera setup. 8
Figure 1.4 3D Depth-range camera. 9
Figure 1.5 Multiview video camera configuration (circular arrangement). 12
Figure 1.6 Prediction structure recommended by H.264/MVC. 15
Figure 1.7 Anaglyph glasses and anaglyph image. 18
Figure 1.8 Linear and circular polarizations (http://www.zalman.com). 20
Figure 1.9 Dolby 3D. 21
Figure 1.10 Parallax barrier display. 23
Figure 1.11 Lenticular lens display. 24
Figure 2.1 Stereo geometry for two identical parallel cameras. 40
Figure 2.2 2D video sequence (a and b), recorder depth map (c and d) estimated depth map by [7] (e and f), and estimated depth map by our approach (g and h). 48
Figure 2.3 Rendered right image based on real depth map (a), estimated depth map by our approach (b), estimated depth map by [7] (c). 50
Figure 2.4 Quantitative analysis of the results. 51
Figure 3.1 Stereo geometry for two identical parallel cameras. 62
Figure 3.2 Relationship between disparity and depth for sample parallel cameras (tc=0.1 m and f=0.05 m). 63
Figure 3.3 2D video frame (a) and magnitude of estimated motion vectors (b). 65
Figure 3.4 (a) Residue frame, and (b) a color-texture segmented frame of "Orbi" sequence. 70
Figure 3.5 Initial depth map after correcting the displacement of object-border pixels. 73
Figure 3.6 Motion information after correcting the displacement of object-body pixels. 74
Figure 3.7 Estimated depth map after perceptual depth enhancement. 76

Figure 3.8 2D video sequence (a, b, c, d), recorder depth map (e, f), depth map estimated by stereo matching (g, h) estimated depth map by [12] (I, j, k, l), and estimated depth map by our approach (m, n, o, p). 78
Figure 3.9 Average subjective test scores of 3D visual perception (a) and picture quality (b) for test streams. The error bars denote the 95% confidence intervals. 80
Figure 3.10 Rendered right image based on real depth map (a), estimated depth map by our approach (b), estimated depth map by [12] (c). 81
Figure 3.11 Quantitative analysis of the results. 82
Figure 4.1 Stereo geometry with parallel cameras 91
Figure 4.2 Stereo test images. 92
Figure 4.3 Subjective results: Rating is expressed as a difference between ratings for the unsynchronized and synchronized zoomed stereo videos. The error bars denote the 95% confidence intervals. 93
Figure 4.4 Digital zooming pattern applied to the left and right views for the objective tests. 98
Figure 4.5 Sample frames of the test video with unsynchronized optical zoom. (a) Captured left-right stereo pair (b) Corrected with proposed method 100
Figure 5.1 Prediction structure recommended by H.264/MVC. 107
Figure 5.2 Prediction structure of proposed adaptive MVC method. 109
Figure 5.3 Experimental results for coding frame “d” within “Ballroom” sequence with reference frames arranged as “S2, b, c, a” versus “b, S2, c, a”. 111
Figure 5.4 Overlapping areas among frames captured by cameras, Area 1: C_1 and C_2, Area 2: C_1, C_2 and C_3, Area 3: C_2 and C_3. 115
Figure 5.5 Our proposed panorama-based MVC prediction structure. 118
Figure 5.6 Multiview images and the created panorama-view. 119
Figure 5.7 Panorama-view and Residue-view creation. 120
Figure 5.8 Flowchart of the proposed panorama-based MVC algorithm. 121
Figure 5.9 Coding results for “Rena” and “Ballroom” test sequences. 123

GLOSSARY

3D TV	Three Dimensional Television
CABAC	Context-Adaptive Binary Arithmetic Coding
CRC	Communications Research Centre Canada
DERS	Depth Estimation Reference Software
DIBR	Depth Image-Based Rendering
DPB	Decoded Picture Buffer
FTV	Free-viewpoint Television
GDV	Global Disparity Vector
GOP	Group Of Pictures
JMVM	Joint Multiview Video Model
KLT	Kanade-Lucas-Tomasi
LANC	Local Application Control Bus System
MPEG	Moving Pictures Experts Group
MSR	Microsoft Research
MTD	Time Difference Method
MV	Motion Vector

MVC	Multiview Video Coding
NCC	Normalized Cross Correlation
PSNR	Peak Signal-to-Noise Ratio
SAD	Sum of Absolute Differences
SEI	Supplemental Enhancement Information
SIFT	Scale Invariant Feature Transform
SMPTE	Society of Motion Picture and Television Engineers
SSD	Sum of Square Differences
VCEG	Video Coding Experts Group

ACKNOWLEDGEMENTS

My PhD was quite a journey, a precious experience for the rest of my personal and professional life. This achievement would not have been possible without the help, support and inspiration of many people. In the first place, I would like to express my sincere appreciation and gratitude to my supervisors. I had the honour to work with two of the most well-recognized researchers in the field of multimedia and image processing: Prof. Rabab Ward and Prof. Panos Nasiopoulos. Dr. Ward is a role model, talented and well-respected researcher. Her comments and suggestions on my work have always made me think outside the box and improve my research and writing skills. Her patience and support gave me the strength to continue. Dr. Nasiopoulos is a well-known and successful leader in the academic and business worlds. My PhD journey would have not taken me through the fascinating world of 3D multimedia, without Dr. Nasiopoulos visionary advice. His constant faith in me and his supervision made this journey easy for me. He generously granted his time and effort all the time to help me overcome my problems. I owe him my thanks and I am indebt to him for the rest of my life. Our philosophical and fun conversations will always be memorable.

I also owe my thanks to my colleagues in the Multimedia & Signal Processing Lab. I was very lucky to be surrounded by wonderful and bright people. Thank you Mehrnoush Khojasteh, Di Xu, Ashfiqua Tahseen Connie, Sergio Infante, Colin Doutre, Zicong (Jack) Mai, Victor Sánchez, Ilker Hacihaliloglu, Rupin Dalvi, Tanaya Guha, Mani Malekesmaeili, Ehsan Nezhadarya, Farhad Faradji, Mohsen (Ali) Amiri, Dr. Kaan

Ershahin, Dr. Ali Bashashati, Dr. Lino Coria, Dr. Hassan Mansour, Dr. Mehrdad Fatourechi, and Dr. Yaser P. Fallah. My special thanks to Qiang Tang for his help and friendship. Many thanks also to Matthias von dem Knesebeck for his friendship and for always being available to discuss my research and rescue me from programming and computer related issues. I also would like to thank Navrup Johal, for helping me with the market research on 3D technology.

I would like to thank the UBC ECE and UBC ICICS staff for their great help and administrative support over the years.

I am also grateful for the financial support from NSERC (Natural Sciences and Engineering Research Council) of Canada. I convey special acknowledgement to British Columbia Innovation Council (BCIC) for financially supporting my project and encouraging me to extend my knowledge in business. After taking the business course to meet BCIC scholarship requirements, I got encouraged to enrol in the UBC Engineering Management subspecialisation program and take more business courses.

I thank Prof. Zahra Moussavi (University of Manitoba) for her friendship and her kind reference letters.

My special thanks to my friends who kept me company during this journey. Our Friday night gatherings were a relief from any kind of stress and work problems. My special thanks to Tina Shoa and Arghavan Emami for their long-term, unconditional and sincere friendship. Their supportive and caring friendship made it much easier for me to live away from my family and country.

I owe special gratitude to my parents and my brother for their unconditional love and continuous support of all my undertakings in life. During the last 8 years that I have been away from them, their thoughtful calls and their inspiring and caring words did not let me feel lonely. They were upset with my failures, they worried for my problems, and they cheered for my achievements as if they are living with me. Finally, my thanks to my uncle for his irreplaceable friendship, his weekend calls and his advice during the course of my PhD.

DEDICATION

To my parents and my brother.

CO-AUTHORSHIP STATEMENT

This thesis presents research conducted by Mahsa Talebpourazad, in collaboration with Dr. Panos Nasiopoulos, Dr. Rabab K. Ward, Colin Doutre, and Dr. Alexis Tourapis.

Manuscript 1: An H.264-based Scheme for 2D to 3D Video Conversion: The identification and design of the research program, the research and data analysis were performed by Mahsa Talebpourazad. The manuscript is the work of Mahsa Talebpourazad, who received suggestions and feedback from her supervisors, Dr. Ward and Dr. Nasiopoulos.

Manuscript 2: Generating the Depth Map from the Motion Information of H.264-Encoded 2D Video Sequence: The identification and design of the research program, the research and data analysis were performed by Mahsa Talebpourazad. The manuscript is the work of Mahsa Talebpourazad, who received suggestions and feedback from her supervisors, Dr. Ward and Dr. Nasiopoulos.

Manuscript 3: Unsynchronized Zoom correction in 3D Video: Mahsa Talebpourazad worked closely with Colin Doutre for developing the original algorithm. Mahsa Talebpourazad is the primary author of this manuscript, performed the subjective tests, performance evaluations, and is responsible for the interpretation of the results. Dr. Nasiopoulos, Dr. Alexis Tourapis and Dr. Ward provided guidance and editorial input into the creation of this manuscript.

Manuscript 4: Efficient inter-view prediction structures for multiview coding: The primary investigator and author of the manuscript was Mahsa Talebpourazad, who conducted the research with guidance from Dr. Rabab Ward and Dr. Panos Nasiopoulos.

The first and last chapters of the thesis were written by Mahsa Talebpourazad, with editing assistance and consultation from Dr. Rabab Ward and Dr. Panos Nasiopoulos.

CHAPTER 1: INTRODUCTION AND OVERVIEW

1.1 Introduction

The history of three-dimensional television (3D TV) can be traced back to 1920s, when the first experimental 3D TV set-up was built [1]. Since then, several attempts have been made to introduce this technology into the market. Despite the immense keenness towards 3D, the great expectations of viewers, content providers and distributors were not fulfilled. The main drawbacks were the discomfort of the viewers (headaches, eyestrain) due to the poor quality content, the low-tech display systems and the high costs involved in the production and distribution of 3D content.

Recently, 3D TV has received increased attention among researchers and technology developers. The showcase of advanced immersive 3D displays by major TV manufacturers in consumer electronics trade shows and the production of compelling 3D movies by Hollywood are evidence that the dream of watching 3D TV at home is not far from reality.

Hollywood is at the forefront, leading the 3D technology revolution with a vested interest in seeing 3D succeed. Studios and content providers are aiming at an unprecedented 3D quality, far distant from the traditional fuzziness that has been indicative of past 3D experience. To this end, Hollywood has pioneered many new 3D initiatives, including the gathering of various stakeholders, defining standards and funding research. 3D films have greater returns at the box office due to their higher admission rates, since consumers have proved to be willing to pay more for the enhanced

3D experience. In fact, most films do not break even at the theatres and more than 60% of Hollywood's revenue comes from home entertainment, i.e., DVD sales and rentals [2]. The studios have not only been vocal about their desire for the development of 3D for home, but they have also backed this assertion with partnerships, investments and development. These investments and commitments to 3D technology at the theatre level ensure that there will be 3D content readily available for home consumers.

The sports industry also has a keen interest in the development of 3D technologies. 3D is especially applicable to sports broadcasts since it adds an immersive experience, allowing viewers to feel like they are in the actual stadium. A selling point for tickets to sporting events is the atmosphere, action, and pace of the game, which cannot be recreated at home on television. 3D technology is much better equipped to provide a new perspective of sports in action and bring a stadium-like experience to the household. 3D broadcasts are also attractive for sports leagues since they provide an alternate revenue stream, which can be priced at a premium.

Market studies predict that 30 million 3D television sets will reach US households by 2012; this number translates to 9% of the entire US TV market [3]. Canadian Communications Research Centre (CRC) speculates that 3D television will be the next major innovation in the television market [4].

3D TV can enormously enhance the viewer's experience by allowing the on-screen images to emerge and penetrate into the spectator's space. The ultimate version of 3D TV, known as Free viewpoint TV (FTV), provides TV viewers with interactive features that allow the viewer to adjust the 3D depth perception based on his/her preferences and also choose a viewing angle within a visual scene (free navigation).

Over the years, a consensus has been reached that a successful introduction of 3D TV broadcast services can only be a lasting success if the perceived image quality and the viewing comfort are better than those of conventional 2D television. This is becoming increasingly feasible because of the recent advances in capturing, coding, and display technologies, the three key components of a future 3D TV broadcast chain (see Figure 1.1).

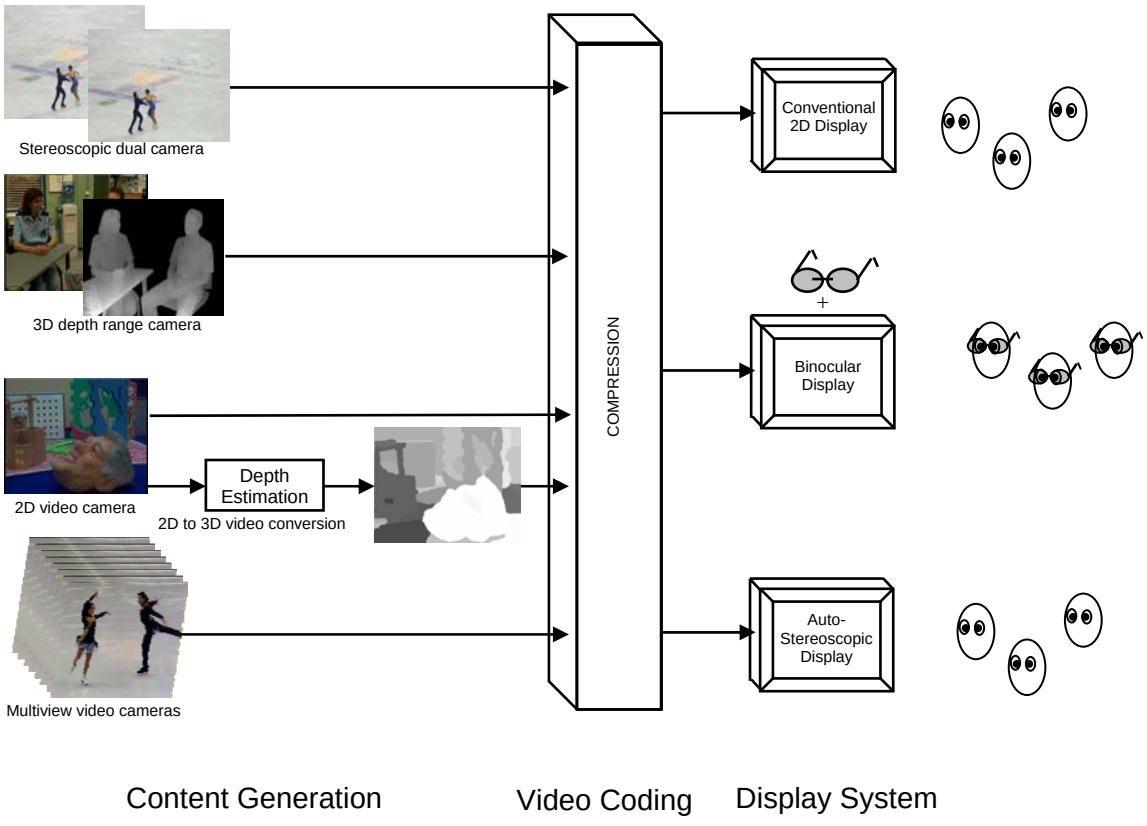

Figure 1.1 Future 3D TV broadcast chain.

In the remainder of this chapter, we provide a brief background on the 3D human visual system, followed by a comprehensive overview of the three components of the 3D TV broadcast chain (3D content generation, 3D video coding and 3D display systems) and the existing challenges. Subsequently, the thesis statement and our research objectives are discussed. At the end, we summarize our research contributions.

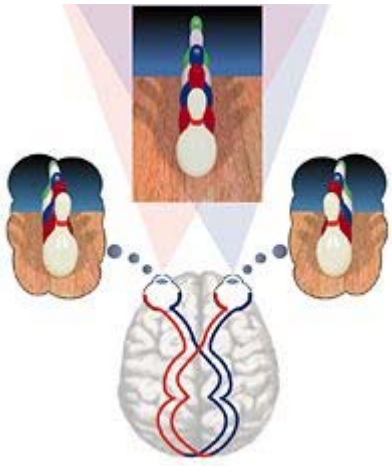

Figure 1.2 3D visual depth perception (http://www.strabismus.org).

1.2 The Human 3D Visual System

Human depth perception is based on a combination of many visual cues as well as internal mental templates and expectations. For most people, the 3D depth perception is realized by two slightly different images being projected on the left and right eye retinas (*binocular parallax*), each represents a slightly different viewpoint. The brain fuses the two images to give the depth perception. The viewer then sees one solid scene instead of two slightly different projections (see Figure 1.2). The perceived image with depth contains all the information present in the two individual viewpoint images. It also conveys something else that is not present in either of them: an intrinsic feeling of depth, distance and solidity. The differences between the left and right eye viewpoint images arise because an object in a scene will not fall in the same spot in both images. This relative displacement is according to the object's distance from the viewer (this displacement is referred as *disparity*).

Although the binocular parallax is the most dominant cue for depth perception, there exist many other depth cues known as monocular depth cues, which do not require the observer to have two eyes to perceive depth. Over the years, the human brain has been trained to perceive depth using these cues. Below, we list a number of such monocular depth cues [5, 6]:

Relative size: If two objects in a scene are known to have the same size (observer's priori knowledge) but they are located at different distances from the observer, then the projection of the near object onto the observer's retina will be larger than the projection of the far object. In addition, an object with a larger size may appear smaller than a smaller object, if it is located at a much further distance from the observer.

Motion parallax: The relative motion between the viewing camera and the observed scene provides an important cue to depth perception: near objects move faster across the retina than distant objects do. This motion may be seen as a form of "disparity over time", represented by the concept of motion field.

Occlusion (Interposition): The principle of depth-from-occlusion has its roots in the phenomenon that an object which overlaps or partly obscures our view of another object is considered to be closer. Occlusion is also known as interposition and offers rich information in relative depth ordering of the objects.

Light and shade: They provide a clue of the relative position of objects in all imagery recorded from the real world, mainly photographs and video material. Shadows also form occlusion (occlusion is also monocular depth cue).

Texture gradient: Texture gradient is another depth cue since the face-texture of a textured material (such as fabric or wood) is more apparent when it is closer.

Haze (Atmosphere scattering): Haze happens when the direction and energy of light propagation through the atmosphere is altered due to diffusion caused by small particles in the atmosphere. As a result, the distant objects visually appear less distinct and more bluish (the atmosphere scatters red light) than objects nearby.

Perspective: An objects appear to be getting gradually smaller as it gets further away from the observer. Also, parallel lines, such as railroad tracks, appear to converge with distance, eventually reaching a vanishing point at the horizon.

1.3 3D Content Generation

Currently there does not exist an industry-wide accepted mastering standard regarding the format of 3D content. Consumer electronics companies are thus delivering various 3D technologies with incompatible formats. This industry fragmentation and lack of standardization has hold back the development of 3D technologies. Standardization is one of the key components needed for the successful development and employment of 3D. To this end, Hollywood has spearheaded efforts to implement standards through the Society of Motion Picture and Television Engineers (SMPTE), the leading technological group in Hollywood. SMPTE has created a task force to discuss possible standards for 3D content. The goal is to make 3D system backward compatible with present 2D system, scalable and deliverable over cable, satellite, disk, and the Internet. SMPTE delivered their report in April 2009. The report defines guidelines for the mastering standard, which will be used by 3D developers. Using these guidelines, SMPTE hopes to

define the core standards by this summer (2010) [7]. The same society also hopes to work with other standards development organizations to develop the standards for complementary products, to ensure compatibility from end to end [7]. It is expected that these interoperable standards are implemented across the industry within two years [7].

In general, there are four types of 3D content generation as shown in Figure 1.1: i) the stereoscopic dual-camera approach, which results in two separate views (left and right), ii) the 3D depth-range camera approach, which generates a 2D image plus a depth map, iii) the 2D-to-3D video conversion approach, which converts existing 2D video material into stereoscopic 3D by estimating a depth map from the 2D video sequence and subsequently rendering the left and right sequences, and iv) the multiview video camera approach. The following subsections present an overview of the different schemes for 3D content generation.

1.3.1 Stereoscopic dual-camera approach

In stereoscopic videos, the function of the retinas in the visual system is mimicked by the lenses of two identical synchronized cameras, which record the left-eye and the right-eye views from two slightly different perspectives (see Figure 1.3). Then, when the viewer watches stereo videos, the recorded right and left view images are projected on the viewer's eyes and the brain reconstructs the third dimension by combining the received visual information. The configuration of the cameras can be parallel (with axial offset of the imaging sensor) or toed-in (where the cameras are angled in). However, to eliminate keystone distortion and depth plane curvature, the parallel camera configuration is preferred (see [8] for more details).

Figure 1.3 Stereoscopic camera setup.

The production of stereoscopic dual-camera video is highly demanding. Two cameras should be configured so that the contrast, brightness, colour, and sharpness of captured images are the same or within a very tight tolerance to prevent eyestrain and headache for the viewer [9]. In addition, the cameras need to be properly calibrated so that the disparity introduced to the viewer is similar to the one he/she receives from the actual scene. This consensus should be satisfied even when visual effects such as zoom-in or zoom-out occur. This is very challenging in the case of 3D. For example, an increased zoom-in may break the 3D effect in the sense that viewers become unable to fuse the right and left view images.

1.3.2 3D depth-range camera approach

An approach to cope with some of the limitations of stereoscopic dual-cameras, is to exploit 3D depth-range cameras. These cameras capture 3D content as two video sequences: a conventional 2D-RGB view and its synchronized depth map [9]. As shown in Figure 1.4, for each image point in the depth map image, the depth information is stored as an 8-bit number (between 0 and 255). The closer the object is, the greater this number is. This format allows easy capturing, simplifies the postproduction, and requires a lower transmission bandwidth compared to the dual-camera configuration.

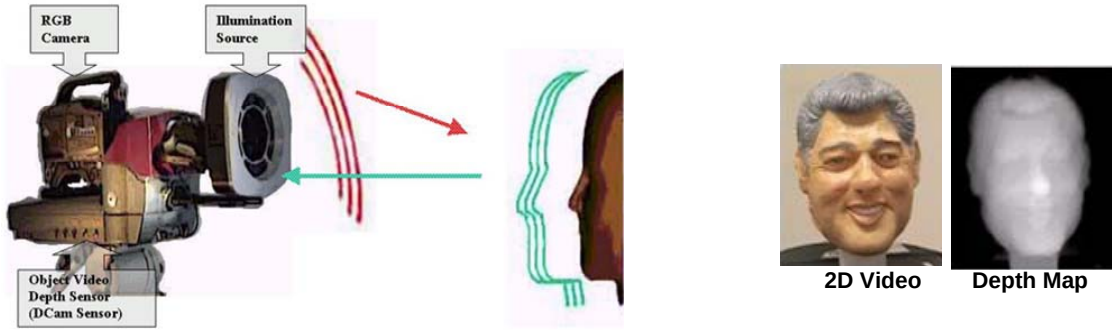

Figure 1.4 3D Depth-range camera.

The disadvantage of this approach is that in order to watch the 3D content captured by a 3D depth-range camera, the left- and right-eye views must be reconstructed at the receiver end. This is achieved using depth image-based rendering (DIBR) techniques [10, 11]. Rendering techniques create two images, one for each eye, in such a way that 1) when independently viewed they present an acceptable image to the visual cortex and 2) when simultaneously watched the viewer can fuse both images and perceive the depth information of the scene as if he/she is viewing a real scene. One potential problem here is estimating the image information of areas that are present in one of the stereo images and occluded in the other [10, 11]. The other issue is related to the conflict of depth cues in a sense that during rendering one cue may become dominant and it may not be the correct/intended one. As a result, the depth perception will be exaggerated or reduced. This means that watching the rendered images will be uncomfortable and in some cases the stereo pairs may not fuse at all (i.e., the viewer would see two separate images).

Examples of 3D depth-range cameras are the *AXI-Vision* developed by *NHK*, and the *Zcam* manufactured by *3DV Systems* [12, 13].

1.3.3 2D to 3D video conversion approach

It is widely accepted that the success of the 3D technology and its market penetration will directly depend on the availability of 3D content. It is probably not realistic (in the introduction phase of 3D TV) to assume that the need for 3D content can be satisfied only with new-recorded materials. One alternative solution is the conversion of existing 2D popular movies and documentaries into 3D format to be watched on 3D screens. Successful implementation of such an approach will also create a new market opportunity for content owners and providers to resell their existing products. It is because of these reasons that 2D to 3D conversion has recently received a lot of attention by the research and industry communities.

Converting 2D content to 3D video streams is possible if the depth information is estimated from the original 2D video sequence. Having the depth information along with the 2D video, 3D video content can be created using the DIBR techniques as discussed in subsection 1.3.2.

Conversion of existing 2D video materials to 3D is a very challenging task. Depth map estimation techniques try to use monocular depth cues and imitate the human visual system when estimating the distance between objects. The difficulty of this task is the absence of the binocular parallax information, which is the most dominant cue for depth description. Depth map estimation techniques generally fall into one of the following categories: manual, semi automatic and automatic. For the manual methods, an operator would manually draw the outlines of objects that are associated with an artistically chosen depth value. As expected, these methods are extremely time consuming and

expensive. For this reason, semi-automatic [14, 15] and automatic techniques [16-24] are preferred for depth map estimation.

1.3.4 Multiview video camera approach

The multiview video camera approach involves capturing the scene from multiple viewpoints with a setup of N synchronized cameras (see Figure 1.5). The configuration concerns of this approach are similar to those of the stereoscopic dual-camera approach, with the exception that there are N synchronized cameras rather than two. In this case, several people can watch 3D videos from slightly different viewing angles. Ultimately, we would like to offer the viewer the opportunity to choose his/her preferred viewing angle (free viewpoint TV). To achieve this, we need to have a high camera density (large N) and the ability to accurately interpolate any possible view in-between using the camera parameters [25, 26].

Figure 1.5 shows a set-up of multiview video cameras in a circular arrangement (there are different arrangements depending on the application) and virtual cameras (representing synthesized intermediate views). In general, the quality of the intermediate views increases as the number of available cameras increases. This is because more original image information becomes available as the number of cameras increases. On the other hand, the use of more cameras increases the capturing and processing expenses but improve the quality of the interpolated views (an obvious trade-off between cost and quality).

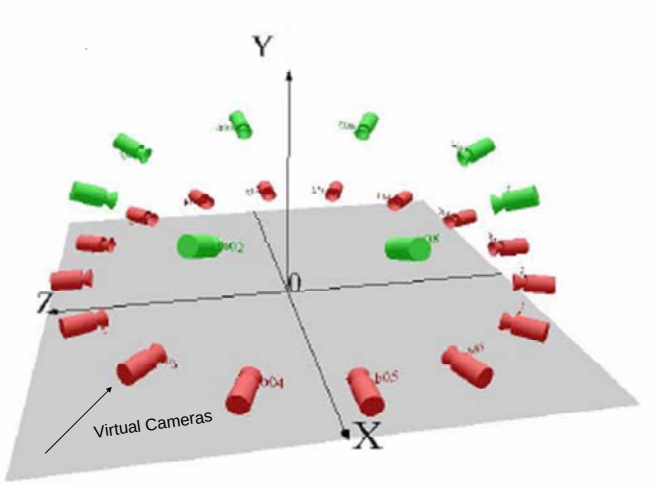

Figure 1.5 Multiview video camera configuration (circular arrangement).

1.4 3D Video Coding

As discussed in subsection 1.3, there are different types of 3D video generation represented by different types of raw data. Coding and compression of these data form the next block in the 3D video processing chain, and that is the scope of this subsection. Many different compression techniques have been proposed over the years. In the following subsections we summarize these techniques under three categories: 1) depth-based coding, 2) multiview video coding and 3) multiview video plus depth coding. In the first category, the depth-based coding targets 3D content in the form of 2D video plus depth recorded by depth-range cameras or generated by 2D to 3D video conversion techniques. In the second category, the multiview video coding targets stereoscopic 3D (two views) and multiview video content. In the third category, multiview video plus depth coding focuses on compression of multiview videos and the corresponding depth maps for FTV applications (depth information is transmitted for synthesizing intermediate views).

