

MULTI-VIEW IMAGE SYNTHESIS TECHNIQUES FOR 3D VISION AND FREE-VIEWPOINT APPLICATIONS

A DISSERTATION SUBMITTED TO
THE CENTRE FOR EXCELLENCE IN SIGNAL AND IMAGE PROCESSING,
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING
AND THE COMMITTEE FOR POSTGRADUATE STUDIES
OF THE UNIVERSITY OF STRATHCLYDE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

By
Nurulfajar Abd Manap
June 2012

© Copyright 2012
by
Nurulfajar Abd Manap

The copyright of this thesis belongs to the author under the terms of the United Kingdom copyright Acts as qualified by University of Strathclyde Regulation 3.49. Due acknowledgment must always be made of the use of any material contained in, or derived from this thesis.

Declaration

I declare that this Thesis embodies my own research work and composed by myself. Where appropriate, I have made acknowledgement to the work of others.

Nurulfajar Abd Manap

Dedication



"So, verily, with every difficulty, there is relief. Verily, with every difficulty there is relief." (94: 5-6)

To my wife, Anis Suhaila Mohd Zain – for her love and strength,
my beautiful princess, Nur Najla Alia – the ‘nur’ of my life,
my dearest parents & family – for their endless *dua*,

And my late son, Amir Yusuf – who has taught me the true meaning of patience,
humble, perseverance, life and hope.

“With you, it wasn’t long enough together, but it was long enough to last forever.”
(27/06/08 – 06/03/09)

Acknowledgements

In the name of Allah the Most Gracious and Most Merciful; many thanks to Him as He gave me the strength, good health, endurance and aptitude to complete this thesis and to experience this 'emotional journey'; and Peace and Prayers upon His final Prophet and Messenger.

During the years I was working on this thesis, I have been assisted and inspired by many people, and it is a pleasure for me to have this opportunity to express my heartfelt gratitude to them.

First and foremost, I would like to express my sincere gratitude to my advisor, **Professor John Soraghan**, for sharing his knowledge of image and video processing, for his kindness, invaluable guidance, and motivation throughout the period of my study. He has always been there to encourage and support me in the most positive ways. I am also very grateful to him for his deep commitment to his profession. He has taught me many things, and this work would not have been possible without his patience and guidance. I appreciate him greatly, and I am blessed and honoured to be his student.

My sincere appreciation also goes to Dr Lykourgos Petropoulakis and Dr Akbar Sheikh Akbari for their insight evaluations, constructive criticisms and invaluable recommendations throughout this thesis.

I would also like to express my thanks to the Ministry of Higher Education, Malaysia and Universiti Teknikal Malaysia Melaka. This PhD research would not have been materialised without generous financial support offered by both organisations.

My appreciation also goes to all academics, supporting staff and friends at Centre for Excellence in Signal and Image Processing (CeSIP) group, Electronic and Electrical Engineering Department, University of Strathclyde. Many thanks to those who helped me in my research works particularly to Dr Muhammad Asif, a friend and lab mate during my first two years of PhD. Special thanks go to Gaetano Di Caterina, who has constantly been helpful and resourceful in the coding and implementation of image and video processing algorithms. I would also like to thank Adel Daas, Sameer, Masrul, Khusairy, Haida, Sheriff, Suhaida and all my friends at CeSIP for providing a rewarding

and exciting research environment, that I have enjoyed being a part of.

I am also indebted to my ‘family’ in Scotland: Dr Khairul Azmi & Sister Asmaheram, Dr Zuraeda & Brother Rosli, Dr Shahabuddin & Sister Ratifah, Dr Che’ Anuar & Dr Rosazra, Ustaz Abduh & Ustazah Safinar, Brother Ariffin & Sister Asma’, and Azrik for their continuous moral support during my years in Glasgow. Not forgetting, many thanks to all my friends, especially Badrul Hisham, Nik Fahusnaza and Zulfadli; for wonderful friendship, motivation and encouragement.