1.4.1 Depth-based coding

Researchers have put much effort in recent years in developing efficient compression techniques for 2D video sequences. The same techniques can be applied to compress 3D content. Experiments on compression of 3D content in the form of a 2D video stream and a depth map sequence show the following: if MPEG-2 is used to compress both streams, the transmission of the depth map stream increases the required bandwidth of 2D video stream by 20% (at a typical broadcast bitrate of 3 Mbit/s for 2D video) [27]. If the H.264/AVC standard is used instead, the required bandwidth increases by only 8% [27, 28]. Higher compression ratios can be however achieved by better exploiting the features of the depth data. For example, Grewatsch et al. took advantage of the existing correlation between the 2D video sequence and the depth map sequence and used the motion vectors (MVs) obtained for the 2D video sequence (based on MPEG-2 standard) to encode the depth map sequence [29]. Another technique [30] improved on [29] by implementing the same concept using the H.264 standard and selectively choosing 2D MVs for coding the depth map sequence.

1.4.2 Multiview video coding

One major challenge with multiview video applications is the transmission of huge amount of data, which requires the development of highly efficient coding schemes. Another challenge is that any compression scheme designed specifically for multiview video streams should support random access functionality, i.e., allowing viewers to access arbitrary views with minimum time-delay (one of the promises of future FTV).

A straightforward approach for coding multiview video content is simulcast coding. This compresses each video stream independently. While this scheme exploits

temporal and spatial correlations within each stream, it does not benefit from the existing correlation between the different views. Multiview sequences show a scene from many different viewing angles, which means that there is a high possibility of inter-view correlation between the multiple streams. The existence of this multiple correlation makes multiview video coding have a different structure from single-view coding techniques. To address this issue, in December 2001, the ISO/IEC Moving Pictures Experts Group (MPEG) established an ad hoc group (AHG) on 3D Audio Visual (3DAV) with a mandate to evaluate and standardize a new technology that extends the capabilities of existing MPEG standards in terms of creating visual 3D impression with interactive features [31]. The investigations by 3DAV AHG showed that H.264 is the most efficient one in terms of compressing 3D content among all existing MPEG standards. Following that, MPEG issued a "Call for Proposals on Multiview Video Coding (MVC)" which were evaluated in January 2005. This led to the development of a new, dedicated standard for MVC under ISO/IEC MPEG and the ITU-T Video Coding Experts Group (VCEG). The MVC standard was added as an extension to H.264/AVC (MPEG-4 Part10, Amendment 4) in July 2008 [32]. H.264/MVC enables efficient encoding of sequences captured simultaneously from multiple cameras for 3D TV applications. This standard uses hierarchical B pictures for each view and at the same time, applies inter-view prediction to every 2nd view, using already encoded frames from adjacent camera views. Figure 1.6 shows the prediction structure supported by the MVC standard which utilizes a hierarchical-B picture structure, involving an 8-view sequences and GOP-length of 8. The horizontal and vertical directions represent the temporal and spatial axes, respectively. As illustrated in Figure 1.6, H.264/MVC tries to predict the video frame

from a given camera using one or more video frames of neighboring-cameras (disparity estimation) in addition to the consecutive frames of the given camera stream (motion estimation). Performance evaluations show that this approach outperforms simulcast coding (coding each stream separately) by an average gain of 1.5 dB PSNR (or an average 20% compression ratio enhancement). The performance in this case strongly depends on the arrangement of the cameras (i.e., positioning and distance between cameras) [33].

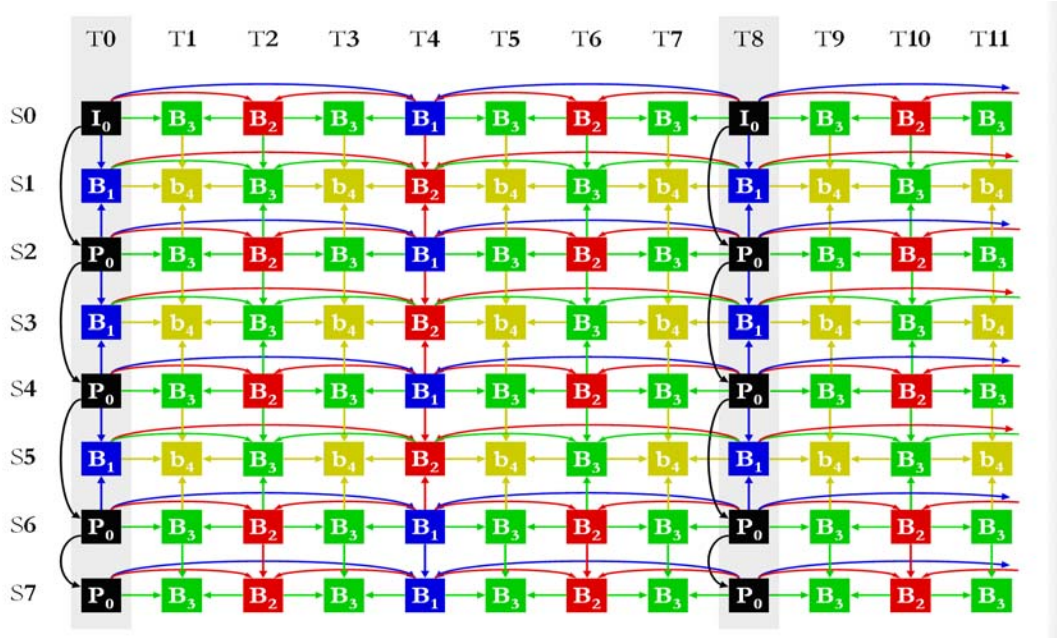

Figure 1.6 Prediction structure recommended by H.264/MVC.

Although this inter-view prediction approach enhances the compression performance of MVC, it also introduces computational complexity and random-access delay. Random access delay is measured based on the maximum number of frames that must be decoded in order to access a B-frame in the hierarchical structure. The access delay for the highest hierarchical order is given by:

$$F_{max} = 3 * level_{max} + 2 * \lfloor (N-1)/2 \rfloor \qquad (1\text{-}1)$$

where $level_{max}$ is the highest hierarchical order and N is the total number of views. For instance, in order to access a B-frame in the 4th hierarchical order (B4-frames in Figure 1.6), 18 frames ($F_{max} = 18$) must be decoded.

MVC is backward compatible with H.264/AVC, i.e., it allows existing 2D play-back devices to decode only one of the two views of a stereoscopic video stream (ignoring the other one).

1.4.3 Multiview video plus depth coding

Current MPEG activities aim at establishing a standard for free viewpoint video (where a 3D viewpoint can be changed freely) by estimating the depth of the scene and synthesizing intermediate views. To this end, the multi-view video plus depth map approach is included in recent proposals and ongoing tests by the MPEG community. In this format, a depth image is estimated for each associated view of the multi-view videos.

In compressing depth map sequences, maintaining the fidelity of the depth information is important, since the quality of the synthesised view is highly dependent on the accuracy of the geometric information provided by depth. Therefore, it is crucial to achieve a good balance between the fidelity of depth data and the associated bandwidth requirements. As reported in [34], the rate used to code the depth map stream with pixel-level accuracy could be quite high and of the same order as that of texture video. Experiments were performed to demonstrate how the synthesised video quality varies as a function of the bit rate for both texture and depth videos. It was found that higher bit

rates were needed to code the depth data so that the view rendering quality around object boundaries is maintained.

Various approaches have been considered in the literature to reduce the required bit rate for coding depth, while maintaining high view synthesis and multiview rendering quality. One approach is to code a reduced resolution version of the depth map using conventional compression techniques and then apply filtering to reconstruct high resolution depth map images [35,36]. This method could provide substantial rate reductions, but the filtering and reconstruction techniques need to be carefully designed to maximize quality. Another method is to code the depth based on the geometric representation of the data [37, 38]. A drawback of this scheme is that it appears difficult to extend it to video applications. An alternative multi-layered coding scheme for depth is suggested in [39]. The argument here is that the quality of depth information around object boundaries needs to be maintained with higher fidelity since it has a notable impact on subjective visual quality [18]. This scheme guarantees a near-lossless bound on the depth values around the edges while allowing lossy compression for the rest of regions. This method effectively improves the visual quality of the synthesized images by maintaining the accuracy of depth information around the edges.

1.5 3D Displays

Displaying 3D content is the last component of the 3D broadcasting chain. Although this part falls outside the scope of our thesis, we feel that a short overview of this topic will help the reader obtain a better understanding and appreciation of the present status of the 3D technology. The 3D displays fall into two categories: Binocular

(with active or passive glasses), and Autostereoscopic (without glasses) displays. The following subsections elaborate on the different 3D displays.

Anaglyph Glasses

Right Image Anaglyph Image Left Image

Figure 1.7 Anaglyph glasses and anaglyph image.

1.5.1 Binocular-with passive glasses

1.5.1.1 Colour filtered-anaglyph

Anaglyph is one of the first commercial methods for displaying 3D (year 1853). In anaglyph displays, the left and right eye images are filtered with near-complementary colors (red and green for Europe, red and blue for the USA). The right and left eye images are superimposed over each other (see Figure 1.7). The viewers are required to wear color-filter glasses to filter the images and perceive depth. This well-known and inexpensive method has been used for stereoscopic cinema and television, and is still popular for viewing stereoscopic images in print (magazines, etc.), since the approach readily lends itself to the production of hard copies. A serious limitation of this method is that color information is lost since it is used as a selection mechanism. Only limited color rendition is possible through the mechanism of binocular color mixture. The other drawback of this system is crosstalk. Crosstalk in a 3D display results in eyes seeing the wrong view (left eye sees the right view image and vice versa). On the actual display,

crosstalk is seen as double contour (ghosting) and is a potential cause of eyestrain and headaches [40, 41].

1.5.1.2 Polarized

Polarization-based displays separate left and right eye images by means of polarized light. Left and right output channels (monitors or projectors) are covered by orthogonally oriented filters, using either linear or circular polarization. The polarized stereo images are projected and superimposed onto the same screen. The observer needs to wear polarized glasses to separate the different views again. When watching with glasses, since each lens passes only the light that is polarized in its polarizing direction and blocks the light polarized in the opposite direction, each eye sees its matching image and the observer perceives depth effect.

Linear polarized glasses use vertical polarization on one lens and horizontal polarization on the other (see Figure 1.8). The 3D effect is perceived as long as the user's head is kept straight. Tilting the head will break the 3D effect and some amount of ghosting or crosstalk may occur.

Circularly polarized lenses are polarized clockwise for one eye and counter-clockwise for the other (see Figure 1.8). This method of polarization will maintain the 3D effect if the head is tilted.

The polarized-based display system offers good quality stereoscopic imagery, with full color rendition at full resolution, and very little crosstalk in the stereo pairs [40]. It is the system most commonly used in stereoscopic cinemas today. The most significant

drawback of this kind of system is the loss of light output due to the use of polarizing filters (which is more evident in circular polarization).

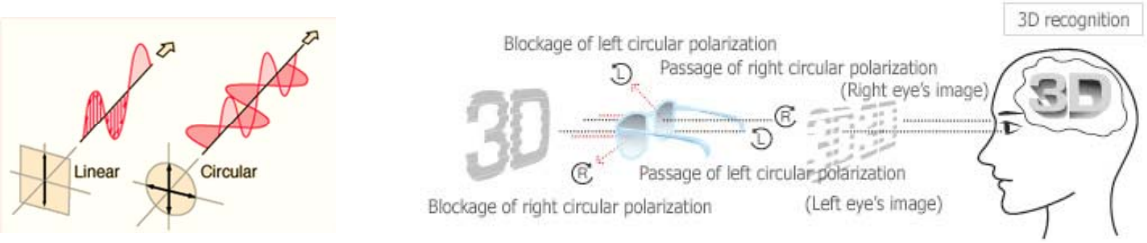

Figure 1.8 Linear and circular polarizations (http://www.zalman.com).

1.5.1.3 Spectrum filtered-Dolby 3D

Dolby 3D uses Infitec technology which stands for interference filter technology. This system encodes left and right images by projecting each with a differently filtered spectrum of light. In this case the light is filtered differently for each view, but both the left and right spectrums appear as white light or near-white light (Figure 1.9). This differentiates Infitec from the anaglyph method which uses red filters for one eye and blue filters for the other. In Dolby's implementation, the light path in the projector is modified with a filter wheel to achieve spectral division of the stereoscopic images (see Figure 1.9). Prior to projection, some color-balancing is applied to the image signal inside Dolby's digital cinema server.

Complementary spectral division glasses are worn by audience members for decoding the images so that left eye images are seen only by the left eye, and right eye images are seen by only the right eye. To accomplish this, Dolby's glasses employ some 50 layers of thin-film coatings to create the appropriate optical interference filters.

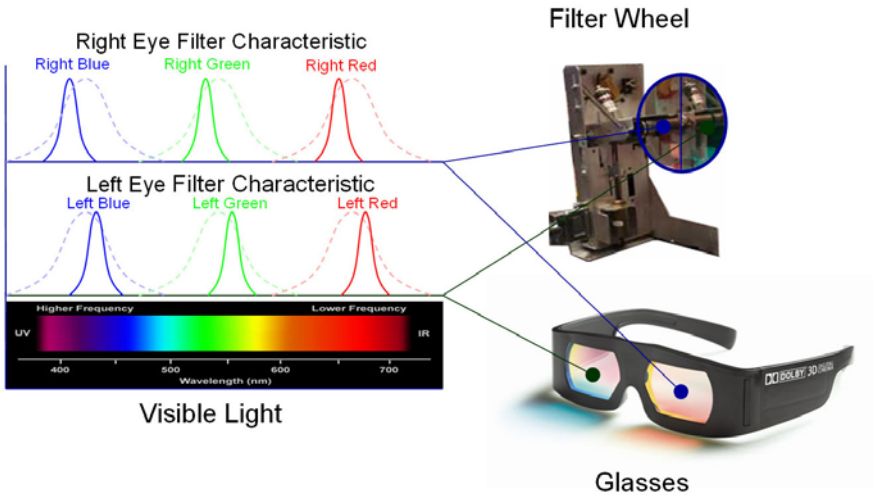

Figure 1.9 Dolby 3D.

1.5.2 Binocular-with active glasses

Shutter glasses are the most commonly used active 3D glasses. The lenses of shutter glasses are actually small LCD screens. When voltage is applied, the "shutters" close and the lens goes dark. This behaviour is synchronized with the screen displaying the 3D content, usually through an infrared transmitter. When the viewer looks at the screen through shuttering eyewear, each shutter is synchronized to occlude the unwanted image and transmit the wanted image. Thus, each eye sees only the appropriate perspective view. The left eye sees only the left view, and the right eye only the right view. On an LCD or LED television, this method of 3D displaying cuts the refresh rate in half and has been known to cause headaches for some people. To eliminate this problem, new display systems use a refresh rate double that of conventional displays (i.e. 120Hz rather than 60Hz).

1.5.3 Autostereoscopic displays

Auto-stereoscopic displays apply optical principles such as diffraction, refraction, reflection and occlusion to direct the light from the different perspective views to the appropriate eye [40], allowing multiple users to watch 3D content at the same time without wearing specialized 3D glasses. This property makes them the best candidate for future consumer 3D TVs. One of the drawbacks of this system is that the resolution for each view drops as the number of views increases. The arrival of high resolution flat panel displays has made multiview applications more feasible [42]. The other important drawback of these systems is the fact that only under a limited horizontal viewing angle the picture will be perceived correctly.

Historically, the two most dominant autostereoscopic techniques are based on parallax barriers and lenticular arrays, and these techniques are still popular today. The following subsections elaborate on parallax barriers and lenticular lenses.

1.5.3.1 Parallax barrier

Parallax barrier displays are based on the principle of occlusion, where part of the image is hidden from one eye but visible to the other eye. As it can be observed in Figure 1.10, at the right viewing distance and angle, each eye can only see the appropriate view, as the other view is occluded by the barrier effect of the vertical slits. Different implementations of this principle are available, including parallax illumination displays (where the opaque barriers are placed behind the image screen) and moving slit displays (use time-sequential instead of stationary slits). The main advantage of these displays is their backward compatibility in a sense that they can be switched to a 2D display mode. It

is imperative that 3D television technology should be compatible with conventional 2D television to ensure a gradual transition from one system to the other.

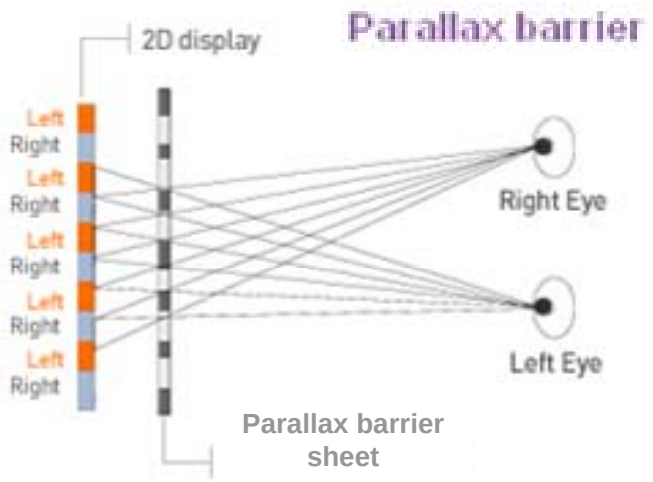

Figure 1.10 Parallax barrier display.

1.5.3.2 Lenticular lens

Lenticular systems are based on the principle of refraction. As it can be observed from Figure 1.11, instead of using a vertical grating as with parallax barrier displays, an array (or sheet) of vertically oriented cylindrical lenses is placed in front of columns of pixels, alternately representing parts of the left and right eye view. Through refraction, the light of each image point is emitted in a specific direction in the horizontal plane. In what is known as the sweet spot of a display, left and right images can be delivered to the correspondent eye to create a 3D effect. Older, less sophisticated systems, required the viewer to sit at a specific distance and angle in order to properly view the image and avoid headaches and eyestrain. Current lenticular lens systems have corrected this by using a slanted lenticular sheet, allowing up to eight viewers to observe a 3D image with no ill effects.

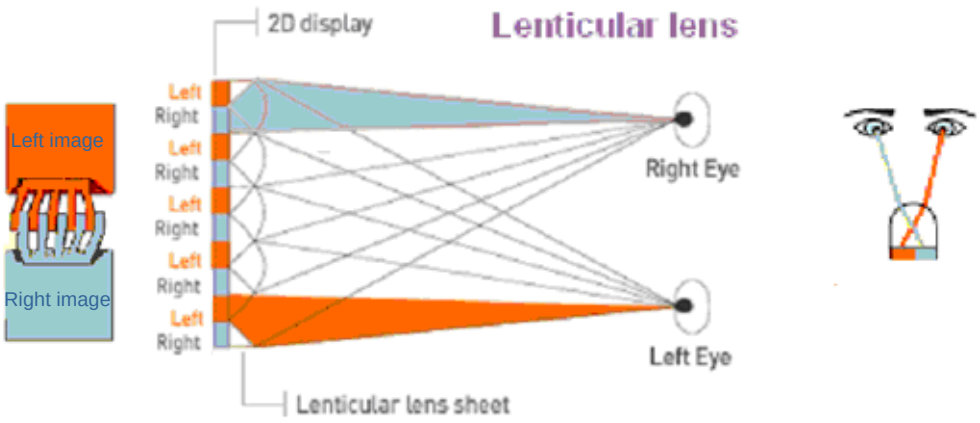

Figure 1.11 Lenticular lens display.

1.6 Specific Challenges in 3D Technology

The success of the 3D technology and the speed at which it will penetrate the entertainment market depend on how well the SMPTE and MPEG Working Groups will be able synchronize the standardization efforts of the three key components: 1) 3D content generation, 2) coding and transmission and 3) playback. Although significant work has been done in recent years regarding each of these components, the resulting findings have only managed to expose the challenges laying ahead. The rest of this subsection briefly points out some of these challenges.

In addition to the efforts for generating 3D movies and other content, the successful introduction of the 3D TV to the consumer market highly depends on the availability of 3D content. It is for this reason that an increased emphasis has been given to the conversion of existing 2D content to 3D format.

2D to 3D video conversion is a very challenging task, since it requires the approximation of depth information of the scene based on monocular depth-cues. The quality of the approximated depth-map has direct effect on the quality of the rendered 3D

content. Thus, there is an obvious need for developing effective depth estimation methods. The success of depth estimation techniques strongly relies on how well they can imitate the human visual system in incorporating monocular depth cues in order to estimate the distance between different objects in a scene. Although several depth estimation methods have been proposed, the generation of high quality 3D content from 2D video streams is still far from reality. Although one may think that there is no need for real-time conversion of existing 2D video content to 3D, there are advantages where such conversion is welcome. For instance, if 2D videos are transmitted and the 2D to 3D conversion is performed at the receiver side, then the required bandwidth will be reduced. In that case many real-time programs are broadcasted only in 2D (e.g., news and sports) can also be viewed in 3D. For this reason, there is a lot of emphasis on developing efficient real-time 2D to 3D methods for the decoder side.

The generation of new 3D content, as discussed before, is highly demanding and requires both the producer and cameraman to be experts in 3D production and also understand the limitations of existing 3D technologies. To avoid fatigue and nausea, left and right pictures must be as identical as possible in terms of vertical shift, colour homogeneity and focus. In most cases, matching the above criteria is only possible by using post-processing correction algorithms.

One of the shooting conditions that can degrade the perceived 3D quality is unsynchronized zooming of the dual cameras. Synchronizing the optical zooming of two identical cameras precisely is not an easy task. If the two cameras have different zoom factors, an object will have different sizes in the left and right views, and thus vertical parallax (vertical shift) will be introduced. This causes eyestrain and interferes with the

fusion of the two images. Correcting the unsynchronized zooming factor requires the development of an efficient post-processing algorithm.

As already discussed in subsection 1.4, one of the major challenges of a 3D broadcast system is the huge amount of information needed to be transmitted. This problem is much more severe in the case of multiview video content. Although the developed MVC standard (H.264/MVC) is more efficient in compressing multiview video streams than simulcast coding (H.264/AVC), its prediction structure introduces random access delay and computational complexity (refer to equation 1-1). This hampers the random access functionality of Free viewpoint TV (FTV) because of the introduced time-delay when arbitrary views are accessed by viewers. A straightforward approach for facilitating random access is to increase the number of I and P frames, but this significantly reduces the compression efficiency. Thus, there is a tradeoff between compression performance and random access time-delay in the prediction structure suggested by H.264/MVC. Moreover, as experimental results show, the inter-view prediction structure of the MVC standard is more efficient in compressing multiview video streams captured by a camera setting where the cameras are close to each other compared to a set-up with large distances between the cameras [33]. The gain strongly depends on the original setting of the multi-camera arrangement: as the interval between cameras increases the disparity between the views increases. In such a case, inter-view prediction using frames from adjacent views may not be successful in reducing the inter-view redundancies.

1.7 Thesis Objectives

Based on the above discussions, the employment of a 3D TV broadcast service faces many challenges in terms of 3D-content generation, transmission and display. The objective of my thesis is to address some of these challenges, specially:

1. To develop efficient real-time methods that can estimate the depth-map of any scene from compressed 2D video sequences. The goal is to design techniques, which can extract and utilize the monocular depth cues from the information in 2D video. Since the implementation of these methods is aimed to be carried at the receiver-side, it is important for these techniques to be of low computational complexity.
2. To develop an efficient post-processing algorithm that can correct the unsynchronized zooming factor of the recorded 3D content by stereo cameras.
3. To design an inter-view prediction structure for multiview video coding (MVC). MVC utilizes the correlation between adjacent streams as well as that within each stream. The designed interview-prediction structure should be effective in terms of compression efficiency as well as random access delay, two of the key factors required for future interactive multiview systems.

1.8 Thesis Contributions

The main contributions of this thesis are summarized as follows:

- We designed a new and efficient method that estimates the depth map of a 2D video sequence using the existing H.264/AVC estimated motion information. This method exploits the existing relationship between the

motion of objects and their distance (motion parallax) from the camera, to estimate the depth map of the scene. Our proposed method modifies the motion information based on the characteristics of the 3D visual perception. One advantage of the proposed approach is that it can be implemented in real-time at the receiver-end, thus the transmission bandwidth requirements are not increased. Performance evaluations show that our method outperforms the other existing H.264 motion-based depth map estimation technique by providing better approximation for the scene's depth map and thus a better 3D visual effect.

- We then utilized color-texture segmentation to identify objects within the scene and estimate their motion (H.264/AVC estimated motion vectors are block-based). This further improves the quality and smoothness of estimated depth maps. Performance evaluation shows that this approach results in a higher quality and a smoother depth map compared to our previous approach.

- We developed an effective algorithm for correcting unsynchronized zoom in 3D videos. The proposed scheme finds matching points (i.e., corresponding points) between the left and right views. The y coordinate of a matching pair in the synchronized zooming is identical. To correct the vertical parallax (vertical shift) introduced by the unsynchronized zoom, a least squares regression is performed on the y coordinates of all matching points. This will determine which view needs to be scaled and to estimate the amount of scaling and translation needed to align the views. Experimental results show

our method produces videos with negligible scale difference and vertical parallax.

- We developed a new prediction structure for coding multi-view camera sequences. In the MVC standard as shown in Figure 1.6, each block in a frame is predicted based on the blocks of the frames within the same stream (temporal and spatial prediction) or the blocks of corresponding frames from adjacent views (inter-view prediction). The frames which are used for predicting each block are called reference frames. Our proposed scheme constructs an extra reference frame, which is used to improve the inter-view prediction in the MVC standard (H.264/MVC). Later, our adaptive approach automatically re-sorts the reference frame list to prevent the use of extra bits for coding reference frame indices. Performance evaluations show that the proposed scheme is effective in compressing multiview streams due to its enhanced inter-view prediction structure.

- We developed an efficient multiview video coding scheme with a new inter-view prediction structure. Our MVC method has merits in terms of coding efficiency and random-access delay. These two capabilities will be required in future interactive multiview systems. Performance evaluations show that the proposed scheme outperforms H.264/MVC in terms of both compression efficiency and random-access delay.

1.9 Thesis Summary

This is a manuscript-based thesis that follows the specifications required by the University of British Columbia for this format. In addition to this introductory chapter, the thesis includes four chapters two of which have been already published, one has been submitted and one is ready for submission to refereed academic journals and they all have been slightly modified in order to offer a logical progression in the thesis. Please note there is some redundancy between chapters because of the manuscript-based format of thesis. The final chapter discusses the conclusions and directions for future work. In what follows, we give a detailed summary for the following four chapters, i.e., chapters 2, 3, 4 and 5, which include the main work/contributions that we have made in this thesis.

In chapter 2, an efficient method that converts 2D video sequences to 3D is presented. This method utilizes the motion information between consecutive frames to approximate the depth map of the scene. To estimate the depth map, the horizontal motion captured by a single camera is revised and then approximated as the displacement between the right and left frames captured by the two cameras in a stereoscopic set-up case. To enhance the visual depth perception, a non-linear scaling model is then applied to the modified motion vectors. The low complexity of our approach and its compatibility with future 3D systems, allows real-time implementations at the receiver-end for no additional bandwidth burden on the network. Performance evaluations show that our approach outperforms the existing H.264-based depth map estimation technique by up to 1.84 dB PSNR, providing more realistic depth representation of the scene. Moreover, the subjective comparison of the results (obtained by viewers watching the generated stereo video sequences on a 3D display system) confirms the better performance of our method.

Chapter 3, presents an algorithm that improves the proposed 2D to 3D conversion scheme of chapter 2 using color-texture segmentation to identify objects and correct motion vectors accordingly. Our objective and more so the subjective evaluations show that the implementation of this approach improves the performance of our other method presented in chapter 2 by enhancing the quality of the estimated depth maps.

In chapter 4, we first present a subjective study that shows the perceived quality of stereo video is greatly reduced when the two views were acquired using different zooming factors. Next, we present a method for correcting such zoom mismatches by cropping and scaling one of the two views, considering the direction of the zoom operation. Our method involves finding matching points, i.e., corresponding points between the left and right views, and performing least-squares regressions to estimate the amount of scaling and cropping required to make the views consistent. Experiments were performed on videos with digitally introduced zoom mismatch and videos with optical unsynchronized zoom. In both cases the results show that our method is highly accurate and produces videos without size differences or vertical parallax between the two views.

In chapter 5, we present two efficient inter-view prediction structures for multiview video coding (MVC): an adaptive MVC and a panorama-based MVC. Our adaptive MVC algorithm synthesizes extra video streams and uses them as extra references when coding the original views. These streams are synthesized based on the already encoded frames from neighboring views and without requiring the scene's depth information. The proposed scheme utilizes both motion and disparity compensation methods to exploit temporal and inter-view correlation within each view sequence and among views, respectively. To guarantee the best bitrate performance, the minimum

number of bits is used for coding the reference frame indices. To carry this effectively, our algorithm adaptively re-sorts the reference frame list.

In the proposed panorama-based scheme, inter-view prediction (disparity estimation), which introduces time-consuming computations and random access delay to MVC, is replaced with a residue-stream coding process. Our algorithm transforms the middle view to a panoramic view of the scene. This is done by expanding each frame of the middle stream by adding the image information of the corresponding frames in the other streams. Then, the residue streams are created as the difference of the luma and chroma values of the overlapping regions of each view and the panoramic view. Finally, the panoramic stream and all the residue streams are encoded separately (simulcast coding). The hierarchical B picture prediction structure is implemented for coding each stream.

Objective evaluations confirm that proposed coding methods result in better compression performance compared to the recent multiview video coding (H.264/MVC) standard. Our adaptive MVC scheme outperforms the standard MVC by up to 1 dB PSNR and enhances the compression ratio by 22.97% while our panorama-based MVC scheme enhances the compression ratio by 24.6% and the quality by up to 2.13 dB PSNR. The panorama-based MVC not only offers superior compression performance compared to the standard and adaptive MVC methods but it also reduces the random-access delay by 39%.

Note that the notations used in the different chapters are independent of each other.

1.10 References[1]

[1] O. Schreer, P. Kauff, T. Sikora, 3D Videocommunication: Algorithms, concepts and real-time systems in human centered communication, John Wiley & Sons, Inc. 1st edition, 2005.

[2] http://www.3dstereomedia.com/content/anaglyph-colorcode-chromadepth

[3]http://www.inition.co.uk/inition/product.php?URL_=product_software_ddd_tridef_range&SubCatID_=72&cur=USD, Dickson

[4] The Future of TV: Illusions of Reality in 3D, http://www.crc.gc.ca/en/html/crc/home/mediazone/eye_on_tech/2007/issue7/3dtv

[5] L. Lipton, The CrystalEyes Handbook, StereoGraphics Corporation, 1991.

[6] N. A. Valyus, Stereoscopy, New York: The Focal Press, 1962.

[7] http://3dcinecast.blogspot.com/2009/05/digital-hollywood-3d-needs-to-make.html

[8] A. Woods, T. Docherty, and R. Koch. "Image distortions in stereoscopic video systems," in SPIE Volume 1915: Stereoscopic Displays and Applications IV. 1993.

[9] L. M. J. Meesters, W. A. IJsselsteijn, and P. J. H. Seuntiens, "A survey of perceptual evaluations and requirements of three-dimensional TV", IEEE Trans. Circuits Syst. Video Technol. 14, 381–391, 2004.

[10] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV", Proc. SPIE 5291, 93–104, 2004.

[11] L. Zhang, "Stereoscopic image generation based on depth images for 3D TV," IEEE Trans. Broadcasting , vol. 51, no.2, pp191-199, 2005.

[12] M. Kawakita, T. Kurita, H. Kikuchi, and S. Inoue, "HDTV AXI-vision camera," in Proc. IBC, pp. 397–404, 2002.

[13] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav, "Depth keying," Proc. SPIE, vol. 5006, pp. 554–563, 2003.

[14] M.T. Pourazad, A. Bashashati, P. Nasiopoulos, and R. K. Ward, "Rapid Conversion of 2D Video to 3D Format Using Random Forests," Electronic Visualization and the Arts (EVA), 2010.

[15] P. Harman, J. Flack, S. Fox, and M. Dowley, "Rapid 2D to 3D conversion," Proc. SPIE, vol. 4660, pp. 78–86, 2002.

[16] S. H. Lai, C. W. Fu, S. Chang, "A generalized depth estimation algorithm with a single image," PAMI, Vol. 14(4), pp. 405-411, 1992.

[1] The references included in this chapter are generic references. More specific references to the subsequent chapters will follow at the end of each chapter.

[17] W. J. Tam, A. Soung Yee, J. Ferreira, S. Tariq, F. Speranza, “Stereoscopic image rendering based on depth maps created from blur and edge information,” In: Proceedings of Stereoscopic Displays and Applications XII, Vol. 5664, pp.104-115, 2005.

[18] W. J. Tam, F. Speranza, L. Zhang, R. Renaud, J. Chan, and C. Vazquez, “Depth image based rendering for multiview stereoscopic displays,” Role of information at object boundaries, Three-Dimensional TV, Video, and Display IV, Vol. 6016, pp. 75-85, 2005.

[19] Y. L. Chang, C. Y. Fang, L.F. Ding, S. Y. Chen, and L.G Chen, “Depth Map Generation for 2D-to-3D Conversion by Short-Term Motion Assisted Colour Segmentation,” IEEE International Conference on Multimedia and Expo, 2007.

[20] C.T. Lin, C.L. Chin, K.W. Fan, C.Y. Lin, “A novel architecture for converting single 2D image into 3D effect image,” 9th International Workshop on Cellular Neural Networks and Their Applications, pp.52-55,May 2005.

[21] T. Okino, H. Murata, K. Taima, T. Iinuma, K. Oketani, “New television with 2D/3D image conversion technologies,” Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems III, vol. 2653, pp. 96-103, 1996.

[22] D. Kim, D. Min, K. Sohn “Stereoscopic video generation method using motion analysis,” In: Proceedings of 3DTV Conf. pp. 1-4, 2007.

[23] I. Ideses, L.P. Yaroslavsky, and B. Fishbain, “Real-time 2D to 3D video conversion,” Journal of Real-Time Image Processing, Vol. 2, no. 1, pp. 3-9, 2007.

[24] M.T. Pourazad, P. Nasiopoulos and R. K.Ward, “Conversion of H.264-encoded 2D video to 3D format,” IEEE Conference on Consumer Electronics, Las Vegas, USA, Jan. 2010.

[25] ISO/IEC JTC1/SC29/WG11, “Depth Estimation Reference Software (DERS) 4.0,” M16605, July 2009.

[26] ISO/IEC JTC1/SC29/WG11, “Reference Softwares for Depth Estimation and View Synthesis,” m15377, April 2008.

[27] C. Fehn, "A 3D-TV system based on video plus depth information," *Signals, Systems and Computers*, vol. 2, pp. 1529–33, Nov. 2003.

[28] ITU-T Rec. H.264—ISO/IEC IS 14496-10, “Advanced video coding for generic audiovisual services,” v3, 2005.

[29] S. Grewatsch, and E. Miiller, "Sharing of motion vectors in 3D video coding," in Proc. ICIP 2004, vol. 5, pp. 3271–74.

[30] M.T. Pourazad, P. Nasiopoulos and R. K.Ward, “An H.264-based video encoding scheme for 3D TV,” European Signal Processing Conference (EUSIPCO), Florence, Italy. Sep. 2006.

[31] ISO/IEC JTC1/SC29/WG11, “Preliminary Requirements for 3D Video Support in MPEG”, WG 11 document N4559, Pattaya, December 2001.

[32] A. Vetro, P. Pandit, H. Kimata, A. Smolic and Y-K. Wang, “Joint Draft 8.0 on Multiview Video Coding,” Joint Video Team, Doc. JVT-AB204, July 2008.

[33] 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. ICIP, Canada, pp. 1717-20, Jul. 2006.

[34] P. Merkle, K. M¨uller, A. Smolic, T. Wiegand, “Experiments on coding of multiview video plus depth,” In: ITU-T & ISO/IEC JTC1/SC29/WG11 Doc. JVT-X064, Geneva, Switzerland, 2007.

[35] A. Vetro, S. Yea, A. Smolic, “Towards a 3D video format for auto-stereoscopic displays,” In: SPIE Conference on Applications of Digital Image Processing XXXI, 2008.

[36] K. Oh, S. Yea, A. Vetro, Y. Ho, “Depth reconstruction filter and down/up sampling for depth coding in 3D video,” 16(9), 747–750, 2009.

[37] Y. Morvan, D. Farin, P. H. N. de With, “Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images,” In: IEEE International Conference on Image Processing, San Antonio, TX, 2007.

[38] P. Merkle, Y. Morvan, A. Smolic, D. Farin, “The effects of multiview depth video compression on multiview rendering,” Image Communication 24(1-2), 73–88, 2009.

[39] S. Yea, A. Vetro, “Multi-layered coding of depth for virtual view synthesis” In: Picture Coding Symposium, Chicago, IL, 2009.

[40] S. Pastoor, and M. W¨opking, “3-D displays: A review of current technologies,” Displays,17:100–110, 1997.

[41] Y.Y. Yeh and L.D. Silverstein, “Limits of fusion and depth judgement in stereoscopic color displays,” Human Factors 32, pp. 45–60. 1990.

[42] I. Sexton, and P. Surman, “Stereoscopic and autostereoscopic display systems,” IEEE Signal Processing Magazine, pages 85–99, 1999.

CHAPTER 2: AN H.264-BASED SCHEME FOR 2D TO 3D VIDEO CONVERSION[2]

2.1 Introduction

The availability of three-dimensional television (3D TV) as a commercial product is not far from reality. 3D TV generates a compelling sense of physical real space for viewers by allowing on-screen scenes to emerge and penetrate into the viewers' space.

The successful introduction of 3D TV to the consumer market would not only rely on technological advances but also on the availability of a wide variety of 3D content. Thus, the creation of new 3D video content as well as the ability to convert existing 2D material to 3D format is of great importance. The latter depends on developing 2D-to-3D conversion tools capable of converting 2D video sequences into 3D ones. This would allow existing popular movies and documentaries to be watched on a 3D screen, and thus create a new market for content owners and providers.

To display 3D content, at least two temporally synchronized video streams (for the right and left eyes) are required. These two streams can be captured by two synchronized cameras. Alternatively, they can be rendered from a 2D video stream and its corresponding depth map, using a process known as depth image based rendering (DIBR) [1]. Thus, in principle, the conversion of 2D content to 3D video streams is possible if the depth information could be derived from the original 2D video sequence.

[2] A version of this chapter has been published. Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (2009) An H.264-based Scheme for 2D to 3D Video Conversion. IEEE Transactions on Consumer Electronic, vol. 55, no. 2, pp 742-748.

Depth map estimation techniques generally fall into one of the following categories: manual, semi automatic and automatic. For the manual methods, an operator manually traces the outlines of objects that are associated with an artistically chosen depth value. As expected, these methods are extremely time consuming and expensive. For this reason, semi automatic and automatic techniques are preferred. These techniques are designed based on the visual depth perception mechanism. There are several factors (referred as monocular depth cues) such as light and shade, relative size, motion parallax, interposition (partial occlusion), textural gradient and geometric perspective, which help the viewer to perceive the relative distance of objects within a scene. In fact, the depth map estimation techniques try to generate binocular parallax (disparity) using monocular depth cues.

A machine learning approach for estimating the depth map of 2D video sequences is proposed in [2]. Although the results of this approach are promising, it requires an operator to input the local depth information of some selected pixels with their color information. Extraction of depth from blur has also been explored [3]. The problem in this case is that depth is not the only cause of the blur in a picture. Other causes include motion, climate conditions and fuzziness of objects within a scene. There are also studies where depth values of the scene are obtained from edge information [4, 5].

The relationship between the distance of a moving object from the camera and its registered displacement in the captured consecutive frames (motion parallax) has also been utilized for estimating the depth map [6-8]. For objects traveling with the same speed, this approach assumes that the still camera always registers the closer objects as covering larger displacements (in pixel) than further objects. This approach is based on

the principle known as the Pulfrich effect [9, 10]. The Pulfrich effect is a psychophysical phenomenon wherein lateral motion of an object in the field of view is interpreted by the visual cortex as having a depth component, due to a relative difference in signal timings between the two eyes. To utilize this principle, the study in [8] uses modified a time difference method (MTD) to detect horizontal motion of objects and determine the image-presentation time-delay to create a stereo pair. The MTD method does not work for images containing objects with complicated motion.

The study in [6] uses color segmentation and the KLT (Kanade-Lucas-Tomasi) feature tracker to estimate motion information. Then, the depth map is approximated based on the motion information. In this approach, factors such as camera movement, scene complexity and the magnitude of estimated motion are used for converting motion information of each frame into depth map information. The complexity of motion information extraction in [6] does not allow real-time implementation of this technique. In [7], motion estimation is based on the H.264 standard, but uses fixed (rather than variable) block-size matching technique. This study assumes that the motion of every object is directly proportional to its distance from the camera, thus the depth map is approximated as a constant factor of the estimated motion. Unfortunately, this is only true for a relatively small part of real life footage (when the camera is panning across a stationary scene, or a still camera captures the scene with moving objects). Otherwise, when the objects and camera are both moving, there would be ambiguity in depth estimation based on motion information. The other issue in [7] is related to the accuracy of H.264-estimated motion vectors when they are used to derive the objects' motion. The principle idea behind motion estimation process in H.264/AVC and other standards relies

on maximizing the compression performance and not on obtaining accurate estimates of the objects' displacement in the scene. Thus, not all motion vectors can be used to accurately estimate the depth unless they reflect the objects' displacement.

In this paper, we present an effective scheme that finds an approximate depth map of the scene using the motion information of the recorded video. The relative motion between two consecutive frames is derived (by the H.264/AVC motion estimation process) at quarter pixel accuracy by a block matching technique where the block sizes are dynamically adjusted according to the video content. When a moving camera captures a scene with moving objects, our proposed scheme provides a solution to resolve the motion ambiguity problem for estimating the scene's depth map.

To resolve the issue regarding the accuracy of motion vectors, we propose an algorithm that examines the motion vectors and (whenever necessary) it properly modifies their values, to make sure the estimated motion vectors reflect the displacement of objects. Since the motion estimation procedure is based on the block-matching technique, there will be depth ambiguity between the foreground and the background at the object boundaries. For this reason, our algorithm re-evaluates and modifies the estimated values of the motion vectors of object-boundary pixels. Then, the absolute horizontal value of the estimated motion vectors are used as initial depth values. To enhance the visual depth perception, we propose to increase the contrast among the initial approximated depth values by using a non-linear scaling model. Finally, a DIBR technique is used to render the stereoscopic video sequences based on the approximated depth map and 2D video.

The rest of this paper is organized as follows. Section 2.2 elaborates on our 2D-to-3D conversion scheme, followed by Section 2.3 which presents the performance evaluation of our scheme and discusses the results in detail. Section 2.4 presents the conclusions.

2.2 Proposed 2D-TO-3D Conversion Scheme

The visual depth perception relies upon the fact that when viewer watches a scene using both eyes, two slightly different images are projected on the left and right eye retinas, each from a slightly different viewpoint. Then the brain fuses the two images to give depth perception, and the viewer sees one solid scene instead of two slightly different projections. The perceived image with depth contains everything which is present in the two individual viewpoint images but adds something that is present in neither of them: an intrinsic feeling of depth, distance and solidity.

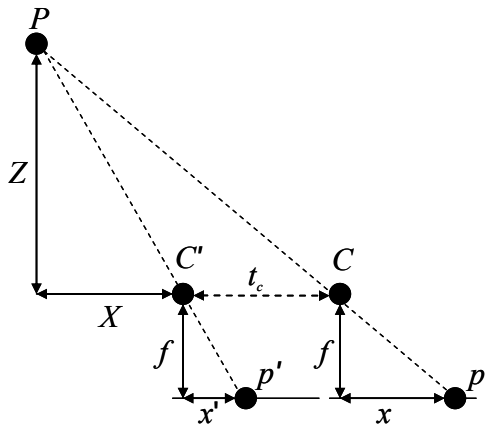

Figure 2.1 Stereo geometry for two identical parallel cameras.

The differences between the left and right eye viewpoint images are due to the fact that objects are relatively displaced according to their distance from the viewer. In digital stereoscopic videos, where the function of the retina in the eye is taken over by the

lenses of two synchronized cameras, this displacement is referred as disparity. Figure 2.1 illustrates a stereoscopic set up where two identical parallel cameras capture a scene-point *P*. *p* and *p′* are the images of *P* as captured by the right (*C*) and left (*C′*) cameras, respectively. *x* and *x′* are the coordinates of *p* and *p′*, respectively. *Z* is the distance of point *P* from the cameras (depth), t_c is the distance between the two cameras (baseline), and *f* is the focal length of cameras. *x* and *x′* are inversely proportional to the depth, Z. Also their difference (*d*), known as disparity, is inversely proportional to *Z* as follows [11]:

$$\boldsymbol{d = x - x' = \frac{f t_c}{Z}} \tag{2-1}$$

This relationship shows that the depth of a point *P* can be easily obtained if the disparity *d* is known.

For our case where the 2D scene captured by only one camera, we shall obtain the depth *Z* of a moving point *P* using the registered displacement that results from its motion. In other words, if the left and right side frames shown in Figure 2.1 are aligned in the time domain, then *p* and *p′* would be two consecutive images of point *P* when it moves. Since the time delay between two consecutive frames is small (as in the studies [6-8]), we assume that the displacement due to motion in the 2D case is equivalent to the disparity in a stereoscopic setup. Then, the depth of point *P* is obtained using (2-1) and the displacement ($x-x'$) between two consecutive frames, is obtained using the H.264/AVC-estimated motion vectors. The following sub-sections provide detailed description of our proposed algorithm.

2.2.1 Motion vector estimation

In our algorithm, the H.264/AVC motion vectors are used to estimate the displacement of objects in the scene. H.264/AVC-based motion vectors (MVs) are estimated using variable block sizes of 16x16, 16x8, 8x16, 8x4, 4x8 and 4x4 pixels, and quarter-pixel matching accuracy [12]. These two features of the H.264/AVC standard (i.e., variable block size and quarter-pixel matching accuracy) have been shown to yield motion vector precision that is far superior to that of any previous standards [12]. An additional advantage of using the H.264/AVC standard is its nomination as the platform for 3D TV applications [13]. In that regard, the proposed scheme will be compatible with future 3D networks and players, and thus could be implemented at the receiver-end where motion vectors will be readily available at no additional computational cost. The existing approach in [7] uses only 4x4 block sizes. Such an implementation will hinder a standard decoder in producing the necessary information. Any effort to force the encoder to use 4x4 blocks will significantly decrease the compression performance, and will significantly increase the computational complexity of the overall system.

Moreover, the use of small block-sizes results in many wrong matches due to ambiguities and noise, but has the advantage of preserving object shapes with relatively fine details. In contrast, the use of large block sizes cuts down on the wrong matches, but has the potential of blurring the objects boundaries [11]. For the above reasons, in our study, a variable block size is used to deal with the basic trade-off involved in selecting the best window size.

2.2.2 Camera motion correction

One of the potential problems that may arise in motion-based depth estimation is depth ambiguity when both the objects and the camera are in motion. In this case, an object that is captured as if it has larger motion than others is not necessarily closer to the camera, since the camera may have moved in the opposite direction from the object. To resolve this issue, the camera motion needs to be estimated and the motion information registered by the camera should be corrected accordingly.

In the case of camera panning, the estimated motion vectors of the stationary areas of a scene are equal to the motion of the camera. These stationary areas are often flagged by the H.264/AVC motion estimation process as 'Skip Mode'. The 'Skip Mode' is used for the 16x16 blocks, where the motion characteristics of the block can be effectively predicted from the motion of its neighboring blocks, and the quantized transform coefficients of the block are all zeros. When a block is skipped, the transformed coefficients and the motion data are not transmitted, since the median of the motion vectors of the surrounding blocks, known as predicted motion vector (MV_p), is used as the motion vector of the block.

As long as there is no camera motion, the predicted motion vector of a skipped block is zero. Moreover, when camera panning is present, the predicted motion vectors of the skipped blocks are all equal to a unique non-zero value (which represents the camera motion). Since most of the skipped blocks over the entire frame are blocks that contain background areas, in our proposed scheme the predicted motion vector (MV_p) with the maximum occurrence is used to estimate the value of camera motion. In order to find such a vector, we compute the histogram of MV_ps of the skipped blocks for each frame.

The MV_p that corresponds to the maximum of the histogram is recognized as the camera motion (panning). The net motion of each object is extracted by subtracting the camera motion from the MVs of all blocks within the frame. This procedure cannot be accommodated in [7] since only 4x4 blocks are used and the size of the skip-mode blocks is 16x16.

Besides panning, camera zoom-in/out can also cause depth ambiguity. To address this issue, we check the tendency of MVs, in the four corners of the frame to detect zoom in/out [6]. Then the estimated MVs are scaled accordingly [6]. Note that zoom-in/out may cause reverse depth or eye fatigue if not corrected in depth estimation.

2.2.3 Correction of displacement estimates

H.264/AVC coding obtains the motion vectors by maximizing the compression performance rather than the accuracy of the estimated motion. Thus, two matching blocks related to a motion vector may not even contain the exact same object or part of the object in the scene. For such a case, the displacement of the object due to motion would not be correctly calculated using the obtained MV.

To check if a motion vector points to the same object (or part of it) in two consecutive frames, our proposed scheme compares the motion of the block with that of its surrounding blocks. This is done by finding the difference between the MV of the block and the median MV of the neighboring blocks (MV_m). If the difference is greater than a pre-defined threshold, the value of the MV is readjusted by making it equal to the MV_m vector. This is necessary, unless the block includes the boundary pixels of a moving object. To determine if this is the case, the "residue frame" is computed as the difference

between the luma of the current frame (which includes the block) and that of the previous frame. In this residual frame, the edges of moving objects appear thicker and with higher density compared to static objects and the background. If the variance of the corresponding block in the "residue frame" is greater than a predefined threshold, the estimated motion vector is not modified since it is considered to be part of a moving object's border. Otherwise, for correction, the median of MVs of adjacent blocks is assigned as the estimated MV of the block.

2.2.4 Displacement correction of object borders

In our study, we use the absolute value of the horizontal component of the motion vectors (i.e., $abs(MV_x)$) for estimating the depth map. This is because disparity is the horizontal displacement between the two camera images (as shown in Figure 2.1). Since all the pixels within each matching block are assumed to have the same amount of motion, the $abs(MV_x)$ is assigned to each pixel within a block. This assumption, however, is not valid for blocks that include both stationary background pixels and moving object pixels. For such blocks, a different procedure should be used, otherwise the resulting object borders in the 3D video constructed using the estimated depth may appear blurred, and small details or even entire objects may be removed.

A computationally expensive solution is to perform pixel-based motion vector estimation for the object-border pixels. We propose an alternative solution which detects the blocks with non-zero motion vector, then, classifies each pixel within each of these blocks as a background pixel or an object pixel. This classification is achieved by first calculating the average luma intensity of the corresponding block in the "residue frame". Then, the pixels within the block (in the current frame) whose corresponding pixels in the

"residue frame" have luma intensities lower than the calculated average are marked as background pixels and the rest are marked as object pixels. The estimated $abs(MV_x)$ is assigned to the object pixels, while the background pixels are assigned the median of the $abs(MV_x)$ of the surrounding pixels that are not object pixels. The background pixels within the block might be utilized in the motion estimation process only if the updated $abs(MV_x)$ has been assigned to them. In our method, we start the motion correction procedure from the corner background pixels within the block to employ non-object pixels of the surrounding blocks in the process. This will result in more accurate motion vector estimates for background pixels located inside the blocks. After pixel-based motion vector estimation, the absolute horizontal value of the motion vector of each pixel is used as its initial depth value.

2.2.5 Perceptual depth enhancement

To enhance the visual depth perception, we propose applying a non-linear scaling model to the initial approximated depth values, i.e., abs(MVx)s. Since there is a non-linear relation between visual depth perception (disparity) and the distance of an object, the proposed scaling factors are defined such that the further the object is, the smaller the scaling factor. This will increase the contrast among depth values and enhance the visual depth perception.

In our model we assume that there are N uniformly spaced depth layers within a scene, i.e., within [Z_{near} and Z_{far}]. A set of scaling factors is defined as:

$$S(i) = \frac{i}{N-1}\left(1 - \frac{Z_{far}}{Z_{near}}\right) + \frac{Z_{far}}{Z_{near}} \qquad 0 \le i \le N\text{-}1 \qquad (2\text{-}2)$$

where S is the scaling factor and i ranges from layer 0, which corresponds to Z_{near} (here i=0, $S(0) = Z_{far} / Z_{near}$) to layer N-1 which corresponds to Z_{far} (i=N-1, $S(N\text{-}1) = 1$). To generate the enhanced depth map, the estimated depth values, i.e., $abs(MV_x)$ are sorted and categorized to N uniformly spaced layers. If $abs(MV_x)$ belongs to the i^{th} category, its value is scaled as follows:

$$D = abs(MVx)S(i) \tag{2-3}$$

where D is the enhanced depth value and $S(i)$ is the scaling factor at the i^{th} depth layer.

Using the approximated depth map and the 2D video sequence, the stereoscopic pair images can be rendered via the depth-image-based rendering algorithm proposed in [1]. This algorithm includes a depth map smoothing process (using asymmetric Gaussian filter) to resolve the occlusion problem of depth image-based rendering. In our implementation, only the right-eye stream is rendered (based on the estimated depth map and the 2D video sequence), and the original 2D video is used as the left-eye stream [5].

2.3 Performance Evaluation

The performance of our depth map estimation method is tested using two 2D video sequences known as “Interview” and “Orbi”. The true depth maps of the test streams have been captured by a 3D-depth range camera (Zcam) [14]. The 2D streams of Interview and Orbi are 10 seconds and 5 seconds long, respectively, with 720×576 pixels resolution and 4:2:2 YUV format. The depth consists of luma information only.

In our experiments, the motion between two consecutive frames is estimated using the JM 12.2 version of the H.264/AVC standard.

We compare our method with the method presented in [7]. Since the recorded depth for each pixel is an integer number between 0 and 255 (where 255 represents the shortest distance from the camera), we assume there are 256 depth layers for the perceptual depth enhancement step, and also the estimated depth maps of both methods are normalized accordingly.

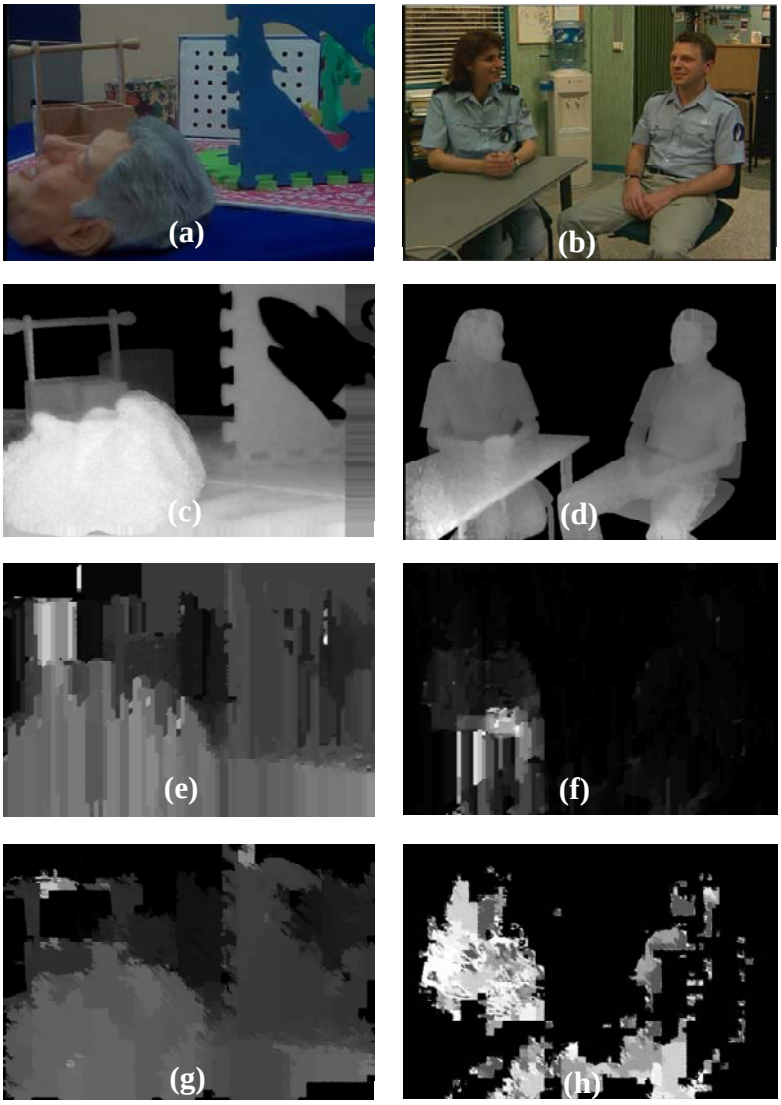

Figure 2.2 2D video sequence (a and b), recorder depth map (c and d) estimated depth map by [7] (e and f), and estimated depth map by our approach (g and h).