I am extremely grateful to all my family members; my father, Abd Manap Arsat, my mother, Jusiah Sukor; my in-laws: Mohd Zain Saat and Ramlah @ Ruby Daud; and all my siblings. Without their prayers, love, kindness, sensibility, devotion, support and patience I could have never reached this point of my life.

My special thanks to my beloved wife: Anis Suhaila Mohd Zain for her love and extreme patience. She has been and continues to be a constant source of inspiration, motivation and strength. She stood by me in every way and I am eternally grateful to have her as my better half. It has been a meaningful journey for both of us; a journey filled with happiness and challenges. May Allah grant all her wishes and give her more beautiful days ahead, in return.

Nur Najla Alia is the ‘Nur’ of my life, the simplest love I have known, and the purest one I owned. I would also like to thank my beautiful girl for her love, patience and sacrifices and may Allah bless her with happiness forever. And to my small fighter, Amir Yusuf: *“You taught me about life and hope. Being with you is the greatest lesson in my life. I miss you always.”*

Last but not least, to all those who have significantly contributed directly and indirectly towards the completion of this thesis; I am sincerely most grateful to all. ‘Alhamdulillah’, thank you again to Almighty Allah for giving me ‘everything’ for me to be where I am now.

*“It is good to have an end to journey towards;
but it is the journey that matters in the end.” – Ursula Le Guin.*

Abstract

The depth information in a scene for a stereo image is used in many image analysis and 3D video processing applications. Novel view synthesis draws a significant research interest because it can drive future 3DTV and free viewpoint video applications, which allows a viewer to perceive 3D depth scenery without wearing any special glasses. The main objective of the multi-view video system is to create another dimension to the viewer and provide 3D information such as depth. This thesis describes new approaches and methods for stereo matching and inter-view synthesis algorithms with application for 3D and free-viewpoint. A Depth Image Layers Separation (DILS) algorithm is proposed to efficiently synthesize the inter-view image based on layered disparity depth map representation through stereo matching and inter-view interpolation. The main idea of this approach is to separate the depth map into several layers of depth based on the disparity distance of the corresponding points. This technique is used to synthesize novel inter-view images based on disparity depth map layers representation. Simulation results show that the concept of depth layers separation is able to create inter-view images and can be integrated with other technique such as the disparity depth refinement and occlusion handlings processes. The DILS algorithm can be performed from a simple to sophisticated stereo matching techniques to synthesize the inter-view images. This technique leads to the second novelty method, Depth Layer Refinement (DLR) that uses the disparity depth layers to refine the disparity map. The main aim of this algorithm is to improve the raw disparity maps in the disparity refinement stage with a basic similarity metric of SAD in the stereo matching algorithm. The edge boundaries and discontinuities region are significantly improved with the proposed techniques compared to the state-of-the-art stereo matching algorithms. The third novelty proposed in the multi-view camera applications known as Multi-Level View Synthesis (MLVS). In this technique, the multi-view synthesis created based on a limited number of cameras to create dense images. The new structures and design are shown to offer improved performance and provide additional views with fewer cameras arrangement compared to the conventional high volume camera configurations for free-viewpoint video acquisition.

List of Abbreviations

3D	Three Dimensional
3DTV	Three Dimensional Television
3DV	Three Dimensional Video
AD	Absolute intensity Differences
ATTEST	Advanced Three-Dimensional Television System Technologies
AVC	Advanced Video Coding
BM	Bi-directional Matching
BMP	Bad Matching Pixels
BP	Belief Propagation
CCD	Charge-Coupled Device
CGI	Computer Graphic Images
CIF	Common Intermediate Format
DIBR	Depth Image Based Rendering
DILS	Depth Image Layers Separation
DLR	Depth Layer Refinement
DP	Dynamic Programming
DSI	Disparity Space Image
EDISON	Edge Detection and Image Segmentation
fps	frame per second
FTV	Free-viewpoint Television
FW	Fixed Window
FVV	Free-Viewpoint Video
GCA	Group of Camera Array
GOP	Group of Pictures
GOV	Group of Views
IC	Intensity Consistent
IBR	Image Based Rendering
IBMR	Image Based Modelling and Rendering
IEC	International Electrotechnical Commission