Figure 2.2 shows a snap-shot of the original 2D stream, the original depth map, and the estimated depth maps generated by [7] and our approach. We observe that our approach generates a more realistic depth. However both techniques fail to estimate depth maps for static objects (e.g., the table in Figure 2.2). This, however, is the drawback of all motion-based depth estimation techniques.

The visual quality of the resulting 3D video streams using our method and the one presented in [7] is subjectively tested against the original depth map based on the ITU-Recommendation BT.500-11 [15]. Fifteen people graded the videos from 1 to 10 in terms of 3D visual perception and visual quality. The evaluation is performed using a SeeReal, C_n 3D display. Table 2.1 illustrates the subjective test scores.

Table 2.1 Subjective test scores for test streams.

Subjective Score (out of 10)		Interview	Orbi
3D Perception	Actual depth	5.50	7.08
	Our method	6.51	7.60
	Existing method	5.49	7.28
Visual Quality	Actual depth	7.91	6.93
	Our method	6.65	6.15
	Existing method	6.04	4.88

The original stereoscopic video had the highest scores in terms of visual quality and our method yielded the highest scores in terms of 3D visual perception. These tests show that: i) the approximated depth map obtained by our method provides the best 3D visual perception and ii) the visual quality of the results by our technique is higher than the one obtained by [7].

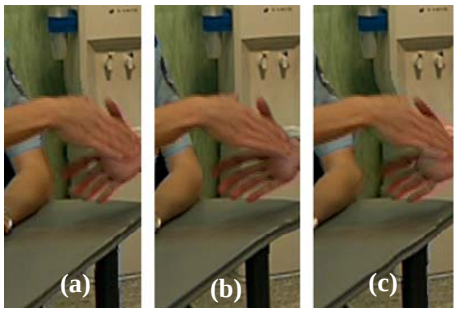

Figure 2.3 Rendered right image based on real depth map (a), estimated depth map by our approach (b), estimated depth map by [7] (c).

Since our technique and the one presented in [7] are both capable of approximating the depth information only for areas with moving objects, watching the resultant stereoscopic video streams tends to create visual discomfort for viewers. On the other hand, since the depth values are prominent in moving-object boundaries, the 3D visual perception is enhanced. Figure 2.3 demonstrates this effect clearly by comparing the right images rendered based on the real depth map, our estimated depth map and the approximated depth map obtained by [7]. As it can be observed, the fingers of the moving hand are longer in the image rendered using the estimated depth of our technique and the one presented in [7] compared to the image obtained by the real depth map. This has the effect of increasing the 3D perception when it is watched on 3D display.

For the quantitative analysis, we chose to compare the quality of the stereoscopic videos synthesized by our technique, the technique proposed in [7], and the stereoscopic videos rendered from the actual (recorded) depth map.

Figure 2.4 illustrates the five different PSNR (Peak signal-to-noise ratio) comparisons that we chose for our analysis. In one scenario we compare the right view generated by our method and the one by [7] with the right view rendered based on the recorded depth map (b and d in Figure 2.4). These comparisons show how close the average quality of the estimated 3D views is to the actual ones. Note that, in this case,

higher PSNR values indicate better visual quality. Table 2.2 shows the obtained average PSNR values. We observe that our method outperforms the proposed method in [7] by 1.7 dB to 1.84 dB.

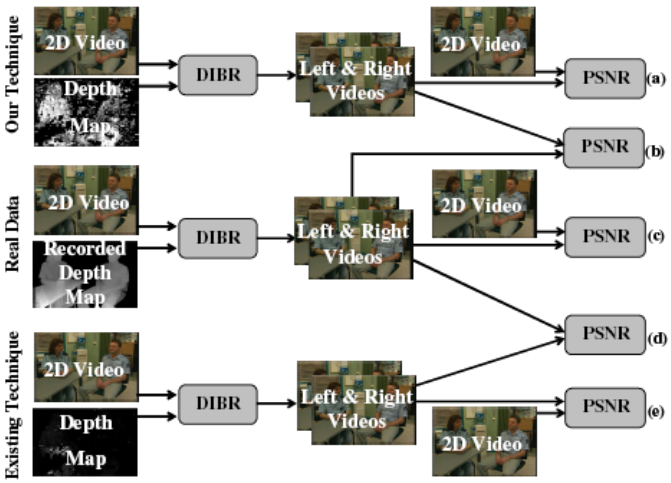

Figure 2.4 Quantitative analysis of the results.

Table 2.2 Average PSNR comparison case b and d in Figure 2.4.

Average PSNR (dB)	Interview	Orbi
3D views based on our method vs actual 3D views	36.31	31.8
3D views based on existing method vs actual 3D views	34.47	30.1

In addition to the above, we also compare the generated right views with the actual 2D video stream (a, c and e in Figure 2.4). These comparisons show how effectively the two different techniques generate the depth perception. In this case, since there is no depth present in the 2D video stream, large PSNR values indicate failure in adding significant depth perception to the stream. Table 2.3 shows the average PSNR values obtained for this case. As expected, we observe that the actual 3D views have the least similarity with the 2D video (no depth perception). More importantly, our method yields a PSNR value very similar to the actual 3D view, while the PSNR value obtained

by [7] is higher than the original recorded depth. This conveys the fact that the depth map estimated by [7] creates the least 3D perception.

Table 2.3 Average PSNR comparison case a, c and e in Figure 2.4.

Average PSNR (dB)	Interview	Orbi
Right view rendered based on the actual depth map vs actual 2D view	32.27	27.85
Right view rendered based on our estimated depth map vs actual 2D view	32.41	27.98
Right view rendered based on the estimated depth map the by existing method vs actual 2D view	36.99	33.34

For further quantitative analysis, we also compute the percentage of the badly matched pixels in the estimated depth obtained by our scheme and [7] as:

$$B = \frac{1}{N}\sum_{(x,y)}(|\ D(x,y) - D_r(x,y)\ |> Th) \tag{2-4}$$

where N is the number of all pixels within the depth map, D is the estimated depth map, D_r is the recorded depth map and Th is the error tolerance. In our experiment we use Th=1 [16]. The results show the percentage of correctly matched pixels is 50% (Interview) and 47% (Orbi) for our method. For [7], the percentage of correctly matched pixels is 34% (Interview) and 27% (Orbi). The comparison confirms that our method outperforms the existing method by 16% to 20%.

2.4 Conclusion

We present a new and efficient method that estimates the depth map of a 2D video sequence using its H.264/AVC estimated motion information. This method exploits the existing relationship between the motion of objects and their distance from the camera, to estimate the depth map of the scene. Our proposed method modifies the motion information based on the characteristics of the 3D visual perception. In this study,

the 2D horizontal motion is taken as the displacement between the right and left frames of a 3D set up. However, for cases involving camera motion, our proposed method provides solutions for issues regarding displacement of object borders and false displacement estimates. One advantage of the proposed approach is that it can be implemented in real-time at the receiver-end, without increasing the transmission bandwidth requirements. Performance evaluations show that our method outperforms the other existing H.264 motion-based depth map estimation technique by 1.7 to 1.84 dB PSNR, i.e., our method provides better approximation for the scene's depth map.

The visual quality of the created 3D stream was also tested subjectively, by having viewers watch the generated 3D streams on a stereoscopic display. The subjective tests show that the 3D streams created by our approach provide viewers with superior 3D experience. Moreover, in terms of visual quality, our approach outperforms the other existing H.264-based depth estimation method.

2.5 References

[1] L. Zhang, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcasting*, vol. 51, no.2, pp191-199, 2005.

[2] P. Harman, J. Flack, S Fox, and M Dowley, "Rapid 2D to 3D Conversion," Proc. SPIE, vol. 4660, pp. 78–86, 2002.

[3] S. H. Lai, C. W. Fu, & S. Chang, "A generalized depth estimation algorithm with a single image," *PAMI*, vol. 14(4), pp. 405-411, 1992.

[4] W. J. Tam, A. Soung Yee, J. Ferreira, S. Tariq, and F. Speranza, "Stereoscopic image rendering based on depth maps created from blur and edge information," *Stereoscopic Displays and Applications XII*, vol. 5664, pp.104-115, 2005.

[5] W. J. Tam, F. Speranza, L. Zhang, R. Renaud, J. Chan, and C. Vazquez, "Depth image based rendering for multiview stereoscopic displays: Role of information at object boundaries", *Three-Dimensional TV, Video, and Display IV*, vol. 6016, pp. 75-85, 2005.

[6] D. Kim, D. Min, K. Sohn, "Stereoscopic video generation method using motion analysis," *3DTV Conf.* pp. 1-4, 2007.

[7] I. Ideses, L. P. Yaroslavsky, and B. Fishbain, "Real-time 2D to 3D video conversion," *Journal of Real-Time Image Processing*, vol. 2, no. 1, pp. 3-9, 2007.

[8] T. Okino, H. Murata, K. Taima, T. Iinuma, K. Oketani, "New television with 2D/3D image conversion technologies," Proc. SPIE, *Stereoscopic Displays and Virtual Reality Systems III*, vol. 2653, pp. 96-103, 1996.

[9] C. Pulfrich, "Die Stereoskopie im Dienste der isochromen und heterochromen Photometrie," *Naturwissenschaften,* vol. 10, no.34, pp 735–743, 1922.

[10] DC. Burr, J. Ross, "How does binocular delay give information about depth?" *Vision Research*, vol. 19, pp. 523–532, 1979.

[11] D. Scharstein, View Synthesis Using Stereo Vision (Lecture Notes in Computer Science), Springer, 1999.

[12] Richardson, Iain E.G., H.264 and MPEG-4 Video Compression: Video Coding for Next generation Multimedia, John Wiley & Sons, Inc., England, 2003.

[13] A. Vetro, P. Pandit, H. Kimata and A. Smolic, "Joint Multiview Video Model (JMVM) 5.0," ISO/IEC JTC1/SC29/WG11/N9214, Lausanne, Switzerland, Jul. 2007.

[14] C. Fehn, "A 3D-TV system based on video plus depth information," *Signals, Systems and Computers*, vol. 2, pp. 1529–33, 2003.

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

[16] D. Scharstain, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,”

CHAPTER 3: GENERATING THE DEPTH MAP FROM THE MOTION INFORMATION OF H.264-ENCODED 2D VIDEO SEQUENCE[3]

3.1 Introduction

Three-dimensional television (3D TV) generates a compelling sense of physical real space for the viewers by allowing on-screen scenes to emerge and penetrate into the viewers' space. Viewers thus feel that they are part of the scene they are watching. It is predicted that by commercialization of 3D TV applications, another revolution will take place in TV's history (the last one being the introduction of digital video broadcasting).

The history of 3D TV can be traced back to 1920s, when the first experimental 3D TV set-up was built [1]. Since then, several attempts have been made to introduce this technology into the market. Despite the immense keenness towards 3D, the great expectations of viewers, content providers and distributors have not yet been fulfilled. The main drawbacks were the discomfort of the viewers (headaches, eyestrain) due to the poor quality content, the low-tech display systems and the high costs involved in the production and distribution of 3D content.

The successful introduction of 3D TV to the consumer market relies not only on technological advances but also on the availability of a wide variety of 3D content. Thus the production of 3D-format videos is important. Equally important is the ability to convert existing 2D material to 3D format. This allows the existing popular movies and

[3] A version of this chapter has been accepted for publication. Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (2010) Generating the Depth Map from the Motion Information of H.264-Encoded 2D Video Sequence. EURASIP Journal on Image and Video Processing.

documentaries to be watched on 3D screens. Converting 2D content to 3D video streams is possible if the depth information is estimated from the original 2D video sequence. Using the depth information, 3D video content in the stereoscopic format (two temporally synchronized video streams, one for the right and another for the left eye) can be rendered from the 2D video stream, via a process known as depth image based rendering (DIBR) [2].

Depth map estimation techniques generally fall into one of the following categories: manual, semi automatic and automatic. For the manual methods, an operator would manually draw the outlines of objects that are associated with an artistically chosen depth value. As expected, these methods are extremely time consuming and expensive. For this reason, semi automatic and automatic techniques are preferred for the depth map estimation. These techniques are designed based on the human visual depth perception mechanism. There are several factors (referred as monocular depth cues) such as light and shade, relative size, motion parallax, interposition (partial occlusion), textural gradient and geometric perspective, which help the human visual system perceive the relative distance of objects within a real scene. In fact, depth map estimation techniques try to use these monocular depth cues and imitate the human visual system when estimating the distance between objects to generate binocular parallax (disparity) for the viewer.

A machine learning approach for estimating the depth map for 2D video sequences is proposed in [3]. Although the results of this approach are promising, it requires an operator to input the local depth information of some selected frames. Extraction of depth from blur has also been explored by researchers [4]. The problem in

this case is that depth is not the only cause of the blur in a picture. Other reasons include motion, climate conditions and fuzziness of objects within a scene. The estimation of depth based on the edge information has also been studied [5, 6]. Another group of researchers has utilized the motion/edge-corrected color-segmentation via a K-means algorithm to estimate the depth map [7]. This algorithm does not provide solutions when the camera is moving or when the objects have complicated motion. Also, since supervised image segmentation is implemented, when the number of objects within the scene is higher than a pre-specified number, the algorithm cannot recognize the silhouette of all objects and estimate their relative distance from the camera. The study in [8] applies an unsupervised image segmentation algorithm to separate the objects. Then to decide on the depth value of each object, the proposed algorithm uses the assumption that the objects on the top part of the image are further from the viewer and the ones at the bottom part of image are closer. This assumption, however, is not valid for all video sequences.

There is a relationship between the distance of moving objects from the camera and their registered motion, which has been utilized in previous studies for motion estimation [9] and also for depth map approximation [10- 12]. For objects traveling with the same speed, but different distance from a still camera, this relationship implies that the camera registers larger displacements (in pixel) for the closer objects to the camera. This approach is based on the principle known as the Pulfrich effect [13, 14]. The Pulfrich effect is a psychophysical phenomenon wherein lateral motion of an object in the field of view is interpreted by the visual cortex as having a depth component, due to a relative difference in the signal timings between two eyes. To utilize this principle, the

study in [10] uses a modified time difference method (MTD) to detect the horizontal motion of objects and determine the image-presentation time-delay for synthesizing a stereo pair. The MTD method does not work for images containing objects with complicated motion.

The study in [11] uses color segmentation and the KLT (Kanade-Lucas-Tomasi) feature tracker to estimate motion information. Then, the depth map is approximated based on the estimated motion information. In this approach, factors such as camera movement, scene complexity and the magnitude of the estimated motion are used for converting the motion information of each frame into depth map. The facts that this method is not based on existing video coding standards and it involves a relatively complex motion estimation extraction process, do not allow its cost-effective real-time implementation at the decoder side. In [12], motion estimation is based on the H.264 standard, but uses fixed (rather than variable) block-size matching technique. This study assumes that the motion of every object is directly proportional to its distance from the camera, thus the depth map is approximated as a constant factor of the estimated motion. Unfortunately, this is only true for a relatively small part of real life footage (when the camera is panning across a stationary scene, or a still camera captures a scene with moving objects). Otherwise, when the objects and the camera are both moving, there would be ambiguity in the depth estimation based on the motion information. The other issue in [12] is related to the accuracy of H.264-estimated motion vectors when they are used to derive the objects' motion. The principle idea behind the motion estimation process in H.264/AVC and other standards relies on maximizing the compression performance and optimizing rate distortion and not on obtaining accurate estimates of the

objects' displacement in the scene. Thus, not all motion vectors can be used to accurately estimate the depth unless they reflect the objects' displacement.

In this paper, we present an effective scheme that finds an approximate depth map of the scene using the motion information of the H.264 encoded video which is derived (at quarter pixel accuracy) via matching blocks with different sizes, where the sizes dynamically adjust to the video content. This proposal is an improved version of algorithm in [12] from two aspects: i) generalizing previous assumption that any motion is directly proportional to distance from camera, and ii) improving accuracy from motion vectors in H.264.

To resolve the issue regarding the accuracy of motion vectors, we propose an algorithm that examines the motion vectors and (whenever necessary) properly modifies their values, to ensure that the values of the motion vectors reflect the displacement of objects. When a moving camera captures a scene with moving objects, our proposed scheme provides a solution to estimate the motion of moving objects and uses this information to find the scene's depth map [15]. Since the motion estimation procedure is based on the block-matching technique, there will be depth ambiguity between the foreground and the background at the object boundaries. To solve this problem, our algorithm first adopts a color-texture segmentation algorithm known as JSEG to properly distinguish between the different object-regions [16]. Then it re-evaluates and modifies the estimated values of the motion vectors of the object-boundary pixels. To ensure that the final estimated depth map is smooth and free of artifacts, our algorithm assumes that each segmented object has a unique depth value, and accordingly corrects the estimated

motion of the object-body pixels using the object-boundary pixels. This enhances the visual quality of 3D video that is rendered based on the estimated depth map.

After refining the motion vectors in different stages, the absolute horizontal values of the refined motion vectors are used to approximate initial depth values. To enhance the visual depth perception, we propose to increase the contrast among the initial approximated depth values by using a non-linear scaling model.

Finally, a DIBR technique is used to render the 3D videos based on the approximated depth map and 2D video. The 3D videos created using this scheme are of the stereoscopic format and are supposed to be watched on 3D TVs in the sense of stereoscopic TV. Note in this paper by "3D TV", we mean 3D TV in the sense of stereoscopic TV.

The rest of this paper is organized as follows. Section 3.2 provides background information on the principal idea behind this study. Section 3.3 elaborates on our 2D-to-3D conversion scheme. Section 3.4 presents the performance evaluation of our scheme and discusses the results in detail. Section 3.5 presents the conclusions.

3.2 Background

In 3D video capturing using stereo camera set-up, the displacement between the left and right camera images is directly related to the distance of objects from the camera. This displacement, which is known as disparity [17], creates an intrinsic feeling of depth for viewers watching stereo videos. Basically when two slightly different images are projected on the left and right eye retinas (each from a slightly different viewpoint) the

brain fuses these images such that the perceived image represents everything included in two images, but in a three-dimensional format.

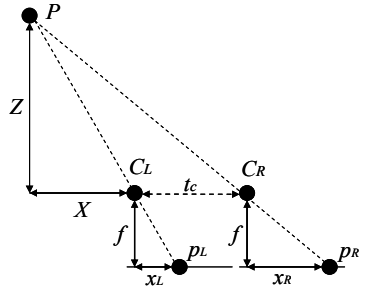

Figure 3.1 Stereo geometry for two identical parallel cameras.

Figure 3.1 illustrates how the disparity is related to depth for two identical parallel cameras. P is a scene point whereas p_L and p_R are its images captured by the left (C_L) and right (C_R) cameras, respectively.

For this case, assuming the images are rectified, the disparity, d ($=x_R-x_L$), is expressed as in [17]:

$$d = x_R - x_L = \frac{ft_c}{Z} \tag{3-1}$$

where x_R and x_L are the coordinates of p_L and p_R, respectively. Z is the distance of point P from the cameras (depth), t_c is the distance between the two cameras (baseline), and f is the focal length of the cameras. In a practical stereoscopic camera set-up, t_c is usually equal to the average distance between the human eyes and f is chosen based on the region of the interest within the scene.

The relationship in (3-1) shows that the depth of a point P (i.e., Z) can be easily obtained if the disparity (x_R-x_L) is known. Figure 3.2 illustrates the relationship between

depth and disparity. Z_{near} and Z_{far} in Figure 3.2 are respectively referred to the nearest and furthest distances within the scene where still 3D perception is possible for the human visual system. In other words, only objects within [Z_{near} and Z_{far}] are perceived as 3D, while the ones outside this range are viewed as 2D objects. In practical 3D TV applications, Z_{far} does not exceed 5 meters, since the depth of objects beyond this distance from the camera are not visually perceptible on a 3D-display. Also Figure 3.2 shows that the 3D visual depth perception of the scene will not change if the viewer moves along horizontal coordinate of the scene.

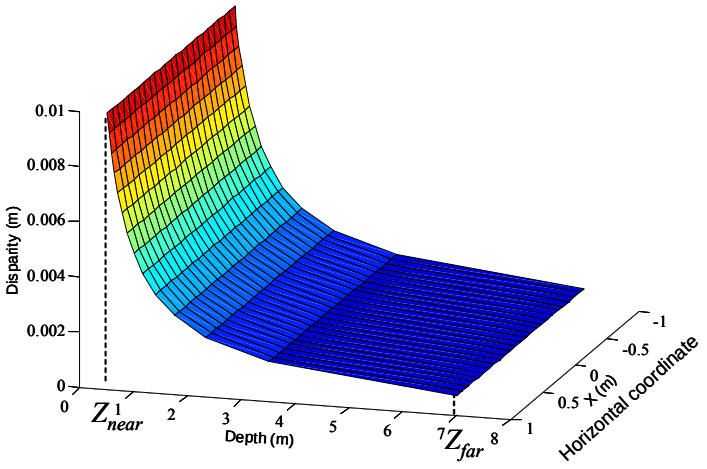

Figure 3.2 Relationship between disparity and depth for sample parallel cameras (tc=0.1 m and f=0.05 m).

For our case where the 2D scene is captured by only one camera, we shall obtain the depth *Z* of a moving point *P* using the registered displacement that is resulted from its motion. In other words, if the left and right side frames shown in Figure 3.1 are aligned in the time domain to form two consecutive frames, then p_R and p_L would be two consecutive images of moving point *P,* and x_R-x_L would be the displacement between them. Since the time delay between two consecutive frames is small (as in the studies

[10]-[12]), we assume that the displacement due to motion in the 2D case is equivalent to the disparity in a stereoscopic setup. Then Z, the depth of point P, can be calculated using (3-1), if $x_R - x_L$ is measured. The displacement $(x_R - x_L)$ between two consecutive frames could be obtained using information embedded in the motion vectors of the encoded video, assuming ft_c is constant. The following section provides detailed information on this process.

3.3 Proposed Scheme

To find the displacement of objects within a scene captured by single camera, we use the motion vector estimation procedure of the H.264 standard. Since H.264 motion vector estimation is block-based (i.e., it measures the displacement of a block and not a point or object), we propose correction steps that re-evaluate and refine the estimated motion vectors in order to calculate the motion vectors for the objects within the scene. Then the resulting object motion vectors are transformed to depth information. The following subsections elaborate on different steps of our proposed scheme.

3.3.1 Motion vector estimation

H.264/AVC-based motion vectors (MVs) are estimated using variable block sizes of 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 pixels, at quarter-pixel matching accuracy [18]. These two H.264 features (variable block size and quarter-pixel matching accuracy) have been shown to yield motion vector precision that is far superior to those of any previous standards [18]. An additional advantage of using the H.264/AVC standard is the fact that H.264 has been chosen as the platform for 3D TV applications [19]. This means that the proposed scheme will be compatible with future 3D networks and players, and

also could be implemented at the receiver-end where the motion vectors are readily available at no additional computational cost. The existing approach in [12] forces the encoder to use only 4x4 block sizes, and this will significantly decrease the compression performance, and will increase the computational complexity of the overall system.

The use of small block sizes results in many wrong matches due to ambiguities and noise, however it preserves object shapes with relatively fine details. In contrast, the use of large block sizes cuts down on the wrong matches, but may blur the objects boundaries [17]. For the above reasons, in our study, a variable block size is used to deal with the basic trade-off involved in selecting the best window size. Figure 3.3a shows a frame from the "Orbi" sequence and Figure 3.3b illustrates the magnitude of estimated motion vectors for this frame (the brighter the region, the higher is the magnitude and vice versa).

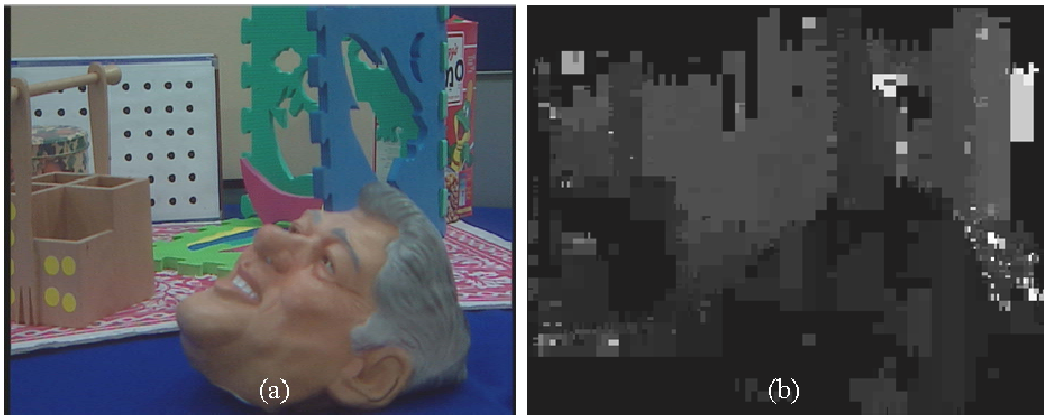

Figure 3.3 2D video frame (a) and magnitude of estimated motion vectors (b).

In H.264, depending on the application and content, some frames are selected to be compressed as I-frames. For compressing I-frames only intra prediction is utilized and no motion estimation is involved. Thus, to retrieve the motion information, we utilize the estimated motion vectors of the P-frame just after each I-frame. Since the MVs for the P-

frame are estimated in relation to the I-frame (as reference), a simple solution for finding MVs of the I-frame, is to invert the MVs of the P-frame. This will give us the displacement of some overlapped blocks of the I-frame. To estimate the approximated MVs of each separate block of the I-frame, the MVs of overlapped blocks are weighted and averaged [20]. Since the use of I-frames in compression is not as common as P-frames and B-frames, an alternative solution is to estimate the MVs of an I-frame at the decoder side by implementing block matching process between the decoded I-frame and the previously decoded frame.

There also exist blocks within a P-frame or B-frame which are compressed based on intra prediction. In this case we estimate the block's motion as the median of its neighbouring-block motion vectors. There is no provision for the intra prediction case in [12].

Multiple reference frames may be used in H.264 motion estimation for enhancing the compression performance. In this case, the estimated motion vectors do not represent the displacement of objects over two consecutive frames any more. To resolve this problem, we assume that the motion vector length is related to the reference image distance [21]. Therefore, to find the motion information between two consecutive frames, the H.264 estimated motion vectors should be rescaled proportionally, according to the distance between the used reference image and the last encoded image in the video sequence as suggested in [21].

For B-frames, forward or backward reference frames are used for predicting the blocks. To ensure the estimated motion vectors represent the displacement of objects over

consecutive frames, the motion vectors of each block should refer to the same region[4] in forward or backward reference frames. If that is not the case, the block's motion is estimated as the median of its neighbouring-block motion vectors.

3.3.2 Camera motion correction

A potential problem that arises in motion-based depth estimation algorithms is when both the objects and the camera are in motion, i.e., neither of them is stationary. In this case, an object that is estimated to have captured larger motion than others may not be closer to the camera, since the camera may have moved in the opposite direction from the object. To resolve this issue, the camera motion needs to be approximated and the motion information registered by the camera should be corrected accordingly.

In the camera panning case, the registered motion information for the stationary areas of the scene would be equivalent to the camera motion. These areas are often flagged as 'Skip Mode' by the H.264/AVC motion estimation process. The 'Skip Mode' is used for 16x16 blocks, where the motion characteristics of the block can be effectively predicted from the motion of its neighboring blocks, and the quantized transform coefficients of the block are all zeros. When a block is skipped, the transformed coefficients and the motion data are not transmitted, since the motion of the block is equivalent to the median of the motion vectors of the surrounding blocks. This median is known as the predicted motion vector (MV_p) [18].

As long as there is no camera motion, the predicted motion vector of a skipped block is zero. However, when camera panning is present, all the predicted motion vectors

[4] Image segmentation is required in such cases (see Section 3.3.3).

of the skipped blocks become equal to a unique non-zero value (which represents the camera motion). Since most of the skipped blocks over the entire frame are blocks that contain background areas, our proposed scheme uses the predicted motion vector (MV_p) with the maximum occurrence to estimate the value of the camera motion. In order to find this vector for each frame, we compute the histogram of the MV_ps of all the skipped blocks. The MV_p that corresponds to the maximum of the histogram is recognized as the camera (panning) motion [15]. The net motion of each object is extracted by subtracting the camera motion from the MVs of all blocks within the frame. This procedure cannot be accommodated in [12] since only 4x4 blocks are used and the size of the skip-mode blocks is 16x16. The following code summarizes the above-mentioned procedure:

for each frame:
1. find skipped-mode blocks with MV_p~=0.
2. calculate the histogram of the MV_ps of blocks found in 1.
3. assign the MV_p with maximum occurrence to camera panning motion.
4. subtract camera panning motion from all the MVs within the frame.

Besides panning, camera zoom-in/out can also cause depth ambiguity. To address this issue, we check the tendency of MVs in the four corners of the frame to detect zoom in/out [11]. Then the estimated MVs are scaled accordingly. Note that zoom-in/out may cause reverse depth or eye fatigue if not corrected in the depth estimation process.

3.3.3 Correction of false displacement estimates

The criterion used in video compression standards is to optimize the rate distortion and maximize the compression performance, i.e., to transmit the least number of bits. The motion vectors obtained by H.264/AVC coding are thus derived so as to minimize the compression rate and optimize rate distortion and not maximize the

accuracy of the estimated motion of the objects within a scene. Thus, the matching blocks determined by a motion vector do not necessarily relate to the same part of an object in the scene. In such a case the estimated motion vectors do not accurately convey the displacement of an object, i.e., they do not point to the corresponding left and right areas as defined by the disparity in a stereoscopic scenario (Figure 3.1).

To check if a motion vector points to the same object (or part of it) in two consecutive frames, our proposed scheme compares the motion of the block with that of its surrounding blocks. To this end, we use two predefined thresholds, Th_1 and Th_2. Threshold Th_1 is defined as the difference between the MV of the block and the median MV of the neighboring blocks (MV_m) of the same object. Our experiments have shown that if this difference is larger than 1, then either the MV is not the actual displacement or there is a moving edge within the block. Threshold Th_2, which is the measure of the variance of the residue block resulting from subtracting the present and previous frame, is then used to help us determine if the block includes the boundary pixels of a moving object. In the residual block, the edges of moving objects appear thicker and with higher density compared to static objects and the background (see Figure 3.4a). Camera motion compensation is taken into account when constructing the "residue frame". Note that the "residue frame" is obtained by direct subtraction of two consecutive frames. In the case of panning, since we know the global motion vector, simple shifting will compensate for camera motion. In the case of zooming, the frame zoomed out more (i.e., shows objects further away) is cropped and scaled to match the other frame.

We have found through performance evaluations that if the variance (i.e.,Th_2) is less than 1000, then there is a very high probability that there is no moving edge within

the block and therefore the MV needs to be replaced by the median MV the instant neighboring blocks. Otherwise (i.e., there is a moving edge within the block), the H.264 estimated MV is correct.

Figure 3.4 (a) Residue frame, and (b) a color-texture segmented frame of "Orbi" sequence.

Assume the MV of a certain block is presently being estimated. To find the adjacent blocks that belong to the same object as this block, we use an unsupervised segmentation algorithm called JSEG [16]. This algorithm consists of color quantization and spatial segmentation as two separate steps. As a result, each frame is segmented into different regions based on the color and texture of the region, without any presumption about the number of objects (see Figure 3.4b). For detailed information on this algorithm see [16].

The following code summarizes the above procedure for displacement correction:

for each frame:

1. calculate MVdiff=abs(MV-MV_p)for each block.
2. compute residue frame as:

   resFrame= abs($luma_{current\ frame}$–$luma_{previous\ frame}$)
3. calculate the variance of each block within residue frame (resVAR).
4. implement JSEG algorithm to segment the frame based on color and texture and distinguish different object regions.

5. for the blocks which MVdiff>Th_1 and resVar<Th_2, new MVs are calculated as of median MV of their instant neighboring blocks belong to the same segmented object.

3.3.4 Displacement correction of object-border pixels

In this study, we use the absolute value of the horizontal component of the motion vectors (i.e., abs(MV_x)) for estimating the depth map. This is because, as explained earlier, the disparity, d ($=x_R-x_L$), which is the horizontal displacement between the two camera images, is related to the depth Z (as shown in Figure 3.1).

Since all the pixels within each matching block are assumed to have the same amount of motion, the abs(MV_x) is assigned to each pixel within the block. This assumption, however, is not valid for blocks that include both stationary background pixels and moving object pixels. For such blocks, a different procedure should be used, otherwise the resulting object borders in the constructed 3D video may appear blurred, and the small details or even entire objects may be removed.

A computationally expensive solution is to perform pixel-based motion vector estimation for the object-border pixels. We propose an alternative solution which detects the border blocks with non-zero motion vectors, then, classifies each pixel within each of these blocks as a background pixel or an object pixel. This process is based on the results of the JSEG algorithm (subsection 3.3.3). The estimated abs(MV_x) is assigned to the object pixels, while the median of the abs(MV_x) of the surrounding non-object pixels is assigned to the background pixels. The background pixels inside the block who have been assigned with updated abs(MV_x) might be utilized in the motion correction process. Note that if the block includes multiple segmented regions, the MV of the pixels within

each segmented region is calculated as the median of MVs of pixels within the neighboring blocks which belong to the similar segmented region.

JSEG algorithm is robust but as it can be observed from Figure 3.4b, over-segmentation may occur because of the varying illumination shades [16]. For this reason, pixel classification of border blocks is further verified by calculating the average luma intensity of the corresponding block in the "residue frame", and then comparing it with the luma intensity of each pixel within the block. The luma intensity of the background pixels in the "residue frame' should be less than the calculated average. This is because the intensity of the still-background pixels in the "residue frame" is close to zero, while the intensity level of moving region edges is high.

In our method, to accurately estimate the motion vectors of the background pixels located in the central part of the blocks, we start the motion correction procedure with those background pixels that have higher number of neighbouring pixels with correct motion-values. This ensures that the median value is mainly based on correct motion values. After each iteration, the abs(MV_x) values are updated and this process continues until all motion values are corrected. After pixel-based motion vector estimation, the absolute horizontal value of the motion vector of each pixel is used to approximate its initial depth value. Figure 3.5 shows the initial depth map after the above-mentioned procedure.

The following code summarizes object-border correction process:

for each object:

1. find blocks which $MV_x \sim = 0$.

2. label pixels within the blocks found in 1 as background-pixel or object-pixel based on the object-segmentation results by JSEG.

3. find corresponding blocks of the ones found in 1 in the resFrame.

4. compute average luma intensity of blocks found in 3 (resAVR)

5. compare the luma intensity background pixels within each block found in 3 (in the resFrame) with resAVR; if it was higher than that, change the pixel's label to object-pixel.

6. label the corresponding pixels within the blocks found in 1 (in the current frame) according to the ones in 5.

7. for background pixels within the blocks found in 1 recalculate motion as of median of abs(MV_x) of instant-neighboring background-pixels by iteration.

8. for the object pixels assign the estimated abs(MV_x) of the block.

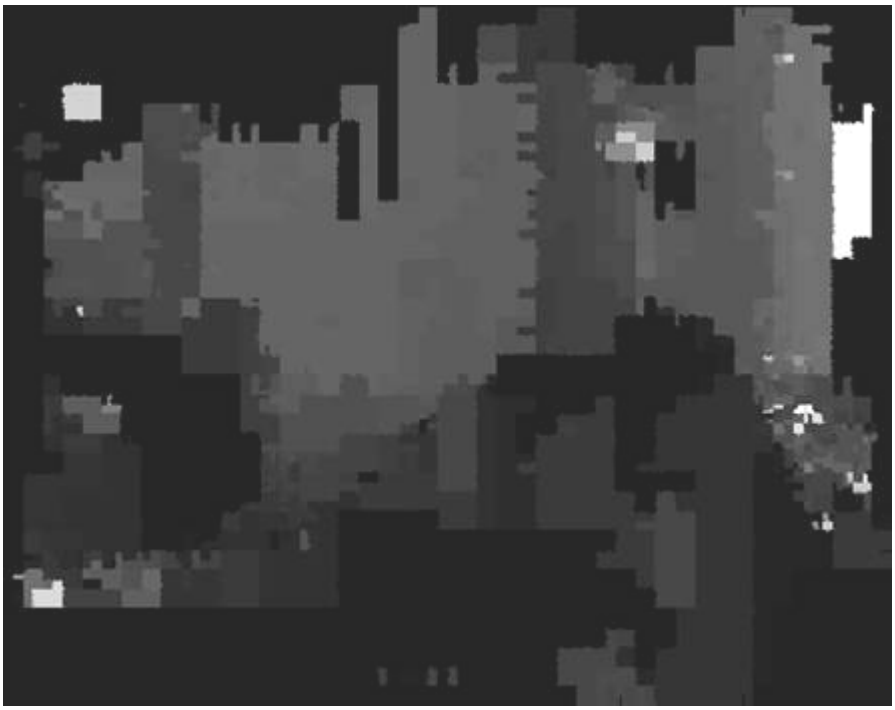

Figure 3.5 Initial depth map after correcting the displacement of object-border pixels.

3.3.5 Displacement correction of object-body pixels

For the proposed method we assume that all the pixels of a segmented object have one depth value. Since the depth of each pixel is related to its abs(MV_x), a common abs(MV_x) should be assigned to all the pixels within each object region. A simple solution is to calculate the average abs(MV_x) of all the pixels within each segmented

object and assign that to all the pixels. The problem with this approach is that the H.264 motion estimation process assigns zero-value motion vector or skip-mode flag to the flat areas of segmented objects (usually middle part). Thus, averaging all the abs(MV_x) will not give an accurate estimate of the motion of the segmented object. Among the pixels of a segmented object, the border/edge pixels are the ones whose motion better represents the motion of the entire object. Thus in our proposed scheme, the average of the abs(MV_x) value of border pixels is assigned to all the pixels of the segmented object and is used in estimating the object's depth value. This may cause some detailed depth information of the central part of objects is lost, but it will not hamper the quality of the final 3D video, since the depth information of the object boundaries is preserved [5, 6].

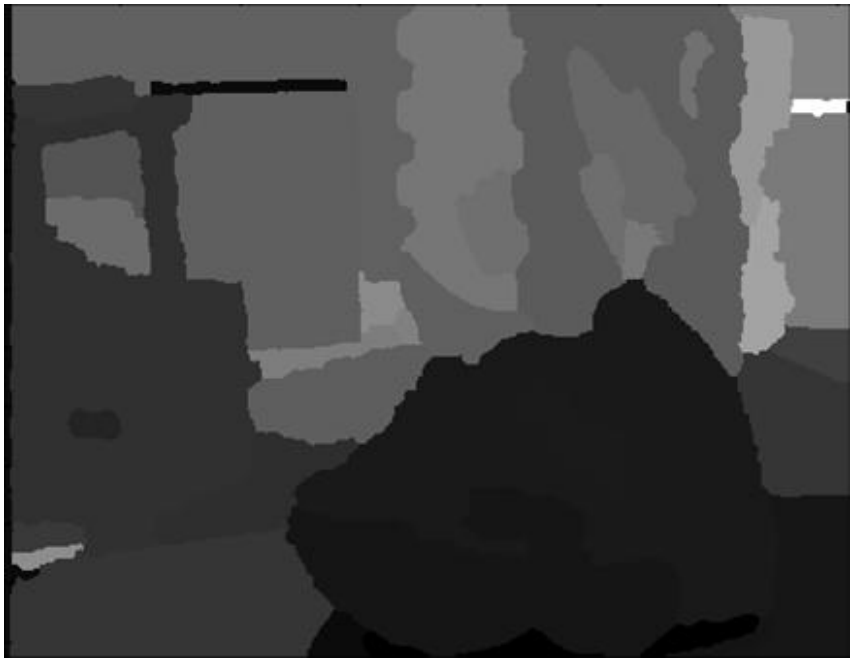

Figure 3.6 Motion information after correcting the displacement of object-body pixels.

Figure 3.6 shows the resulting motion information after correcting the displacement using the above-described process. In our study, the thickness of the object-

boundary was set to four pixels at most, with the motion of a segmented object estimated as the average of the motion values of these pixels.

3.3.6 Perceptual depth enhancement

The visual depth perception on 3D display systems is limited to the objects within a certain range from the camera. This means that only the objects within a certain distance from camera can be seen on a 3D display system, while distant objects have no depth perception. Considering this limitation, to enhance visual depth perception of videos to be watched on 3D display, we apply a non-linear scaling model to the abs(MV_x) values, which increases the disparity of closer objects and decreases that of distant objects. Since there is a non-linear relation between the visual depth perception (disparity) and the distance of an object as shown in Figure 3.2, the proposed scaling factors are defined such that the further the object is, the smaller the scaling factor. This will increase the contrast among depth values and enhance the visual depth perception.

Our model assumes that there are N uniformly spaced depth layers within a scene, i.e., within [Z_{near} and Z_{far}]. A set of scaling factors is defined as:

$$S(i) = \frac{i}{N-1}(1 - \frac{Z_{far}}{Z_{near}}) + \frac{Z_{far}}{Z_{near}} \qquad 0 \le i \le N\text{-}1 \qquad (3\text{-}2)$$

where S is the scaling factor and i ranges from layer 0, which corresponds to Z_{near} (here i=0, $S(0) = Z_{far} / Z_{near}$) to layer N-1 which corresponds to Z_{far} (i=N-1, $S(N\text{-}1) = 1$). To generate the enhanced depth map, the $abs(MV_x)$ values are sorted and categorized to N uniformly spaced layers. If $abs(MV_x)$ belongs to the i^{th} category, its value is scaled as follows:

$$D = abs(MVx)S(i) \tag{3-3}$$

where D is the enhanced disparity value and $S(i)$ is the scaling factor at the i^{th} depth layer. Figure 3.7 shows the estimated depth information after perceptual depth enhancement.

Figure 3.7 Estimated depth map after perceptual depth enhancement.

Using the approximated depth map and the 2D video sequence, the stereoscopic pair images are rendered via the depth-image-based rendering algorithm proposed in [2]. This algorithm includes a depth map smoothing process (using asymmetric Gaussian filter) to resolve the occlusion problem of depth image-based rendering. In our implementation, only the right-eye stream is rendered (based on the estimated depth map and the 2D video sequence), and the original 2D video is used as the left-eye stream [5, 6].

3.4 Performance Evaluation and Discussion

The performance of our proposed depth map estimation method is tested using the 2D video sequences “Interview”, “Orbi”, “Breakdancers” and “Ballet”. “Interview” and

"Orbi" have been captured by a 3D-depth range camera (Zcam) [22], which measures the depth map of the scene while recording the 2D video. Thus, the true depth-measures of the scene are available in the form of sequences which consist of luma information only. The 2D streams of "Interview" and "Orbi" are 10-seconds and 5-seconds long respectively, with 720×576 pixels resolution and 4:2:2 YUV format [23]. "Breakdancers" and "Ballet" are two multi-view video test sequences (8 views), with 1024x768 pixels resolution, which provided by Microsoft Research (MSR) group for research on multiview video coding (MVC) standard [24]. MSR also has provided the approximated depth map of the scene using stereo matching. For our experiment, the forth view video sequence of each test set and its corresponding depth map was used.

In our experiments, the motion between two consecutive frames is estimated using the H.264 encoder (JM 12.2 version). We assume that the broadcasted video is of acceptable visual quality. For this reason, in our experiments the quantization parameter was set to QP=30, which yields PSNR (Peak signal-to-noise ratio) values above 37 dB for the tested sequences. The GOP (Group of Picture) size was set to 25 frames, and the frame structure was IBBP with 5 reference frames. We compare our method with the one presented in [12]. Since the recorded depth per each pixel is an integer number between 0 and 255, (where 255 represent the shortest distance from the camera), we assume there are 256 depth layers for the perceptual depth enhancement step. For the same reason, the estimated depth maps of both methods are normalized as integer numbers between 0 and 255.

Figure 3.8 2D video sequence (a, b, c, d), recorder depth map (e, f), depth map estimated by stereo matching (g, h) estimated depth map by [12] (I, j, k, l), and estimated depth map by our approach (m, n, o, p).

Figure 3.8 shows a snap-shot of the original 2D stream, the original depth map, and the estimated depth maps generated by [12] and our approach. The experimental results have been posted online for further review [25]. We observe that our approach yields more realistic depth estimates compared to [12]. As it is illustrated in Figure 3.8, unlike [12], our method can approximate the depth information of an entire object even if only partial motion information of the object is available. This is due to our object-body displacement correction procedure. The success of this procedure, however, depends on the accuracy of the adopted color-texture segmentation algorithm. As it has been reported, JSEG is a robust segmentation algorithm but has some limitations (over

segmentation where variation of illumination shades exists) [16]. Therefore, in order to further improve our depth estimation technique in the future, more research is needed on enhancing the segmentation procedure.

Comparing the results in Figure 3.8 shows that both techniques like other motion-based depth estimation approaches fail to estimate the depth maps of static objects (e.g., the table in Figure 3.8b). According to the human visual system, which integrates different depth cues to perceive the depth, one can expect the integration of depth cues to provide more sufficient means for depth map estimation techniques [5, 6]. Improvement on retrieving depth information of static objects may require integration of our approach with other depth cues, such as sharpness, and it is recommended path for future research.

The visual quality of the resulting 3D video streams using our method and the one presented in [12] are subjectively tested against the original depth map (for Orbi and Interview) and the one acquired via stereo matching by MSR (for Breakdancers and Ballet), based on the ITU-R Recommendation BT.500-11 [26]. Eighteen people graded the videos from 1 to 100 in terms of 3D visual perception and visual picture quality in two separate experiment sets (with a short rest period between sessions). The evaluation is performed using the SeeReal, C_n 3D display. For the 3D visual perception part of the experiment, the viewers were asked to score the videos based on the depth or volume that objects appeared to have. For the visual picture quality part, the viewers were asked to rank the videos based on picture quality, which could be affected by visual noise, blur, or various other distortions and picture instabilities (which may cause visual discomfort for viewers). Higher picture quality corresponded to higher scores.

Figure 3.9 illustrates the average scores of our subjective test. The original stereoscopic video had the highest scores in terms of visual picture quality and our method yielded the highest scores in terms of 3D visual perception. These tests show that the approximated depth map obtained by our method provides the best 3D visual perception and the visual picture quality of the results by our technique is higher than those of the existing method.

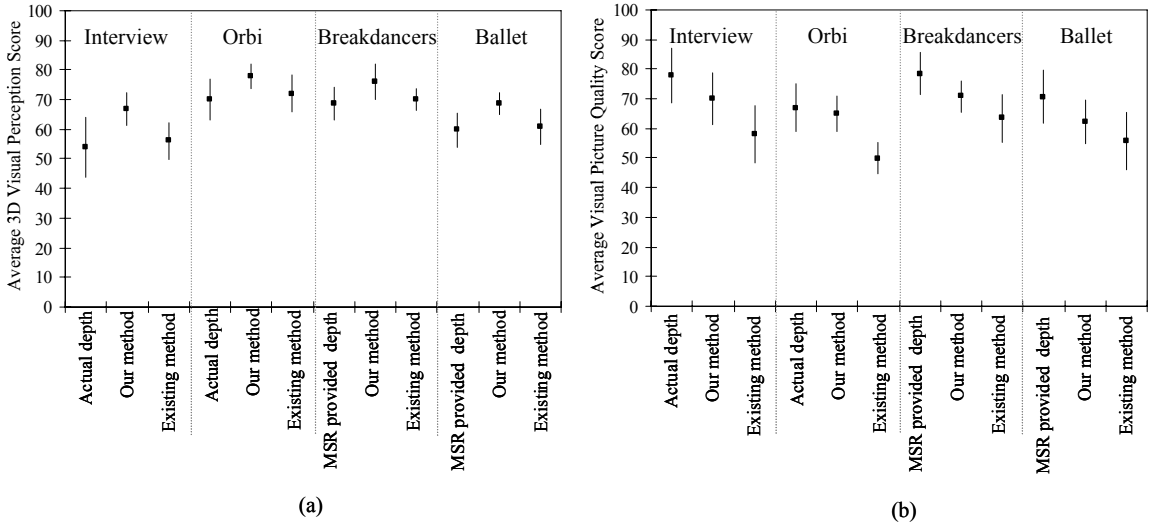

Figure 3.9 Average subjective test scores of 3D visual perception (a) and picture quality (b) for test streams. The error bars denote the 95% confidence intervals.

Since our technique and the one suggested by [12] are both capable of approximating the depth information only for areas with moving objects, watching the resultant stereoscopic video streams may create visual discomfort for viewers. However the use of the object-body displacement correction procedure in our method has been successful in reducing this effect. This procedure reduces the artifacts and results in a smooth depth map. Without it, the rendered stereoscopic image would have the binocular parallax effect only for parts of moving objects (for which the H.264-based motion

information is non-zero). In this case, some part of an object is perceived in 3D format and the rest in 2D, something that would increase the viewer's visual discomfort.

The visual 3D perception improvement obtained by our method is due to two factors: the perceptual depth enhancement step, and the prominence of the depth for moving-objects. Figure 3.10 demonstrates this effect clearly. As it can be observed, the fingers of the moving hand in image (b), rendered using the depth estimated by our technique, are longer than the ones based on the real depth map (a) and the estimated depth map by [12] (c). This effect increases the 3D perception when the rendered videos are watched on a 3D display.

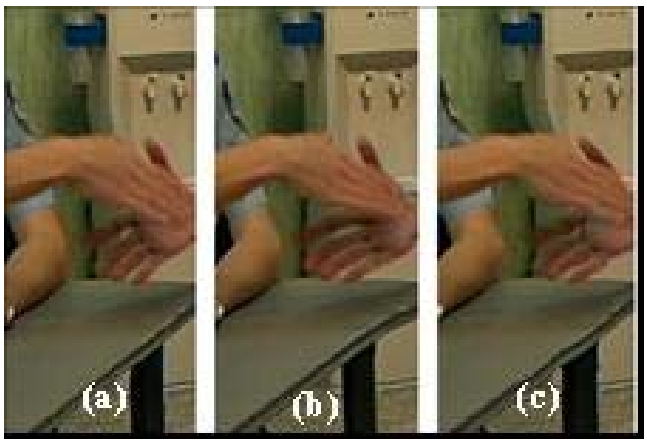

Figure 3.10 Rendered right image based on real depth map (a), estimated depth map by our approach (b), estimated depth map by [12] (c).

For the quantitative analysis, we chose to compare the quality of the stereoscopic videos synthesized using our technique with those of the technique proposed in [12] and the stereoscopic videos rendered from the actual (recorded) depth map. Note, since the ground truth depth maps of Breakdancers and Ballet is not available, the results for these two streams did not go under quantitative analysis.

Figure 3.11 illustrates the five different PSNR comparisons that we chose for our analysis. In one scenario we compare the right view generated by our method and the one by [12] with the right view rendered based on recorded depth map (b and d in Figure 3.11). These comparisons show how close the average quality of the estimated 3D views is to the actual ones. In this case, the higher PSNR values indicate better visual quality. Table 3.1 shows the average PSNR values obtained for this case. We observe that our method outperforms the proposed method in [12] by 1.81 dB to 1.98 dB.

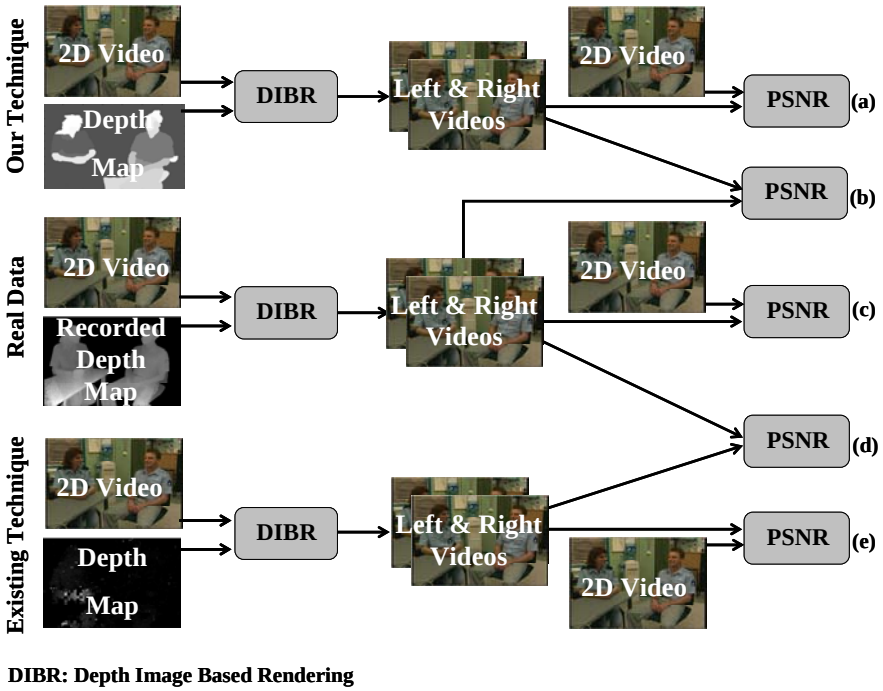

Figure 3.11 Quantitative analysis of the results.

Table 3.1 Average PSNR comparison case b and d in Figure 3.11.

Average PSNR (dB)	Interview	Orbi
3D views based on our method vs actual 3D views	36.45	31.91
3D views based on existing method vs actual 3D views	34.47	30.1

In addition to the above, we also compare the generated right views with the actual 2D video stream (a, c and e in Figure 3.11). These comparisons show how effectively the two different techniques generate depth perception. In this case, since there is no depth present in the 2D video stream, large PSNR values indicate failure in adding significant depth perception to the stream. Table 3.2 shows the average PSNR values obtained for this case. As expected, we observe that the actual 3D views have the least similarity with the 2D video (no depth perception). More importantly, our method yields a PSNR value that is very similar to the actual 3D views while the PSNR value obtained by the [12] is higher than the original recorded depth. This conveys the fact that the depth map estimated by [12] creates the least 3D perception.

Table 3.2 Average PSNR comparison case a, c and e in Figure 3.11.

Average PSNR (dB)	Interview	Orbi
Right view rendered based on the actual depth map vs actual 2D view	32.27	27.85
Right view rendered based on our estimated depth map vs actual 2D view	32.38	27.89
Right view rendered based on the estimated depth map the by existing method vs actual 2D view	36.99	33.34

The percentages of the badly matched pixels for the estimated depth yielded by our scheme and by [12] were computed as:

$$B = \frac{1}{N}\sum_{(x,y)}(|\ D(x,y) - D_r(x,y)\ | > Th) \tag{3-4}$$

where N is the number of all pixels within the depth map, D is the estimated depth map, D_r is the recorded depth map and *Th* is the error tolerance. In our experiment we use *Th*=1 [27]. The results show that for our method the percentage of the correctly matched pixels is 53% (Interview) and 48% (Orbi). For [12], the percentage of correctly matched

pixels is 34% (Interview) and 27% (Orbi). The comparison confirms that our method outperforms the existing method by 19% to 21%.