ISO	International Organization for Standardization
JMVM	Joint Multi-view Video Model
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
LC	Locally Consistent
LCD	Liquid Crystal Display
LDI	Layered Depth Image
LI	Linear Interpolation
LR	Left-to-Right
LRCC	Left-Right Consistency Check
MAD	Mean Absolute Differences
ME	Motion Estimation
MI	Mutual Information
MLVS	Multi-Level View Synthesis
MMRG	METU (Middle East Technical University) Multimedia Research Group
MOP	Matrix of Pictures
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MVC	Multi-view Video Coding
MVI	Multi-View Imaging
NCC	Normalized Cross Correlation
PSNR	Peak Signal to Noise Ratio
PTZ	Pan-Tilt-Zoom
RAM	Random-Access Memory
RANSAC	Random Sample Consensus
RMS	Root Mean Squared
SAD	Sum of Absolute Differences
SD	Square intensity Differences
SHD	Sum of Hamming Differences
SIFT	Scale-Invariant Feature Transform
SIMD	Single Instruction Multiple Data
SLAM	Simultaneous Localization Automatic Mapping
SURF	Speeded-Up Robust Features
SMP	Single Matching Phase

SNR	Signal-to-Noise-Ratio
SSD	Sum of Squared Differences
SSIM	Structural SIMilarity
TOF	Time of Flight
UL	Upper-to-Lower
ULCC	Upper-Lower Consistency Check
WTA	Winner-Take-All
voxel	volume elements

Table of Contents

DECLARATION	III
DEDICATION	IV
ACKNOWLEDGEMENTS	V
ABSTRACT	VII
LIST OF ABBREVIATIONS	VIII
TABLE OF CONTENTS	XI
LIST OF FIGURES	XV
LIST OF TABLES	XX
CHAPTER 1	
INTRODUCTION	1
1.1 Preface	1
1.2 Research Motivations	3
1.3 Summary of Original Contributions	6
1.4 Thesis Organization	9
CHAPTER 2	
FUNDAMENTALS OF MULTI-VIEW IMAGING AND 3D VIDEO	12
2.1 Introduction	12
2.2 Stereo Vision to Multi-view	13
2.3 Camera Projective to Two-View Geometry	16
2.3.1 <i>Homography</i>	19
2.3.2 <i>Two-View Geometry</i>	20
2.3.3 <i>Triangulation</i>	23
2.4 Applications of Multi-view Imaging	24
2.4.1 <i>Stereoscopic Displays</i>	24
2.4.2 <i>Free-Viewpoint Video</i>	26
2.4.3 <i>Video Editing and Special Effects</i>	28
2.5 3D Video System	29
2.5.1 <i>Camera Acquisition, Calibration and Rectification</i>	30
2.5.2 <i>Stereo Correspondence and Depth Map</i>	32
2.5.3 <i>Multi-view Compression</i>	33
2.5.4 <i>Rendering for 3D</i>	33
2.6 Multi-view Video Coding Algorithms	34
2.6.1 <i>Conventional Stereo Video Coding</i>	34
2.6.2 <i>Video Plus Depth Data</i>	36