3.5 Conclusion

We present a new and efficient method that approximates the depth map of a 2D video sequence using H.264/AVC estimated motion information. This method exploits the existing relationship between the motion of objects and their distance from the camera to estimate the depth map of the scene. Our proposed method revises the motion information based on the characteristics of the 3D visual perception. In this study, the 2D horizontal motion is approximated as the displacement existing between the right and left images when the scene is captured by a stereoscopic camera. For cases involving a moving camera and for possible problems regarding the displacement of object borders and false displacement estimates, our proposed method provides appropriate solutions. To improve the quality and smoothness of the estimated depth, our algorithm utilizes color-texture segmentation. Our proposed approach can be implemented in real-time at the receiver-end, offering 3D experience without increasing transmission bandwidth requirements. Performance evaluations have shown that our approach outperforms the other existing H.264 motion-based depth map estimation technique by up to 1.98 dB PSNR, i.e., providing more realistic depth information of the scene.

The visual quality of our constructed 3D stream was also tested subjectively, with viewers watching the generated 3D streams on a stereoscopic display. The subjective tests showed that the 3D streams created based on our approach provided viewers with superior 3D experience. Moreover, in terms of visual quality, our approach outperforms the other existing H.264-based depth estimation method.

3.6 References

[1] O. Schreer, P. Kauff, T. Sikora, 3D Video communication: Algorithms, concepts and real-time systems in human centered communicatio*n*, John Wiley & Sons, Inc. 1st edition, 2005.

[2] L. Zhang, W.J. Tam, "Stereoscopic image generation based on depth images for 3D TV," In: IEEE Trans. Broadcasting , vol. 51, no.2, pp191-199, 2005.

[3] P. Harman, J. Flack, S. Fox, M. Dowley, "Rapid 2D to 3D Conversion," In: Proceedings of SPIE, vol. 4660, pp. 78–86 (2002).

[4] S.H. Lai, C.W. Fu, S. Chang, "A generalized depth estimation algorithm with a single image," PAMI, Vol. 14(4), pp. 405-411, 1992.

[5] W. J. Tam, A. Soung Yee, J. Ferreira, S. Tariq, F. Speranza, "Stereoscopic image rendering based on depth maps created from blur and edge information," In: Proceedings of Stereoscopic Displays and Applications XII, Vol. 5664, pp.104-115, 2005.

[6] W. J. Tam, F. Speranza, L. Zhang, R. Renaud, J. Chan, and C. Vazquez, "Depth image based rendering for multiview stereoscopic displays: Role of information at object boundaries," Three-Dimensional TV, Video, and Display IV, Vol. 6016, pp. 75-85, 2005.

[7] Y.L. Chang, C.Y. Fang, L.F. Ding, S.Y. Chen, and L.G. Chen, "Depth Map Generation for 2D-to-3D Conversion by Short-Term Motion Assisted Color Segmentation," IEEE International Conference on Multimedia and Expo, 2007.

[8] C.T. Lin, C.L. Chin, K.W. Fan, C.Y. Lin, "A novel architecture for converting single 2D image into 3D effect image," 9th International Workshop on Cellular Neural Networks and Their Applications, pp.52-55,May 2005.

[9] G. Cheung, A. Ortega, and T. Sakamoto, "Fast H.264 Mode Selection Using Depth Information for Distributed Game Viewing," IS&T/SPIE Visual Communications and Image Processing (VCIP), San Jose, CA, 2008.

[10] T. Okino, H. Murata, K. Taima, T. Iinuma, K. Oketani, "New television with 2D/3D image conversion technologies," Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems III, vol. 2653, pp. 96-103, 1996.

[11] D. Kim, D. Min, K. Sohn, "Stereoscopic video generation method using motion analysis," In: Proceedings of 3DTV Conf. pp. 1-4, 2007.

[12] I. Ideses, L.P. Yaroslavsky, and B. Fishbain, "Real-time 2D to 3D video conversion," Journal of Real-Time Image Processing, Vol. 2, no. 1, pp. 3-9, 2007.

[13] C. Pulfrich, „Die Stereoskopie im Dienste der isochromen und heterochromen Photometrie," Naturwissenschaften, vol. 10, no.34, pp 735–743, 1922.

[14] DC. Burr, J. Ross, "How does binocular delay give information about depth?," *Vision Research*, vol. 19, pp. 523–532, 1979.

[15] M.T. Pourazad, P. Nasiopoulos, and R.K. Ward, "Converting H.264-derived Motion Information into Depth Map," 15th International MultiMedia Modeling Conference (MMM2009), France, pp. 108-118, January 2009.

[16] Y. Deng, and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," IEEE Trans. Pattern Anal. Machine Intell., vol. 23, no. 8, pp. 800-810, 2001.

[17] D. Scharstein,:View Synthesis Using Stereo Vision. In: Lecture Notes in Computer Science, Springer, 1999.

[18] I.E.G. Richardson,: H.264 and MPEG-4 Video Compression: Video Coding for Next generation Multimedia, John Wiley & Sons, Inc., England, 2003.

[19] A. Vetro, P. Pandit, H. Kimata, and A. Smolic, "Joint Multiview Video Model (JMVM) 5.0," ISO/IEC JTC1/SC29/WG11/N9214, Lausanne, Switzerland, Jul. 2007.

[20] J. Kim, Y. Kim, J. Park, J. Kang, B. Lee, "Stereoscopic conversion of two-dimensional movie encoded in MPEG-2," Proceedings of SPIE, vol. 6311, pp. 631105.1-631105.8, 2006.

[21] S., Moiron, S. Faria, P. Assuncao, V. Silva, A. Navarro, "H.264/AVC to MPEG-2 Video Transcoding Architecture," In: Proceeding of Conference on Telecommunications- ConfTele, May 2007, pp. 449–452, 2007.

[22] C. Fehn, "A 3D-TV system based on video plus depth information," Signals, Systems and Computers, vol. 2, pp. 1529–33, 2003.

[23] C. Fehn, K. Schüür, I. Feldmann, P. Kauff, and A. Smolic, "Proposed Experimental Conditions for EE4 in MPEG 3DAV," ISO/IEC JTC1/SC29/WG11, MPEG02/M9016, Shanghai, China, October 2002.

[24] D. Kim, N. Hur, and S. I. Lee, "Anchor bitstreams for Call for Proposals on multi-view video coding (Microsoft sequences)," ISO/IEC JTC1/SC29/WG11, MPEG2004/M12280, Pozan, July 2005.

[25] http://www.ece.ubc.ca/~pourazad/eurasip09

[26] Methodology for the subjective assessment of the quality of television pictures, ITU-R Recommendation BT.500-11.

[27] D. Scharstain, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," In: International Journal of Computer Vision, vol. 47, pp. 7-42, 2004.

CHAPTER 4: UNSYNCHRONIZED ZOOM CORRECTION IN 3D VIDEO[5]

4.1 Introduction

The majority of 3D content is produced using a dual-camera configuration, which generates a stereo pair where the left-eye and the right-eye views are separately recorded from slightly different perspectives. Creating visually pleasing stereoscopic video is a very tricky task and requires that both the director and the camera operators are extremely knowledgeable of the challenges of 3D capturing and the limitations of the 3D displaying devices.

In fact, defining the Quality of Experience in 3D is a new and challenging task, and has attracted the attention of many different standardization bodies such as MPEG, ITU and SMPTE. The challenges arise from the differences between the 3D capturing devices and the way human visual system perceives 3D, and the limitations of the 3D displaying mechanisms. While we may not yet have a clear understanding of how different capturing and displaying parameters affect 3D quality, we definitely know that unsychronized zooming of dual cameras will degrade the perceived 3D video quality. It is well known that precise synchronization of the optical zooming of two identical cameras is very difficult. In practice, zoom progression is not a linear function of the magnification factor and this function is inconsistent between optical lenses. If dual cameras are set to the same zoom progression course, the final magnification will not be

[5] A version of this chapter will be submitted for publication. Pourazad, M.T., Doutre, C., Nasiopoulos, P., Tourapis, A., and Ward, R.K. (2010) Unsynchronized zoom correction in 3D video.

identical (optics discrepancies) [1]. The effect of having different magnification factors on each camera in a stereo set-up is that the objects will have different size in the left and right views. This will introduce vertical parallax which causes eye-strain and interferes with the fusion of the two images (convergence artifacts) [2]. The other issue in synchronized zooming of dual cameras is related to the tendency of the cameras' optical axis to roam around during zooming, a process that is inconsistent from one lens to another. This also causes vertical parallax and convergence artifacts.

One possible solution to the zooming synchronization problem in a stereoscopic camera setup is to use high-end computerized motorized 3D rigs [1]. These rigs use zoom look-up tables to compensate for the optics discrepancies and lenses' roaming tendency compensation and they utilize the motion control of at least one of the cameras. These 3D rigs are not commercially available for sale to end users. This set-up is custom made for professional film making companies [1].

There are less perfect alternative solutions for ordinary amateur videographers willing to capture quality stereoscopic videos with synchronized zooming effects. They can either use zoom synchronizing electronic devices at the time of capturing or run a "correction pass" during postproduction to match the stereo images' magnification.

Electronic zoom synchronization is possible through the communication port of video cameras, known as the LANC (Local Application Control Bus System) protocol. Sony developed the LANC protocol in the 1980s, to allow sending commands to the camera, like controlling the shutter and the zoom [3]. What makes LANC suitable for 3D is its ability to switch the cameras on and off and to get the video clock signal. Recently, some dedicated devices for synchronizing cameras have been introduced to the market,

like the “LANC Shepherd” [4] and the “ste-fra® LANC” [5]. These devices switch on both cameras at once by applying ground voltage (0V) to the LANC signal pin for a certain period of time (about 140ms). In addition, they check the video clocks and display the actual synchronization delay. Within a couple of tries, both cameras will “boot” relatively synchronized (very small initial time disparity) [1]. Due to several reasons, the initial time disparity is usually larger than zero. However, since internal oscillators in both devices are running at slightly different frequencies, the time disparity changes over time and soon becomes unacceptable. Thus, after certain time of shooting, the cameras should reboot for synchronization. Another device, the 3D LANC Master, controls the time drift and keeps the camera in synch for hours [6, 7]. This device is not commercially available [7].

The unsynchronized zooming problem can also be corrected through post-processing algorithms. One solution proposed in [8] is to first determine the stereo cameras calibration parameters for different zoom settings. At the time of unsynchronized zooming, correct the camera parameters from the nearest calibrated zoom positions and use that knowledge to rectify the stereo images accordingly. However, in this case, information about the camera parameters is required. The other post-proceesing approach involves digital cropping and scaling of one of the views in order to make it match the other. Achieving that requires accurate estimation of the relative amount of zoom between the two videos. Several methods have been proposed for estimating zooming in monoscopic (single-view) video [9-11]. All of these involve relating the optical flow (i.e., the motion field) to camera parameters, such as zoom (focal length), translation and rotation. A common problem with these methods is separating the optical flow, caused

by changing camera parameters, from the optical flow caused by object motion. Determining which parts of the motion field are affected by object motion is an ill-posed problem, and it affects the accuracy of zoom estimation methods. To correct unsynchronized zoom in stereo video, one needs to estimate the zoom ratio between the two views. There is no previous work that has addressed this problem.

In this paper, first, we present a subjective study of the impact of zoom mismatch on the perceived quality of 3D video. The results show that unsynchronized zoom severely degrades 3D video quality. Next, we present a method for correcting zoom mismatch by applying digital cropping and scaling to one of the views. In the case of zooming in, the view that is originally zoomed in less is scaled to match the one that is zoomed in more, and in the case of zooming out, the view originally zoomed out more is scaled to match the one which is originally zoomed out less. To avoid the effect of object motion on zoom estimation, we use only vertical coordinates for estimating the zoom ratio between the views. This takes advantage of the constraint that there should be no vertical parallax between images captured with parallel cameras, which holds regardless of object motion.

The rest of this paper is organized as follows. A subjective study showing the impact of zoom mismatch in stereo videos is presented in Section 4.2. Our proposed method for correcting unsynchronized zoom is described in Section 4.3. Experimental results are presented and discussed in Section 4.4. Finally, conclusions are given in Section 4.5.

4.2 Impact of Zoom Mismatch on Subjective 3D Quality

Consider the stereo geometry of two parallel cameras shown in Figure 4.1. A point with world coordinates (X,Y,Z) is projected onto the left and right image planes at image coordinates $(x_{L,O},y_{L,O})$ and $(x_{R,O},y_{R,O})$. The subscripts L and R indicate the left and right images respectively, and the O subscript indicates that the coordinates are measured relative to the optical center of the camera. The cameras are separated by a baseline distance B and have focal lengths f_L and f_R. In a real camera the focal length is a property of the optical system (a higher focal length meaning more optical magnification), but even synthetic images are usually rendered with a virtual focal length in projecting a 3D model on to a virtual image plane.

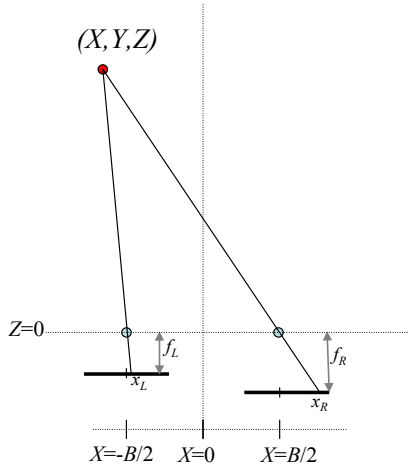

Figure 4.1 Stereo geometry with parallel cameras

Using similar triangles, simple equations for the image coordinates can be found as follows:

$$x_{L,O} = f_L \frac{X + B/2}{Z} \qquad x_{R,O} = f_R \frac{X - B/2}{Z}$$
$$y_{L,O} = f_L \frac{Y}{Z} \qquad y_{R,O} = f_R \frac{Y}{Z} \tag{4-1}$$

Ideally, the focal lengths of both cameras should be the same, i.e., $f_L = f_R$. In that case, $y_{L,O} = y_{R,O}$, so the images will have no vertical parallax, which is a very important requirement for 3D video. Vertical parallax in stereo images causes eyestrain, and if it is too large the human visual system will not be able to fuse the images at all, resulting in a loss or distortion of the 3D effect [2]. If the two initially synchronized cameras are zoomed by a different amount, it means that their focal lengths will be different. Consequently, there will be vertical parallax between the images, and objects will have a different size in each image; both results are highly undesirable in 3D video.

Figure 4.2 Stereo test images.

To evaluate the effect this unsynchronization has on viewers, we performed a subjective experiment using zoom-mismatched stereo video. In the test videos, the two views were zoomed in or zoomed out linearly and with the left view always having a larger scaling factor than the left (the difference was a constant value). To prepare an appropriate data set, we captured the stereo video streams "Emily", "Friends", "Arch", and "ICICS" using two full HD parallel cameras with baseline distance of 9cm. The captured stereo images were rectified to ensure right and left views are aligned. Figure 4.2 shows the snapshot of the test images. To synthesize left-view streams with zoom-

in/out effect, the left-eye image was digitally scaled (about its center) with the scaling factor being increased from 1X up to 1.8X of its original size. In each frame the scaling factor was increased 0.01X from that of the previous frame. Then the videos were zoomed back out to the original size. The right-view had the same zoom pattern as the left view, only with the scaling factor higher by a constant amount so that unsynchronized zooming is simulated.

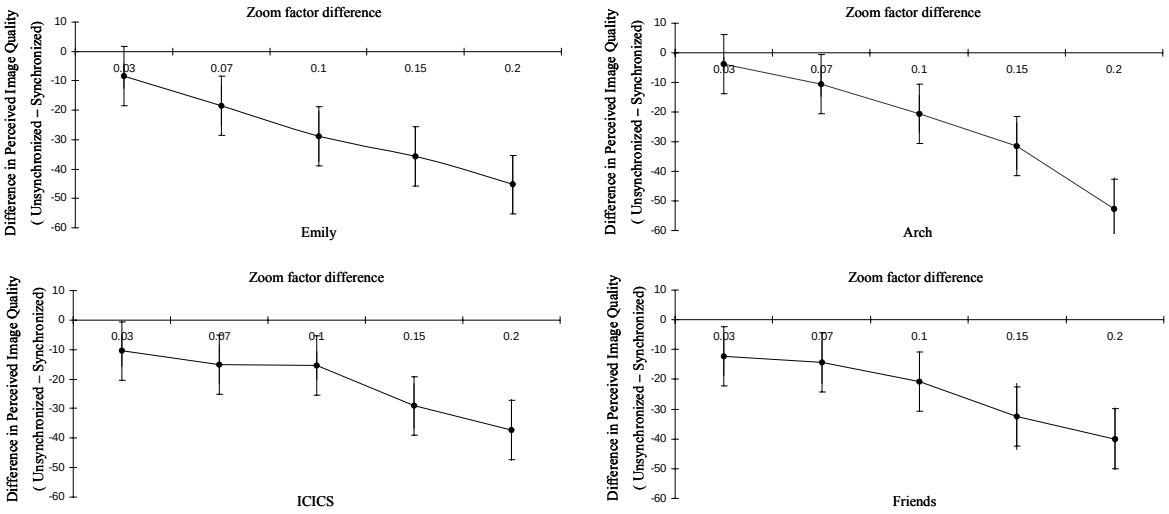

Figure 4.3 Subjective results: Rating is expressed as a difference between ratings for the unsynchronized and synchronized zoomed stereo videos. The error bars denote the 95% confidence intervals.

The test videos were shown to fifteen subjects on a 22” widescreen monitor with the 1680x1050 resolution and refreshing frequency of 120Hz. Most of the subjects had not participated in stereoscopic experiments before, and all viewers were naive to the underlying purpose of the experiment. In our experiment, viewers rated the test stereoscopic video streams using the double-stimulus continuous-quality scale method suggested by the ITU-R Recommendation 500 [12], in which viewers wore NVIDIA GeForce 3D vision (active) glasses and graded two versions of the same video stream (synchronized and unsynchronized zooming effect) announced as "A" and "B", from

"Bad" to "Excellent". The subjects were not informed which one of the video streams had the unsynchronized zooming effect. A number of zoom factor differences were tested, ranging from 0.03 to 0.2 (i.e., 3% to 20% size difference). For analysis, the ratings were digitized to the range between 0 and 100 units. Figure 4.3 shows the results of the subjective test. We observe that the quality of the perceived 3D image degrades dramatically when the difference between the zooming factor of right and left view streams increases. These results show that zoom mismatch causes a large drop in perceived 3D quality, and should be corrected.

4.3 Proposed Zoom Correction Method

The problem of unsynchronized zooming in stereo video can be fixed by applying cropping and scaling to one of the views. Referring to the camera model of Figure 4.1, the view with a shorter focal length has to be scaled to match the view with a longer focal length. If the cameras are zooming in, the view that has been zoomed less will have a lower focal length and should be scaled. If the cameras are zooming out, the view that has been zoomed out more will have a shorter focal length and should be scaled. For each frame in the videos, we estimate the amount of scaling that needs to be applied based on the y coordinates of matching points found between the two views.

To simplify notation, let us assume the right video has higher focal length, i.e. f_R is greater then f_L. According to equation (4-1), the left image could simply be scaled by a factor f_R/f_L, and the corresponding coordinates would be exactly the same as if the left image was captured with the same focal length as the right image. However, trying to implement this in practice is challenging. The image would have to be scaled about its optical center, which in general is not the same as the geometric center of the image [13].

The optical center of a camera changes as the camera zooms, so it is difficult to find the precise optical center of an image for every frame during zooming [13].

Instead of trying to directly estimate the optical center of each image and then apply scaling about that, we work in a coordinate system with the top left corner of the image defined as the origin. In this system, the optical center of the image is defined as (u,v) relative to the top left corner of the image. The coordinates of a point relative to the corner, which we will denote (x_L, y_L), are found simply by adding (u,v) to the coordinates in (4-1) that are relative to the optical center.

$$\begin{aligned} x_L &= x_{L,O} + u_L \qquad x_R = x_{R,O} + u_R \\ y_L &= y_{L,O} + v_L \qquad y_R = y_{R,O} + v_R \end{aligned} \tag{4-2}$$

Combining (4-2) and (4-1), we can find an expression that relates the y coordinates of the left and right images, with all the coordinates expressed relative to the image corners:

$$y_R = \frac{f_R}{f_L} y_L + v_R - \frac{f_R}{f_L} v_L \tag{4-3}$$

Equation (4-3) shows that we can apply a simple linear transform of the form:

$$y_L^{'} = s y_L + t_y \tag{4-4}$$

that will make the y coordinates of the left image match those of the right image (i.e., $y_L^{'} = y_R$), where s is a scaling factor $s = f_R / f_L$ and t_y is the amount of vertical translation $t_y = v_R - f_R v_L / f_L$. Estimating the parameters s and t_y is sufficient for scaling one image so that the images will have no vertical parallax. Therefore, we do not need to explicitly calculate the focal lengths or optical centers.

In order to estimate the parameters s and t_y, we find a number of matching points between the left and right images. There are many methods for finding matching points between images; we choose the Scale Invariant Feature Transform (SIFT) [14], as it is one of the most popular and reliable matching methods. Using SIFT feature matching provides a number of points (x_L,y_L) and (x_R,y_R) that match between the left and right images. A matrix equation relating these matching points to the scaling and translation parameters can be written as:

$$\begin{bmatrix} y_R \\ \vdots \end{bmatrix} = \begin{bmatrix} y_L & 1 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} s \\ t_y \end{bmatrix} \tag{4-5}$$

where each row in the matrices contains the data for one matching point. A standard linear least squares regression can be used to estimate s and t_y based on equation (4-5). With the parameters estimated, a simple scaling transform can be applied to the left image to make it match the right image:

$$\begin{bmatrix} x_L^{'} \\ y_L^{'} \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x_L \\ y_L \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{4-6}$$

Applying equation (4-6) simply requires re-sampling the image, which can easily be done with standard methods such as bilinear or bicubic interpolation. One additional parameter is required in (4-6), the translation along the x-axis. This parameter is much less important than t_y, as any choice of t_x will give images without vertical parallax. We choose t_x so that equal amounts of cropping will be applied to the left and right of the image during scaling. To achieve this, t_x is chosen as:

$$t_x = \frac{W}{2}(1-s) \tag{4-7}$$

where W is the width of the image.

The way we have calculated the parameter s, its value reflects the amount of scaling required if the left view were scaled to make it match the right view. Of course it is possible that instead the right view should be scaled to make it match the left view. We always want to scale up one of the images, because scaling down an image would result in there being missing data around the edges of the scaled image. Scaling up simply requires some image data be cropped from around the edges.

If the least squares regression gives a value of s less than one, it means we would have to scale down the left image to make it match the right one. In that case we actually want to scale up the right image. Simple rearranging of equation (4-4) shows that the appropriate scaling and translation values for modifying the right image are:

$$s' = \frac{1}{s} \qquad t_y' = -\frac{t_y}{s} \tag{4-8}$$

where s and t_y are the parameters as estimated with the least squares regression based on (4-5), and s' and t_y' are the modified values that should be used for scaling the right image.

Note that the value of s is related to which camera has been zoomed more, depending on whether the cameras are zooming in or zooming out. If the cameras are zooming in, a value of $s<1$ indicates the left view has been zoomed in more (and hence objects appear bigger in it), and if $s>1$ then the right camera has been zoomed in more. If the cameras are zooming out, a value of $s<1$ would indicate the right camera has been zoomed out more, and hence objects appear bigger in the left view. A value of $s>1$

indicates the left camera has been zoomed out more, and objects appear larger in the right view.

Our complete algorithm can be summarized as follows. SIFT is used to find matching points between the left and right images, and a linear least squares regression is performed based on equation (4-5). If the regression produces $s>1$, then the left image is scaled and cropped with equation (4-6). If $s<1$, then equation (4-6) is applied to the right image with the modified parameter values of (4-8). This entire process is repeated for each temporal frame in the stereo video.

4.4 Experimental Results

4.4.1 Objective results on digitally zoomed videos

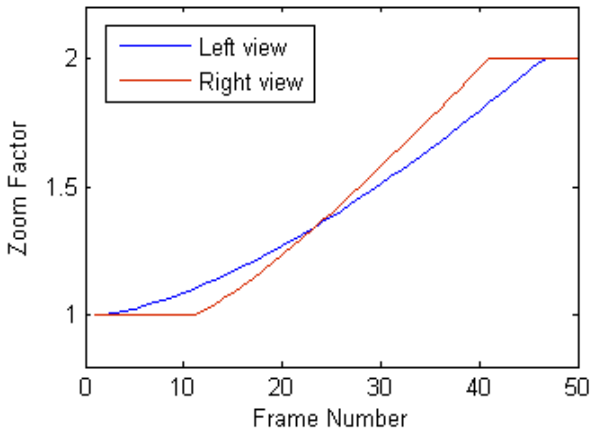

Figure 4.4 Digital zooming pattern applied to the left and right views for the objective tests.

In order to objectively measure the performance of the proposed method, we applied digital zooming to two standard test stereo videos, "soccer2" and "puppy". Both videos have resolution 720x480 pixels. We applied a different digital zooming pattern to the left and right view of each video, starting at original size and zooming in to a maximum zoom factor of 2X. The profile of the left and right zoom is illustrated in

Figure 4.4. Since we applied digital zooming to these videos, the ground truth values for s and t_y are known.

We corrected the videos with our proposed algorithm, and measured the mean absolute difference between the estimated parameters and the known ground truth parameters. We also report the maximum vertical parallax that is introduced in each frame due to the error in the estimated parameters, which is calculated as:

$$\Delta y_{\max} = \max_{y\in[1,H]} \left| s_{gt} y + t_{y,gt} - \left(s_{est} y + t_{y,est} \right) \right| \tag{4-9}$$

In (4-9) the parameters with the subscript '*gt*' are the ground truth parameters used in our experiment, and the parameters with an '*est*' subscript are those estimated with our method. The range of the argument 'y' in (4-9) is from 1 to H (the image height), but the maximum of (4-9) will always occur at one of the endpoints. Therefore only the y=1 and y=H cases need to be evaluated to find the maximum.

Table 4.1 Accuracy of estimated correction parameters.

Video	Mean Absolute Difference		Max Vertical Parallax, Δy_{max} (pixels)
	s	t_y	
soccer2	0.00038	0.127	0.236
puppy	0.00026	0.074	0.126

In Table 4.1, we report the absolute difference of the estimated correction parameters, as well as the maximum vertical parallax. All values are averaged over the 50 frames of the zoom-in. From Table 4.1, we can see the proposed method is very accurate at estimating the correction parameters. The maximum vertical parallax introduced is well below one pixel for both test videos, which is below the limit of what is noticeable by the human visual system.

4.4.2 Results on 3D video with unsynchronized optical zoom

In the previous section, we showed that our proposed method works well on test videos where zoom mismatch was introduced synthetically through digital zooming. In order to test our method on video with optical zoom, we captured a stereo video pair with hand-controlled optical zoom. In the video, both cameras start at 1X zoom and zoom-in to a factor of just over 2X, and then zoom back out to 1X, lasting about 150 frames. The zooming was controlled by hand with no attempt to synchronize the two views. In Figure 4.5a, the captured video shows clear zoom mismatch between the left and right views, and the 3D effect is lost during most of the zooming due to excessive size differences and vertical parallax. After correction with our method, the left and right views show no noticeable size differences or vertical disparity (Figure 4.5b), and 3D effect is perceived during the zooming.

Figure 4.5 Sample frames of the test video with unsynchronized optical zoom. (a) Captured left-right stereo pair (b) Corrected with proposed method.

4.5 Conclusions

In this paper we propose a method for correcting unsynchronized zoom in 3D videos. For each frame, a set of matching points is found between the left and right views with the SIFT algorithm. A least squares regression is performed on the y

coordinates of these matching points to determine which view needs to be scaled and to estimate the amount of scaling and translation needed to align the views. Experimental results show our method produces videos with negligible scale difference and vertical parallax.

4.6 References

[1] B. Mendiburu, “3D Movie Making: Stereoscopic Digital Cinema from Script to Screen,” Focal Press, April 22, 2009.

[2] A. Woods, T. Docherty, and R. Koch, “Image distortions in stereoscopic video systems,” Proc. SPIE, vol. 1915, pp. 36–48, 1993.

[3] M. Boehmel, “How SONY LANC protocol works,” http://www.boehmel.de/lanc.htm.

[4] R. Crockett, LANC Shepherd, http://www.ledametrix.com/index.html.

[5] W. Bloss, ste-fra® LANC. http://www.digi-dat.de/produkte/index_eng.html.

[6] D. Vrancic, S. L. Smith, “Permanent synchronization of camcorders via LANC protocol,” Proc. SPIE, Vol. 6055, 60550I, 2006.

[7] http://dsc.ijs.si/3DLANCMaster/

[8] T.Xian, S.-Y. Park, M. Subbarao, "New Dynamic Zoom Calibration Technique for a stereo-vision based multi-view 3D modeling system", Photonics East, Philadelphia, Proceedings of SPIE, Vol. 5606, pages 106 to 116, Oct. 26, 2004.

[9] Y.P. Tan, S. R. Kulkarni and P. J. Ramadge, “A New Method for Camera Motion Parameter Estimation,” IEEE International Conference on Image Processing, Vol. 1, pp. 406-409, Oct 1995.

[10] I. Grinias and G. Tziritas. “Robust pan, tilt and zoom estimation,” Int. Conf. on Digital Signal Processing, 2002.

[11] Y. P. Tan, S. R. Kulkarni, and P. Ramadge, “Rapid estimation of camera motion from compressed video with application to video annotation,” IEEE Trans. Circuits Syst. Video Technol., vol. 10, pp. 133–146, Feb. 2000.

[12] ITU-R Recommendation BT.500-10, Methodology for the subjective assessment of the quality of television pictures, 2000.

[13] R.G.Willson and S.A.Shafer, “What is the Center of the Image?,” Journal of the Optical Society of America A, Vol.11, no.11, pp. 2946-2955, Nov. 1994.

[14] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, Nov. 2004.

CHAPTER 5: EFFICIENT INTER-VIEW PREDICTION FOR MULTIVIEW CODING[6]

5.1 Introduction

Free viewpoint television (FTV) is believed to be the next major revolution in TV's history [1]. FTV allows on-screen images to emerge or penetrate into the viewer's space (3D perception). It also provides TV viewers with interactive features: the viewer can adjust the 3D depth perception based on his/her preferences and can also choose a viewing angle within a visual scene (free navigation). FTV involves capturing the scene from multiple views with a setup of N synchronized cameras and then transmitting the multiview streams to the end-user. One major challenge in multiview applications is the transmission of huge amount of data, requiring the development of highly efficient coding schemes. Another challenge is that any compression scheme designed specifically for multiview video streams should support random access functionality, allowing viewers to access arbitrary views with minimum time-delay.

A straightforward approach for multiview video coding (MVC) is simulcast coding, which compresses each video stream independently [2]. While this scheme excessively exploits temporal and spatial correlations within each stream, it does not benefit from the existing correlation between different views. Multiview sequences show a scene from many different viewing angles, which means that there is a high possibility of inter-view correlation between the multiple streams. The existence of this multiple

[6] A version of this chapter has been submitted for publication. Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (2010) Efficient inter-view prediction structures for multiview coding.

correlation makes multiview video coding have a different structure from single-view coding techniques. This issue is addressed in the latest recommendation for MVC by the ISO/IEC Moving Pictures Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG), known as H.246/MVC. H.246/MVC uses hierarchical B pictures for each view and, at the same time, applies inter-view prediction to every 2nd view, using already encoded frames from adjacent camera views [3, 4]. The objective is to predict the video frame from a given camera using one or more neighboring-camera video frames (disparity estimation) in addition to the consecutive frames of the given camera stream (motion estimation). This approach can improve the PSNR (Peak signal-to-noise ratio) quality by up to 3.2 dB compared to simulcast coding (coding each stream separately). The performance in this case strongly depends on the arrangement of the cameras [4]. Although inter-view prediction enhances the compression performance of MVC, it also introduces computational complexity and random-access delay. A straightforward approach for facilitating random access is to increase the number of I and P frames, which in turn hampers the compression efficiency. Thus, there is a tradeoff between compression performance and random access time-delay in the prediction structure suggested by H.264/MVC.

To improve the multiview video compression efficiency, the study in [5] suggests using the already encoded frames from adjacent views and the depth map to synthesize a virtual view. Then the synthesized frames are used for predicting the actual view. This scheme can achieve PSNR gains of up to 2dB relative to simulcast coding. An important issue here is the computing, coding and transmission of the depth map information [5]. Depth maps either exist, or must be obtained. In the former case, 5 to 10% of the bitrate

needs to be devoted for transmitting the depth maps. For the latter case, a method that conveys depth maps to the decoder needs to be defined at the encoder side. In terms of random access delay, the proposed method in [5] introduces similar time-delay to that of H.264/MVC.

To reduce random access delay, the study in [6] proposes an image-stitching based MVC method. This approach generates a stitched reference and encodes multiview sequences using inter-view prediction. The result is reduced-delays during the decoding stage, but this approach still involves high computational complexity due to inter-view prediction. Experimental results show that this MVC method increases the PSNR by 1.5~2.0 dB and reduces the bit rate by 10% compared to simulcast coding. In the best scenario, the performance of this scheme in terms of PSNR is expected to be similar to H.264/MVC.

In this paper, we investigate two new multiview video coding schemes which try to benefit from the recommended approaches in [5] and [6] while overcoming their drawbacks. In our first method (Adaptive MVC), inter-view prediction is exploited by using the already encoded frames from the neighbouring views as well as a synthesized version of the to-be-coded frame. This synthesized frame is constructed using the already encoded frames from neighbouring views, with no need to have depth information of the scene (unlike [5]). Moreover, in our method, the indices of all possible reference frames stored in the decoded picture buffer (DPB) are adaptively re-sorted to guarantee the best bitrate performance. The use of multi reference frames significantly enhances the coding efficiency in conventional 2D video. However, the multi reference frames are not sequential in the multiview case and our study shows that the proper selection of the

reference frames and the order of sorting can improve the bitrate. There is no provision for this issue in the MVC standard. The only available option is to specify if all the inter prediction reference pictures are placed ahead or after the inter-view prediction ones in the reference picture list.

Our second approach proposes a new panorama-based multiview video coding structure, which outperforms the current MVC standard in terms of computational complexity and random access functionality. It turns out that the same method offers exceptional compression performance as well. The proposed multiview video coding structure transforms the middle view to a panoramic view of the scene. In order to take advantage of the existing correlation between views, instead of using inter-view prediction, residue streams are created as the difference of the luma and chroma values of overlapping regions of each view and the panoramic view. Finally, the panoramic stream and all residue streams are encoded separately using hierarchical B pictures. This structure eliminates the computational complexity and random-access delay that exist in H.264/MVC while it enhances the coding efficiency.

The rest of the paper is structured as follows. Section 5.2 presents a short overview of the current MVC standard. Section 5.3 elaborates on our proposed approaches. Experimental results are presented and discussed in Section 5.4 and conclusions are drawn in Section 5.5.

5.2 Overview of H.264/MVC

H.264/MVC is the latest recommendation for multivew video coding by MPEG and the ITU-T VCEG. Figure 5.1 shows the prediction structure supported by

H.264/MVC which utilizes hierarchical-B picture structure, involving an 8-view sequence and GOP (Group of Picture) length of 8. The horizontal and vertical directions represent the temporal and spatial axes, respectively.

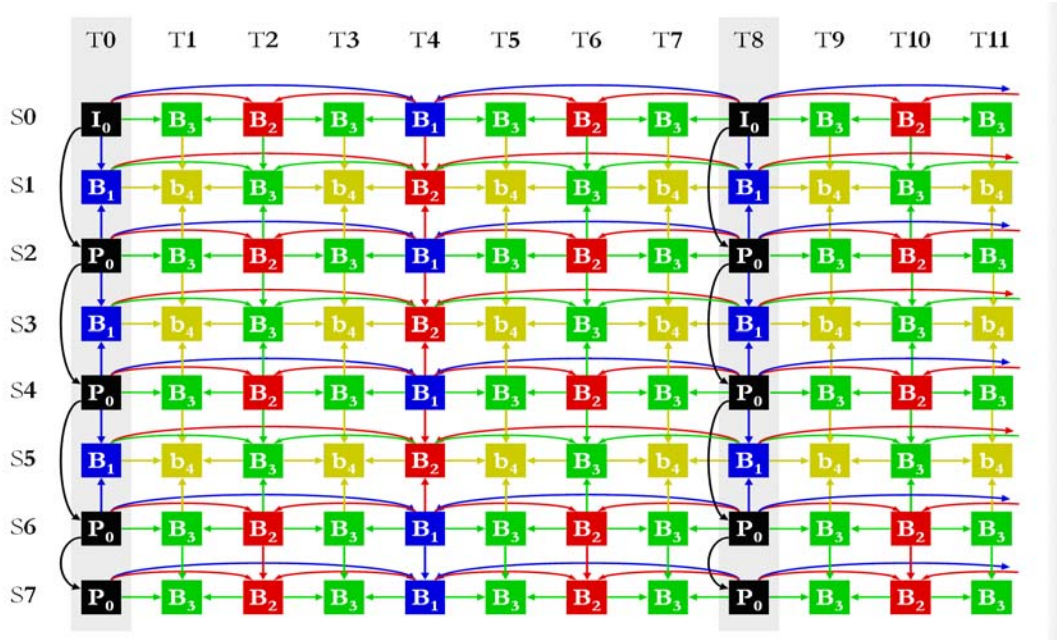

Figure 5.1 Prediction structure recommended by H.264/MVC.

As illustrated in Figure 5.1, H.264/MVC tries to predict the video frame from a given camera using one or more video frames of neighboring-cameras (disparity estimation) in addition to the consecutive frames of the given camera stream (motion estimation). Although this inter-view prediction approach enhances the compression performance of MVC, it also introduces computational complexity and random-access delay. Random access delay is measured based on the maximum number of frames needed to be decoded in order to access a B-frame in the hierarchical structure. The access delay for the highest hierarchical order is given by:

$$F_{max} = 3 * level_{max} + 2 * \lfloor (N-1)/2 \rfloor \qquad (5\text{-}1)$$

where $level_{max}$ is the highest hierarchical order and N is the total number of views. For instance, in order to access a B-frame in the 4th hierarchical order (B4-frames in Figure 5.1), 18 frames ($F_{max} = 18$) must be decoded.

5.3 Proposed MVC Methods

The following subsections elaborate on our proposed MVC schemes: adaptive MVC and panorama-based MVC. Note that both of the proposed methods are based on the H.264/AVC standard which has been recognized as the most promising video compression platform for MVC [7].

5.3.1 Adaptive MVC method

The prediction structure of our adaptive MVC method is illustrated in Figure 5.2. For the sake of simplicity, Figure 5.2 shows only three of the camera views involved in inter-view prediction. It illustrates the hierarchical B picture prediction structure that is applied to video streams captured by cameras C_1 and C_3 and the inter-view prediction scheme applied to the video stream captured by camera C_2.

Unlike the existing methods ([3] and [5]) that use either synthesized frames or adjacent views to improve compression, our proposed scheme is based on a combination of these approaches. To properly address the synthesized reference frame and other reference frames in the reference frame list, we have proposed an adaptive reference-frame resorting technique. As it can be observed from Figure 5.2, the random access delay of our MVC method is similar to that of MVC standard. The following subsections elaborate on this technique as well as our synthesized reference construction approach.

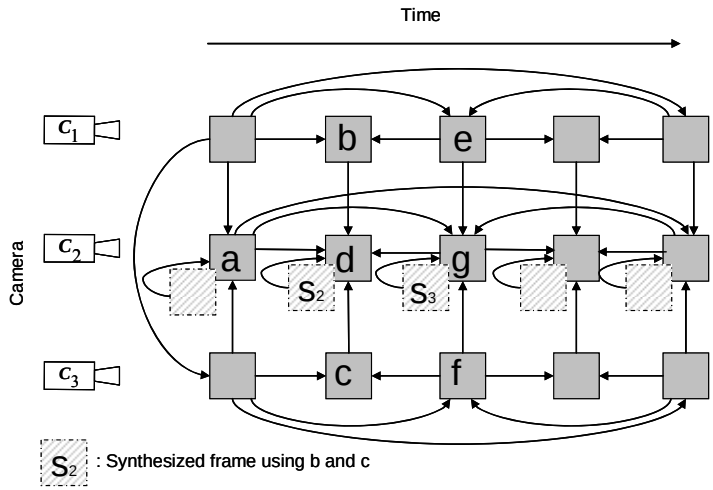

Figure 5.2 Prediction structure of proposed adaptive MVC method.

5.3.1.1 Adaptive re-sorting of reference frame indices

Similar to conventional 2D videos, using multi reference frames in multiview coding can assist in improving the compression bitrate of multiview video sequences. However, in the case of MVC, using reference frames belonging to neighboring views may cause increased bitrates if reference frame management is not handled properly. This is because the entropy coding schemes used by the H.264/AVC standard to code the reference frame indices were designed based on the distribution models of 2D video sequences. The principle idea is to assign short codewords to frequently-occurring elements while infrequent elements are assigned longer codewords. H.264/AVC uses "Exponential Golomb Codes" to code the reference frame indices [8]. Table 5.1 shows the first nine elements of the Exponential Golomb Code table for a given code number, which is the reference frame index in this case.

Table 5.1 Exponential Golomb codes.

Exp Golomb Codes									
code Num	0	1	2	3	4	5	6	7	8
Code	1	010	011	00100	00101	00110	00111	0001000	0001001

The original H.264 standard assigns index 0 (the shortest codeword) to the last encoded frame by default and larger indices to the temporally distant frames from the to-be coded frame. This works pretty well for temporal prediction of normal 2D video sequences, since normally the last encoded frame has the highest correlation with the to-be coded frame. However, for the case of inter-view prediction (as shown in Figure 5.2) the last encoded frame is not necessarily the most frequently referenced one. Because in inter-view hierarchical prediction frames from other views are engaged, it is difficult to predict which one has the highest correlation with the to-be coded frame. For this reason, the existing Exponential Golomb Code is not efficient for the multiview hierarchical approach. Currently, this problem is not addressed by the MVC standard. The only available option is to pre-specify if all the inter prediction reference pictures are reordered so they are placed ahead of the inter-view prediction pictures or after them in the reference picture lists. In order to show the importance of proper ordering the reference frame indices, Figure 5.3 compares two reference-frame sorting scenarios for a multiview test sequence ("Ballroom"). Note that S_2 represents the proposed synthesized reference frame (see subsection 5.3.1.2). Considering "d" as the frame to be coded (see Figure 5.2), in one scenario the reference frames are sorted as "S_2, b, c, a" and in the other as "b, S_2, c, a" (where index 0 is assigned to frame "a"). We observe that 308 more bits are needed for the transmission of frame "d" for the first scenario. Moreover, reordering the reference frames has changed the referral percentage to all frames due to the rate distortion optimization used by H.264/AVC.

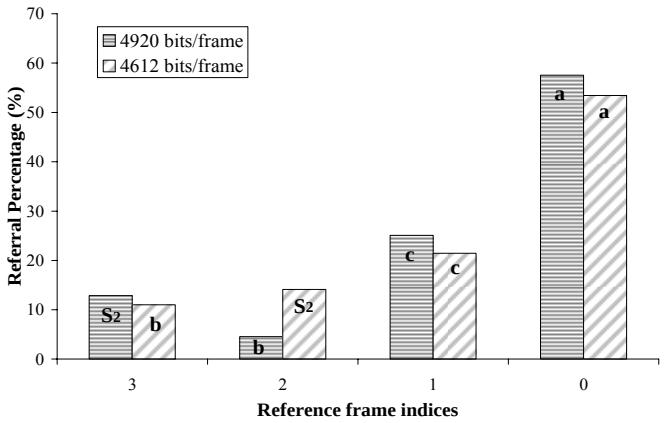

Figure 5.3 Experimental results for coding frame "d" within "Ballroom" sequence with reference frames arranged as "S2, b, c, a" versus "b, S2, c, a".

One possible approach to resolve the reference-frame management issue is to consider designing a new Exponential Golomb Code using a distribution model for MVC. However, performance evaluations have shown that it is almost impossible to come up with such a model, since factors such as the distance between cameras and multiview content directly affect the number of times different frames are referenced. In fact, our tests have shown that even within consecutive GOPs (of a 3D sequence captured by the same set of cameras) the frequency at which the frames are referenced varies drastically.

In this study, we propose an adaptive histogram-based technique which automatically reorders the indices for the reference frames according to the frequency at which they are referred. To achieve this while encoding each frame within a view, the percentage of referral to different reference frames is counted. Then, when encoding the next frame, the reference frame indices are re-sorted (if necessary) such that shorter codes are assigned to the more frequently used reference frames.

In order to accurately decode the compressed stream, extra bits need to be added to the frame header to convey the re-sorting reference frame lists. To achieve this, reference frames are indexed during encoding by numbers in the binary format and with a default arrangement. Since up to five reference frames are used in our inter-view prediction, only a total of fifteen bits is needed to define this arrangement. If during encoding, the proposed histogram-based technique identifies that reordering is needed, the new arrangement information (15 bits) is added to the to-be coded frame's header information accompanied with a flag that indicates whether the reference frames need to be reordered or not. The resulting increase in the total bit rate is negligible, considering the fact that this extra byte is sent only when reference frame reordering takes place. In the worst case scenario, when resorting is needed for every frame, the bit rate of a 30fps video sequence will increase by a mere 450 bits/second. This is a negligible amount compared to the savings achieved as illustrated in Figure 5.3.

5.3.1.2 Synthesized frame construction

The objective here is to synthesize an extra reference frame based on existing information in the encoder, such that this synthesized reference frame is more similar to the to-be-coded frame than other reference frames. This will result in reduced residue information (enhancing motion estimation process) which in turn will improve the overall compression performance. As Figure 5.2 shows, for each frame captured by camera C_2 one extra reference frame will be synthesized based on the decoded version of the already encoded corresponding frames from C_1 and C_3. In order to synthesize such a frame, the overlapping areas among camera views need to be determined first. Thus, we must know the global disparity vectors among the adjacent multiview frames. A global disparity

vector (GDV) shows the horizontal and vertical displacement between two multiview pictures. Basically, a GDV yields the best overlapping area for the two adjacent camera frames. The simplest way to find a GDV is as follows. First, we match two images. Then, we shift one image by a pixel and calculate the matching error between the two overlapping regions, repeatedly. The shift that corresponds to the minimum matching error (which maximizes similarity) is defined as the GDV between the two views.

In video coding, the sum of absolute differences (SAD) or the sum of square differences (SSD) is used as the cost function for finding matching areas. However, these cost functions are sensitive to the brightness level of the two frames, and have been shown to achieve very poor performance when there are brightness variations between views [9]. Since among multiview video streams there may be substantial brightness variations, a more robust matching criterion is needed. For this reason, we use the normalized cross correlation (NCC) defined as:

$$NCC(\Delta x, \Delta y) = \frac{\sum_{x,y}[Y_n(x,y) - m_n][Y_{n+1}(x-\Delta x, y-\Delta y) - m_{n+1}]}{\sqrt{\sum_{x,y}[Y_n(x,y) - m_n]^2 . \sum_{x,y}[Y_{n+1}(x-\Delta x, y-\Delta y) - m_{n+1}]^2}} \quad (5\text{-}2)$$

where $Y_n(x,y)$ and $Y_{n+1}(x,y)$ are the luma information of two adjacent views, m_n and m_{n+1} are the mean of Y_n and Y_{n+1}, and Δx and Δy are horizontal and vertical shifts. The Δx and Δy that maximize NCC will be used as horizontal and vertical components of GDV:

$$GDV_{n,n+1} = \arg\max NCC(\Delta x, \Delta y) \quad (5\text{-}3)$$

Note that in equation (5-2), which calculates NCC, the mean of the luma values of the overlapped area is subtracted from the luma value of each pixel and the energy of the overlapped area is normalized. This provides robustness to changes in brightness between

the views. This procedure is implemented only once at the beginning of encoding, assuming the camera arrangement remains unchanged. The information about GDV of adjacent views is sent to the receiver as an MVC supplemental enhancement information (SEI) message [3] for recovering the original views at the decoder side.

It is common practice that, before coding starts, all multiview videos undergo a pre-processing stage, which includes rectification and colour-correction. With such pre-processing at the encoder side, the inter-view correlation among multiview videos is increased, resulting in better overall coding efficiency [10]. Assuming that cameras are arranged in parallel, rectification involves applying a homography matrix to frames in order to make the image plane parallel to the baseline (the line connecting the camera centres).

Colour-correction is also important in multiview applications, and involves correcting variations in the colour of views captured with different cameras, something that negatively affects performance when the videos are compressed with inter-view prediction. In our implementation, we applied the colour-correction algorithm suggested in [11]. In this approach, one view is chosen as the reference and all other views are corrected to match it. The corrected YUV values are expressed as a weighted linear sum of the original YUV values and an offset. Disparity estimation is used to find matching points between the view being corrected and the reference, and a least squares regression is performed on the set of matching points to find the optimal weight values for correction.

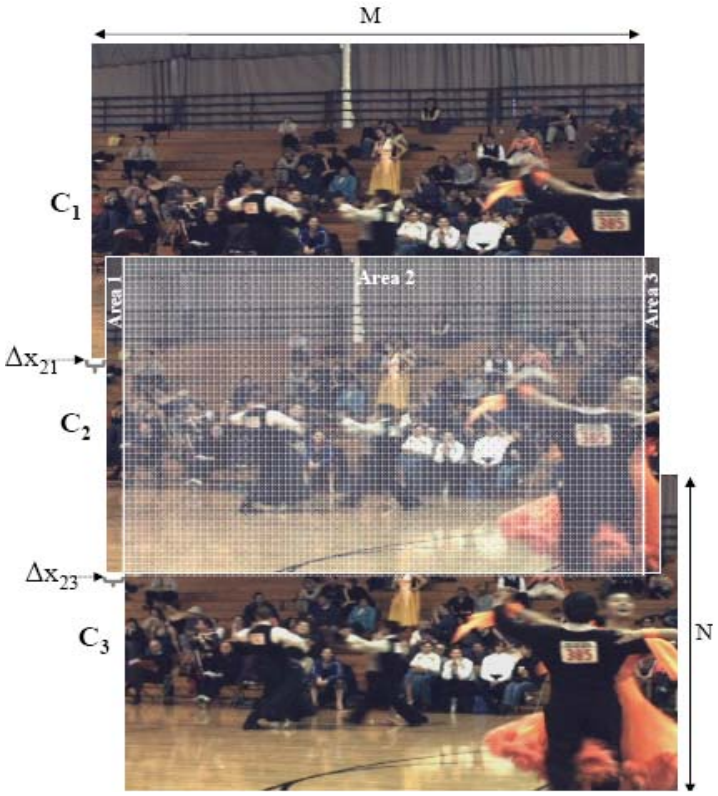

Figure 5.4 Overlapping areas among frames captured by cameras, Area 1: C_1 and C_2, Area 2: C_1, C_2 and C_3, Area 3: C_2 and C_3.

Because of the parallel arrangement of cameras and the rectification of all the video streams, only a horizontal shift is needed for maximizing the normalized cross correlation and thus finding the overlap areas as shown in Figure 5.4. The corresponding horizontal shift for frames captured by C_1 and C_2 is given by $\Delta x_{2,1}$. The same procedure is used for finding the best overlap between frames captured by C_2 and C_3 and the corresponding horizontal shift of $\Delta x_{2,3}$. Having $\Delta x_{2,1}$ and $\Delta x_{2,3}$ as well as the decoded versions of the C_1 and C_3 frames (available in the decoded picture buffer), we can construct the synthesized frame. The intensity and colour of each pixel within the synthesized reference frame are set to the average intensity and colour of the corresponding pixels in C_1 and C_3 for the common overlapped area (denoted as Area 2 in

Figure 5.4). For the non-overlapped areas (Area 1 and Area 3 in Figure 5.4), the synthesized frame's intensity and colour are equal to the intensity and colour of either C_1 or C_3. The synthesized reference frame is added to the decoded picture buffer, so it can be used as an extra reference frame, and the multiview videos are compressed based on the prediction structure shown in Figure 5.2. Note that the synthesized frame is not transmitted, but rather it is reconstructed at the decoder side (using readily available decoded version of corresponding frames from adjacent views and GDV information) whenever it is used as a reference.

5.3.2 Panorama-based MVC method

While our adaptive MVC method outperforms the H.264/MVC standard in compression performance, it involves the same or slightly higher computational complexity and the same random access delay. The proposed panorama-based MVC method is designed to address these two issues while trying to improve on the compression performance as well. Figure 5.5 shows the prediction structure of our proposed panorama-based MVC method which is applied to an 8-view sequence. The horizontal and vertical directions represent the temporal and spatial axes, respectively. The main feature of our proposed prediction structure is that the middle view (which is chosen as the base view) is transformed to a panoramic view through a process described in detail in the following subsection. The reason for choosing the middle view as the base is because this view generally has more overlapping parts with the other views (an important requirement for creating the panoramic view). If the number of total views is even, one of the middle views is chosen as the base view. As shown in Figure 5.1, the reference MVC codec uses the leftmost view as the base view.

As it can be observed from Figure 5.5, in our approach, we reduce the redundancy between views by subtracting the luma and chroma values of each view from corresponding parts of the panoramic view. The resulting residue frames have very low energy, thus improving the overall compression performance [2]. Compression of residues is an approach that has been shown to be very effective in compressing MRI (magnetic resonance imaging) images [12]. This is because very low energy remains in the final residue frames, which in turn may be represented by a much smaller number of bits (i.e., better compression) [2].

Since the proposed prediction structure does not include traditional inter-view prediction, the level of computational complexity and random-access delay inherent in the standard MVC is significantly reduced. In our proposed panorama-based MVC structure, the maximum number of frames that must be decoded in order to access a B-frame with the highest hierarchical order is formulated as follows:

$$F_{\max} = 2 * level_{\max} + 5 \tag{5-4}$$

where $level_{max}$ is the highest hierarchical order. Applying this to the coding structure shown in Figure 5.5 indicates that a total of $F_{max} = 11$ reference frames are required to be decoded to access a B-frame in the 3rd hierarchical order (compared to 18 for the H.264/MVC). As equation (5-4) shows, the random-access delay in our approach is independent of the increase in the total number of views.

The following subsections elaborate on the process of creating the panorama view and residue video streams in our method.

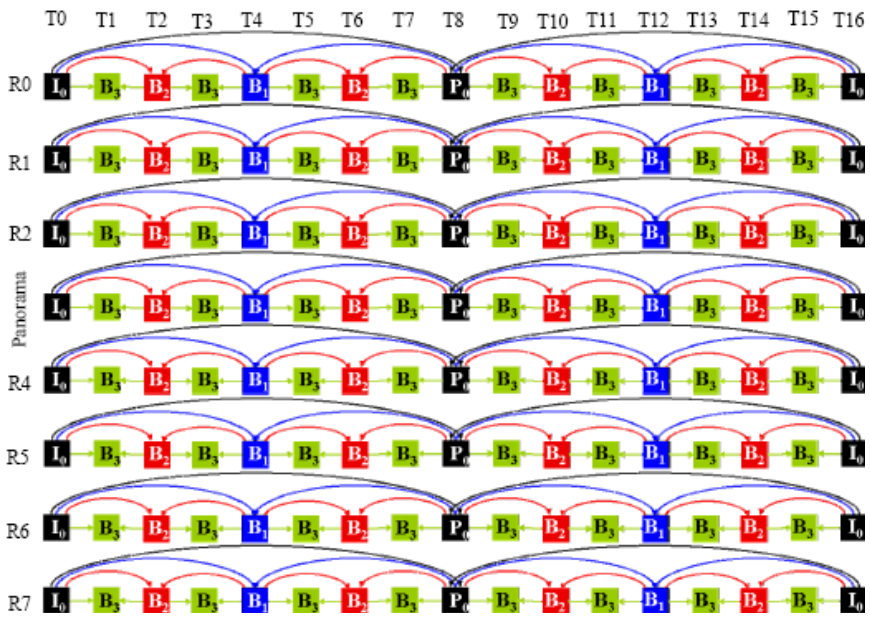

Figure 5.5 Our proposed panorama-based MVC prediction structure.

5.3.2.1 Panorama-view creation

The panorama-view creation in our approach is the process of transforming the base/middle view of multiview images (acquired by multi cameras with parallel set-up) into a panoramic view that includes all parts of the scene. Figure 5.6 shows a panorama image which is composed of four images. In this example, we perform the panorama-view creation with the assumption that there is no vertical disparity (for the sake of simplicity). The 4^{th} view is chosen as the base view and is transformed to the panorama view as shown in Figure 5.6. We obtain the panorama view by using the entire 4^{th} image and stitching the image sections of views 1, 2, and 3 that do not exist in the 4^{th} image into the base view. In order to create such a view, the overlapping areas between each pair of adjacent camera views must be determined first. In order to achieve this, we must estimate the global disparity vectors among the adjacent multiviews as explained in subsection 5.3.1.2. Here also the information about GDV of adjacent views is sent to the

receiver as an MVC SEI message [3] for enabling the recovery of original views at the decoder side. Also in a pre-processing stage, all multiview videos are rectified and colour-corrected (see subsection 5.3.1.2).

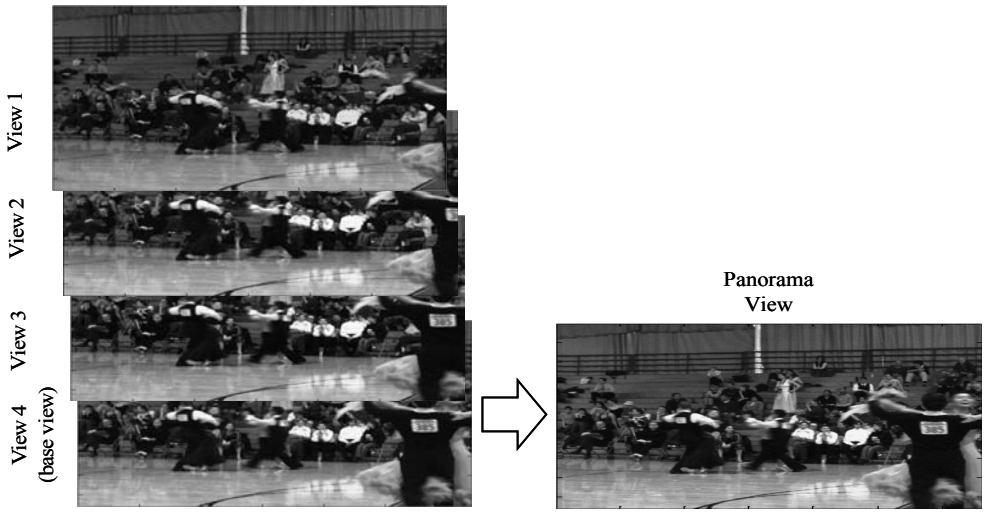

Figure 5.6 Multiview images and the created panorama-view.

5.3.2.2 Residue-view creation and encoding

Figure 5.7 shows the block diagram of the redundancy reduction method implemented in our prediction structure. As it can be seen, residue-views are created by subtracting the luma and chroma values of each view from those of the corresponding overlapping region on the panorama view. The overlapping area on the panorama image is determined using the global disparity vector between views. Since the multiview camera setting is parallel and, as explained above, the video streams are rectified and colour-corrected, the luma and chroma values of the overlapped parts in the residue frame are close to zero while those of the non-overlapped areas are all zeros (as shown in Figure 5.7). In order to have all-positive values in the residue frame, the luma and chroma values are shifted by 255. As a result, the residual pixel-values are stored in the 9-bit format instead of the 8-bit.

During encoding, all residue streams and the panorama view stream are compressed separately using the hierarchical B-picture prediction-structure (simulcast coding). As it can be observed from Figure 5.5, the proposed panorama-based MVC scheme does not implement inter-view prediction (disparity estimation). This has advantages in terms of computational complexity and random-access delay, while improving compression performance.

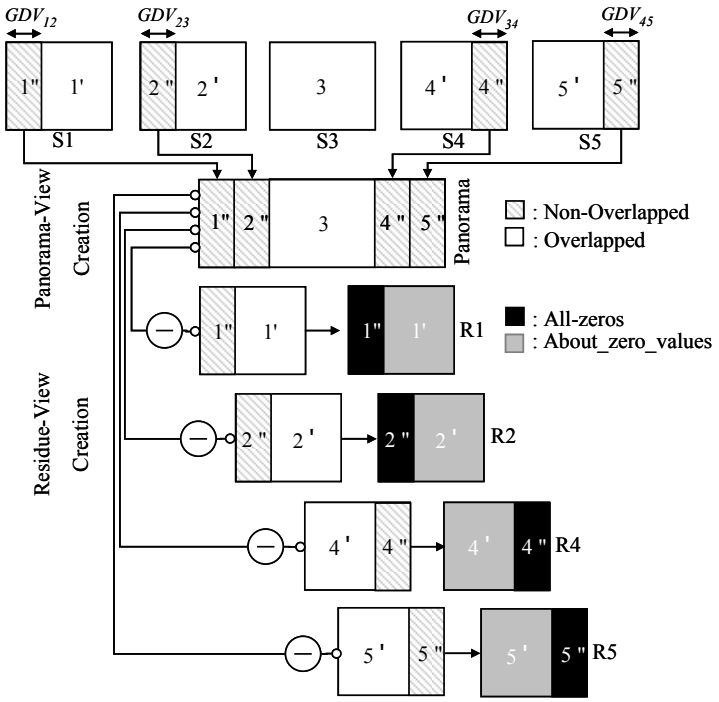

Figure 5.7 Panorama-view and Residue-view creation.

In order to recover any view at the decoder side, only the corresponding residue-view, the panorama-view and GDV information are required. The flowchart of the proposed MVC algorithm is shown in Figure 5.8.

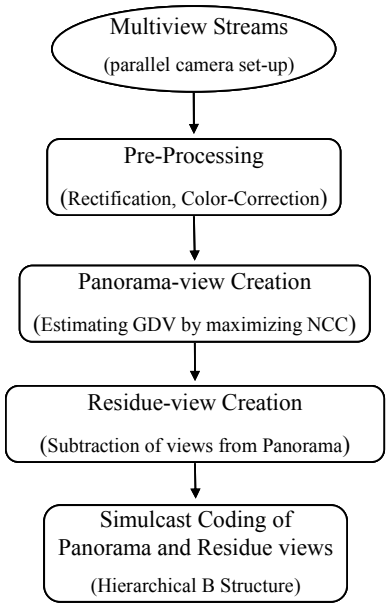

Figure 5.8 Flowchart of the proposed panorama-based MVC algorithm.

5.4 Experiments and Discussion

For our experiments we used the test sequences suggested by the MPEG−MVC group. Table 5.2 lists the names, size, frame rate, and camera arrangement of these multiview video streams. The test sequences are in YUV 4:2:0 format and have been already rectified using the homography matrix. Only 16 center views of the "Rena" test set (camera no.38 – 53) are publicly accessible. For our experiment, before encoding is performed by our proposed schemes or by the MVC standard, all the test sequences have been colour-corrected using the technique recommended in [11].

Table 5.2 Test sequences.

Sequences	Image Property	Camera Arrangement
Ballroom	640x480, 25fps (rectified)	8 cameras with 20cm spacing; 1D/parallel
Rena	640x480, 30fps (rectified)	100 cameras with 5cm spacing; 1D/parallel

Both of our proposed MVC prediction structure methods were implemented on H.264 (version JM 12.2), which was configured to employ Context-adaptive Binary Arithmetic Coding (CABAC) [2]. The performance of our proposed multiview video coding schemes was tested against the MVC standard (JMVM 8.0) in terms of PSNR and bitrates. The MVC standard was configured to employ CABAC with GOP-length of 8 (common practice since GOP=16 introduces much more access delay for H.246/MVC). The GOP sizes of 8 and 16 were used for our proposed adaptive MVC scheme and panorama-based MVC method, respectively. Note that if the GOP size of 16 is used for H.264/MVC and our proposed adaptive MVC, the maximum number of reference frames needed to be decoded to access the B-frame in the highest hierarchical order would become 21 instead of 18 frames, with a very slight bitrate improvement. The illumination compensation was not used in the encoding process since the test sequences were colour-corrected before coding.

Figure 5.9 shows the performance of our proposed MVC schemes versus the H.264/MVC standard in terms of PSNR and bitrates. It is observed that our proposed schemes outperform the MVC standard by up to 1dB PSNR in the adaptive MVC case and by up to 2.13 dB in the panorama-based MVC method. More specifically, the average picture quality improvement achieved by the adaptive MVC scheme is 1 dB for 'Rena', and 0.42 dB for 'Ballroom'. In other words, the proposed adaptive MVC approach enhances the average compression ratio by 22.97% for 'Rena' and by 10.14% for 'Ballroom'. For the panorama-based MVC approach, the average picture quality improvement over the MVC standard is 2.13 dB for 'Rena', and 0.56 dB for 'Ballroom'. The proposed panorama-based MVC approach enhances the average compression ratio

by 24.6% for ‘Rena’ and by 11.99% for ‘Ballroom’. We observe that our methods result in a higher coding improvement for ‘Rena’, which is captured with cameras positioned 5cm apart, than for ‘Ballroom’ captured with cameras positioned at 20cm apart. This shows that our methods work better for multiview applications where cameras are arranged for a natural 3D experience (small distance between cameras). Comparing the performance of the panorama-based MVC against the adaptive MVC we observe that the panorama-based MVC results in higher compression efficiency especially when the multi cameras are closer together. Note that, although in our panorama-based approach the required bitrate for the panorama view is higher compared to that for the original middle stream (due to larger frame-size), the overall average bitrate is much less than the standard MVC. This is due to the significant reduction in the information carried in the generated residue streams compared to that in the original streams.

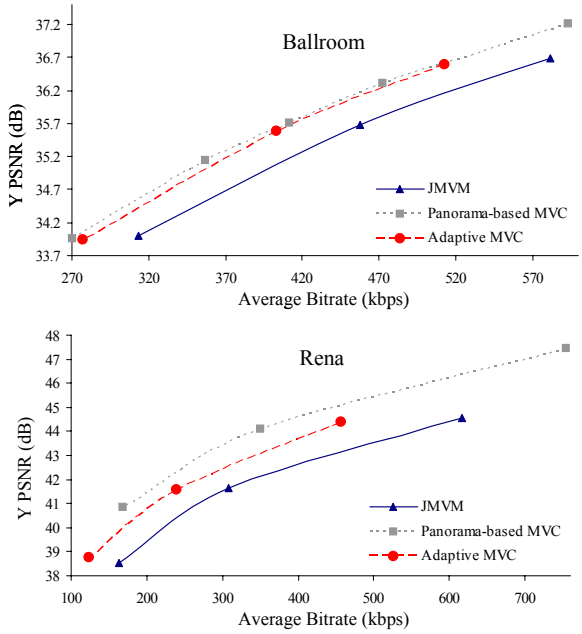

Figure 5.9 Coding results for “Rena” and “Ballroom” test sequences.

In the proposed panorama-based MVC scheme, since inter-view prediction was not exercised, the computational-complexity and speed of the coding process are significantly reduced compared to H.264/MVC and the stitching method proposed in [6]. In addition, comparing equations (5-1) and (5-4) confirms that our panorama-based scheme imposes less random-access delays than the standard MVC approach. This is always true, if there are at least three multiviews and at least one B-frame is used, which is the case in multiview video coding. Regarding random-access delay, the maximum number of reference frames that need to be decoded to access the B-frame in the highest hierarchical order is 11 frames for the panorama-based MVC scheme compared to 18 frames required for the MVC standard (39% improvement). The random access delay imposed by the adaptive MVC is similar to that of the standard MVC (i.e., 18 frames).

5.5 Conclusion

We have presented two efficient video coding schemes called adaptive MVC and panorama-based MVC. The proposed adaptive MVC scheme constructs an extra reference frame, which is used to improve the accuracy of motion estimation process of MVC standard. Later, our adaptive approach automatically re-sorts the reference frame list to prevent the use of extra bits for coding reference frame indices. Performance evaluations show that the proposed scheme outperforms H.264/MVC by up to 1 dB PSNR (up to 22.97% compression ratio enhancement). The reason is due to the synthesized reference frame and the adaptive re-sorting of reference frame indices used by our method.

In the panorama-based MVC, inter-view prediction is replaced with a residue-stream coding process. Our algorithm transforms the middle view to a panoramic view of

the scene. Then the residue streams are created as the difference of the luma and chroma values of overlapping regions of each view and the panoramic view. Finally, the panoramic stream and all the residue streams are encoded separately (simulcast coding). Performance evaluations show that the proposed scheme outperforms the MVC standard by up to 2.13 dB PSNR (or up to 24.6% compression ratio enhancement). This is due to the compression of multi residue streams (which include zero or close to zero luma and chroma information) instead of original multiview streams. In addition, since inter-view prediction (which imposes time-consuming complex composition and random-access delay to MVC) is replaced with the residue-stream coding process, the random-access delay is reduced by 39%.

In summary, both of the proposed prediction structures enhance the compression ratio compared to MVC standard. However, the panorama-based MVC shows significantly superior compression performance when the multiview sequence is captured via close-distant cameras. In addition, the panorama-based MVC involves less computational complexity and lower random-access delay.

5.6 References

[1] “Applications and Requirements for 3DAV,” document N5877 MPEG Meeting, Trondheim, Norway, Jul. 2003.

[2] I. E.G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia, John Wiley & Sons, Inc., England, 2003.

[3] A. Vetro, P. Pandit, H. Kimata and A. Smolic, "Joint Multiview Video Model (JMVM) 8.0," ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Doc. JVT-AA207, Apr. 2008.

[4] 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. ICIP, Canada, pp. 1717-20, Jul. 2006.

[5] E. Martinian, A. Behrens, J. Xin, A. Vetro, H. Sun, “Extensions of H.264/AVC for Multiview Video Compression,” in Proc. ICIP, Canada, pp. 2981-4, Jul. 2006.

[6] K Sohn, Y Kim, H Ko, J Seo, "Image stitch-based multiview video coding," in Proc. SPIE The International Society for Optical Engineering, 2007.

[7] P. Pandit, A. Vetro, Y. Chen, “Joint Multiview Video Model (JMVM) 7 Reference Software,” N9579, MPEG of ISO/IEC JTC1/SC29/WG11, Antalya, Jan. 2008.

[8] D. Marpe, T. Wiegand, and G. J. Sullivan, “The H.264 / MPEG4 Advanced Video Coding Standard and its Applications”, IEEE Communications Magazine, Vol. 44, No. 8, pp. 134-144, Aug. 2006.

[9] H. Hirschmuller, and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," Proc IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2007.

[10] “Survey of Algorithms used for Multi-view Video Coding (MVC),” ISO/IEC JTC1/SC29/WG11/N6909, China, Jan. 2005.

[11] C. Doutre, and P. Nasiopoulos, "A Colour Correction Preprocessing Method For Multiview Video Coding," Proc European Signal Processing Conference (EUSIPCO 2008), Aug. 2008.

[12] V. Sanchez, P. Nasiopoulos and R. Abugharbieh, “Efficient lossless compression of 4-D medical images based on the advanced video coding scheme”, IEEE Transactions on Information Technology in Biomedicine, Vol. 12, No. 4, pp. 442 – 446, July 2008.

CHAPTER 6: CONCLUSIONS

6.1 Significance of the Research

The availability of three-dimensional (3D) TV as a commercialized product is not far from reality. Future 3D TV sets will not only allow the on-screen images to emerge or penetrate into the viewers' space, but will also provide viewers with interactivity features such as depth-perception adjustment and random-access to different viewing angles. This can be achieved by capturing the scene from multi-view points with a setup of N synchronized cameras and transmitting the resulted multi-view streams to end users. The introduction of 3D TV to the consumer market will be an endless success if a wide variety of 3D content is available, the quality of the delivered (compressed) content is high and the overhead for transmitting the additional data (second view) is not prohibitive. To this end, the objectives of my PhD thesis have been to 1) propose algorithms for high-quality 3D content generation from 2D videos, 2) enhance the quality of captured 3D content and 3) develop efficient video compression schemes for 3D TV applications.

In chapter 2, an efficient method that converts 2D video sequences to 3D is presented. This method utilizes the motion information between consecutive frames to approximate the depth map of the scene. To estimate the depth map, the horizontal motion captured by a single camera is revised and then approximated as the displacement between the right and left frames captured by the two cameras used in a stereoscopic set-up. To enhance the visual depth perception, a non-linear scaling model is then applied to

the modified motion vectors. The low complexity of our approach and its compatibility with future 3D systems, allows real-time implementations at the receiver-end with no additional-bandwidth burden on the network. Performance evaluations show that our method outperforms the existing H.264-based depth map estimation technique [1] by up to 1.84 dB PSNR, providing more realistic depth representation of the scene. Moreover, the subjective comparison of the results (obtained by viewers watching the generated stereo video sequences on a 3D display system) confirms the better performance of our method.

The presented algorithm in chapter 3, aims at improving the proposed 2D to 3D conversion scheme in chapter 2 by using color-texture segmentation to identify objects and correct motion vectors accordingly. Our objective and the subjective evaluations show that this approach improves the performance of our method presented in chapter 2 by enhancing the quality of the estimated depth maps.

In chapter 4, we first study the impact of zoom mismatch on subjective 3D quality. Then we propose an effective post-processing algorithm, which aims at correcting the vertical parallax caused by the unsynchronized zooming in stereo video recording. The proposed scheme finds the matching points, i.e., the corresponding points in the left and right views. The relationship between the points is then found (using least squares regression) so as to estimate the amount of scaling and translation needed to align the views. Experimental results show that our method produces videos with negligible scale difference and vertical parallax.

Chapter 5 presents two schemes for efficiently encoding N multiview video streams. These multiview video coding (MVC) schemes utilize the strong correlation that

exists between all multi-view streams to improve compression efficiency. The proposed schemes result in a more accurate motion prediction and a less computationally complex prediction structure compared to the recent H.264/MVC standard. Experimental results confirm that both proposed schemes outperform the recent MVC standard in terms of compression efficiency. In addition, one of the proposed approaches introduces significantly less random access delay compared to the MVC standard due to its prediction structure.

6.2 Potential Applications of the Research Findings

Considering the recent penetration of 3D technology to the entertainment market, there are several applications for the schemes proposed in this thesis.

- The main target of our proposed scheme for 2D to 3D video conversion are content producers. These include movie studios (Hollywood), film production companies, television networks, and content owners in general. The generation of 3D content from existing 2D material will not only help enable the 3D market but it will also allow the above parties to increase their revenues by reselling existing 2D content.
- Our proposed 2D to 3D video conversion method can also be embedded in receivers (TVs or receiver boxes) so that the conversion of live 2D content to 3D format is carried in real-time. This has the advantage that it allows the network providers to broadcast in 2D while the viewer is able to watch the same content in 3D format without adding any burden to the network or additional cost during capturing.

- The other application of our 2D to 3D video conversion schemes is on playback devices such as DVD, Blu-ray and set-top boxes. The implementation of our algorithms in these devices will allow home viewers to watch 2D video in 3D format.
- Current consumer 3D video cameras which utilize bundled dual lenses do not allow users to zoom in/out while recording video [2]. Our proposed zoom correction algorithm can be embedded in consumer 3D cameras. This would enable users to have the zoom effect in their recording video. The proposed algorithm can also be used as a post-processing tool for correcting the 3D content captured by a stereo camera setup.
- Our proposed coding schemes for multiview video streams may be used for the transmission and storage of 3D content. Transmission of 3D content in the form of multiview is one of the major challenges of the 3D broadcasting system. Also, storing the huge amount of data, needed for representing multiview video streams, on Blu-ray discs or Personal Video Recorders is challenging, since the visual quality has to be kept at very high levels and there are always memory restrictions for both applications.

6.3 Contributions

The algorithms proposed in the preceding chapters address 3D broadcasting system problems from three aspects: 1) 3D content generation (by converting 2D video to 3D format), 2) enhancing the quality of captured 3D content, and 3) developing efficient video compression schemes for 3D TV applications.

- We designed a new and efficient method that estimates the depth map of a 2D video sequence using the existing H.264/AVC estimated motion information. Our proposed method modifies the motion information based on the characteristics of the human 3D visual perception. One advantage of the proposed approach is that it can be implemented in real-time at the receiver-end, without increasing the transmission bandwidth requirements. Performance evaluations showed that our method outperforms the other existing H.264 motion-based depth map estimation technique by providing better approximation for the scene's depth map and thus a better 3D visual effect.

- We further improved the quality and smoothness of the depth maps estimated by our above-mentioned method by identifying the objects in the scene and correcting the motion information accordingly. Performance evaluations showed that this approach results in higher quality and a smoother depth map compared to our previous approach.

- We investigated the effect of unsynchronized zoomed stereo videos on viewers through subjective tests. The results of our investigation motivated us to develop an effective algorithm for correcting the unsynchronized zoom effect in 3D videos. Our proposed scheme finds the matching points between the left and right views. This information is used to estimate the amount of scaling and translation needed to align the views and thus remove the vertical parallax due to unsynchronized zoom. Experimental results showed that our method produces videos with negligible scale difference and vertical parallax.

- We developed a new structure for coding multi-view camera sequences. The proposed scheme constructs an additional reference frame (besides the existing reference frames) to improve the inter-view prediction in the H.264/MVC standard. This scheme then automatically re-sorts the reference frame list so as to prevent the use of extra bits for coding reference frame indices. Performance evaluations showed that the proposed scheme is effective in compressing multiview streams due to its enhanced inter-view prediction structure.

- We developed a new multiview video coding scheme which has merits in terms of coding efficiency and random-access delay, two key requirements of future interactive multiview systems. Performance evaluations show that the proposed scheme outperforms H.264/MVC in terms of compression efficiency as well as random-access delay.

6.4 Suggestions for Future Research

The success of the 3D technology and the speed at which it will penetrate the entertainment market depend on how well the SMPTE and MPEG Working Groups will synchronize the standardization efforts of the three key components: 1) 3D content generation, 2) coding and transmission and 3) playback. Although significant work has been done in recent years regarding each of these components, the resulting findings have only managed to expose the challenges that lay ahead. The proposed methods in this thesis provide solutions to some of these challenges, but there are several other ideas that can be further explored. The following subsections addresse some of these ideas.

6.4.1 3D content recording

The availability of a wide variety of 3D content is one of the major requirements for the successful introduction of 3D TV to the consumer market. Although there are several studies regarding the guidelines for recording suitable 3D content for viewers [3-5], in practice 3D content generation is challenging, expensive and time consuming. More studies are required about the impacts of different camera setup parameters (baseline distance, focal length, and etc.) on the quality of perceived 3D content. The appropriate camera parameters should be defined for scenes with different depth ranges. The type and resolution of the display device (theatre, HD, active or passive glasses, etc.) should also be considered when choosing these parameters. Eventually these parameters can be added as default settings on 3D cameras to facilitate 3D video capturing.

6.4.2 3D video quality metrics

Another key factor for the successful penetration of 3D technology to the consumer market is to ensure that the new experience is superior to the one presently offered to the consumers. Assessing 3D quality is a huge challenge on its own. It seems that perceived user experience is psychological in nature and viewing 3D content introduces a new dimension of different environmental and display conditions. Therefore, new techniques are needed to assess this kind of experience. Currently there is no objective metric for measuring the quality of 3D content. Even the subjective test standards are not fully dedicated to the evaluation of 3D content quality. More study and research are required for defining the quality of experience in 3D and for developing a 3D quality measure metric.

6.4.3 2D to 3D video conversion using multiple monocular cues

In chapters 2 and 3 we proposed low-complexity real-time depth estimation techniques which utilize the motion parallax depth cue. As described in chapter 1, the human visual system, in addition to binocular parallax, utilizes several monocular depth cues to distinguish the distance between objects. Thus, it is expected that developing algorithms that properly integrate several monocular depth cues would enhance the quality of the estimated depth map and eventually that of the generated 3D content.

In recent years, machine learning has been receiving increasing attention in depth-map estimation application [6-9]. This is more evident in the area of 2D to 3D video conversion where supervised learning appears highly advantageous. The existing machine learning-based depth estimation algorithms are either semi-automatic [6, 7] or have been designed for images and not video (in a sense that they do not utilize motion parallax) [8, 9]. More study and research are required for developing fully automatic machine learning-based algorithms that integrate many monocular depth cues for estimating high quality depth map streams for 2D videos. As expected, such approaches may have to be used offline due to their high complexity.

6.5 References

[1] I. Ideses, L.P. Yaroslavsky, and B. Fishbain, "Real-time 2D to 3D video conversion," Journal of Real-Time Image Processing, Vol. 2, no. 1, pp. 3-9, 2007.

[2] http://www.fujifilm.com/products/3d/camera/finepix_real3dw1/

[3] L. M. J. Meesters, W. A. IJsselsteijn, and P. J. H. Seuntiens, "A survey of perceptual evaluations and requirements of three-dimensional TV", IEEE Trans. Circuits Syst. Video Technol. 14, 381–391 (2004).

[4] B. Mendiburu, "3D Movie Making: Stereoscopic Digital Cinema from Script to Screen," Focal Press, April 22, 2009.

[5] L. Lipton, "Stereographics developer's handbook," StereoGraphics Corporation, 1997.

[6] M.T. Pourazad, A. Bashashati, P. Nasiopoulos, and R. K. Ward, "Rapid Conversion of 2D Video to 3D Format Using Random Forests," Electronic Visualization and the Arts (EVA), 2010.

[7] P. Harman, J. Flack, S. Fox, and M. Dowley, "Rapid 2D to 3D Conversion," In Proceedings of SPIE, vol. 4660, pp. 78–86, 2002.

[8] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," In NIPS 18, 2005.

[9] A. Torralba, and A. Oliva, "Depth Estimation from Image Structure," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 9, pp. 1226–1238, 2002.

APPENDIX A: LIST OF RECENT PUBLICATIONS

Journals

- Pourazad, M.T., Bashashati, A., Nasiopoulos P., and Ward, R. K. (2010) Random Forests-based approach for 2D to 3D video conversion (to be submitted)
- Pourazad, M.T., Doutre, C., Nasiopoulos, P., Tourapis, A., and Ward, R.K. (2010) Unsynchronized zoom correction in 3D video (to be submitted).
- Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (April 2010) Efficient inter-view prediction structures for multiview coding. IEEE Transactions on Circuits and Systems for Video Technology (submitted).
- Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (Mar. 2010) Generating the Depth Map from the Motion Information of H.264-Encoded 2D Video Sequence. EURASIP Journal on Image and Video Processing (accepted).
- Pourazad, M.T., Nasiopoulos, P., and Ward,R.K. (May 2009) An H.264-based Scheme for 2D to 3D Video Conversion. IEEE Transactions on Consumer Electronic, vol. 55, no. 2, pp 742-748.

Conference Proceedings

- Pourazad, M.T., Bashashati, A., Nasiopoulos P., and Ward, R. K. (Sep. 2010) Random Forests-based approach for 2D to 3D video conversion, ECCV 2010 Workshop on Reconstruction and Modelling of Large-Scale 3D Virtual Environments, Crete, Greece (submitted).
- Doutre, C., Pourazad, M.T., Tourapis, A., Nasiopoulos, P., and Ward, R. K. (May 2010) Correcting Unsynchronized Zoom in 3D Video, IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France.
- Pourazad, M.T., Bashashati, A., Nasiopoulos P., and Ward, R. K. (Apr. 2010) Rapid Conversion of 2D Video to 3D Format Using Random Forests, Electronic Visualization and the Arts (EVA), Florence, Italy.
- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Jan. 2010) Conversion of H.264-encoded 2D video to 3D format, IEEE Conference on Consumer Electronics, Las Vegas, USA.

- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Nov. 2009) An efficient low random access delay panorama-based multiview video coding scheme, IEEE International Conference on Image Processing, Cairo, Egypt.
- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Jul. 2009) A new prediction structure for multiview video coding, 16th IEEE International Conference on Digital Signal Processing, 1-5 pages, Santorini, Greece.
- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Jan. 2009) Converting H.264-derived motion information into depth map, Proc.15th International Multimedia Modeling Conference, Lecture Notes in Computer Science 5371 Springer 2009, pp. 108-118, Sophia-Antipolis, France.
- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Jan. 2009) An H.264-based scheme for 2D to 3D video conversion, IEEE Conference on Consumer Electronics, Las Vegas, USA.
- Pourazad, M.T., Nasiopoulos P., and Ward, R. K. (Sep. 2006) An H.264-based video encoding scheme for 3D TV, European Signal Processing Conference (EUSIPCO), Florence, Italy.