2.6.3	<i>Multi-view Video Coding</i>	37
2.6.4	<i>MVC Test and Analysis</i>	39
2.7	Conclusion	41
CHAPTER 3		
STEREO MATCHING AND VIEW SYNTHESIS ALGORITHMS		43
3.1	Introduction	43
3.2	Stereo Matching Algorithms	44
3.2.1	<i>Matching Cost Computation and Aggregation</i>	47
3.2.2	<i>Disparity Computation and Optimization</i>	52
3.2.3	<i>Disparity Refinement</i>	54
3.2.4	<i>Summary of Stereo Matching Algorithms</i>	56
3.3	3D Scene Representation	59
3.4	View Synthesis Algorithms	64
3.5	Image Based Rendering	67
3.5.1	<i>Rendering With Explicit Geometry</i>	68
3.5.2	<i>Rendering With Implicit Geometry</i>	69
3.5.3	<i>Rendering Without Geometry</i>	69
3.6	Layered Image Based Rendering	71
3.7	Performance Evaluation	74
3.7.1	<i>Peak Signal-to-Noise Ratio (PSNR)</i>	74
3.7.2	<i>Structural SIMilarity (SSIM) Index</i>	75
3.8	Conclusion	78
CHAPTER 4		
VIRTUAL VIEW SYNTHESIS BASED ON DEPTH IMAGE LAYERS SEPARATION (DILS)		80
4.1	Introduction	80
4.2	Depth Map Layers Representation	81
4.2.1	<i>Depth Map</i>	81
4.2.2	<i>Layers Based Disparity Depth Map Separation</i>	84
4.2.3	<i>Inter-view Synthesis Based on Disparity Depth Map</i>	84
4.3	System Design Architecture	86
4.3.1	<i>Stereo Matching Engine</i>	87
4.3.1.1	<i>Stereo Disparity Estimation</i>	88
4.3.1.2	<i>Stereo Disparity Refinement</i>	89
4.3.2	<i>Inter-View Synthesis with DILS</i>	90
4.3.2.1	<i>Histogram Distribution</i>	92
4.3.2.2	<i>Layers Identification</i>	94
4.3.2.3	<i>Layers Separation</i>	100
4.3.2.4	<i>Image Translation (Left)</i>	102
4.3.2.5	<i>Mask Layers</i>	103
4.3.2.6	<i>Synthesis: Intermediate View Interpolation</i>	103
4.3.2.7	<i>Image Translation (Right)</i>	104

4.3.2.8	<i>View Synthesis</i>	105
4.3.2.9	<i>Hole-Filling</i>	105
4.4	Results and Discussion	106
4.4.1	<i>Performance Evaluation of Conventional Linear Interpolation and DILS</i>	107
4.4.2	<i>Performance Evaluation Based on AD-Census and SAD Disparity Depth Map</i>	114
4.5	Conclusion	117
CHAPTER 5		
DEPTH LAYER REFINEMENT (DLR) ALGORITHM FOR DISPARITY DEPTH MAP		118
5.1	Introduction	118
5.2	Overview of System Design	119
5.3	Disparity Layer Refinement	121
5.3.1	<i>Stereo Matching and Layers Extraction</i>	123
5.3.2	<i>Boundaries and Edges Identification</i>	124
5.3.3	<i>Morphological Process</i>	127
5.3.4	<i>Layers Composition</i>	130
5.4	Performance Evaluation	131
5.4.1	<i>Quality Metric</i>	131
5.5	Results and Discussion	132
5.5.1	<i>Performance Evaluation Based on Different Similarity Metric</i>	132
5.5.2	<i>Performance Based on Middlebury Stereo Evaluation</i>	135
5.6	Conclusion	140
CHAPTER 6		
MULTI-LEVEL VIEW SYNTHESIS (MLVS) BASED ON DILS ALGORITHM FOR MULTI-CAMERA ARRAY		142
6.1	Introduction	142
6.2	Multi-Camera Array Configuration and Applications	143
6.3	System Design Architecture	146
6.3.1	<i>Multi-Level View Synthesis Algorithm</i>	148
6.3.2	<i>Multi-Camera Array Datasets</i>	153
6.4	Experimental Results	154
6.5	Cookies Datasets Analysis	156
6.5.1	<i>MLVS: Level 1</i>	156
6.5.2	<i>MLVS: Level 2</i>	160
6.5.3	<i>MLVS: Level 3</i>	163
6.6	Lego Datasets Analysis	166
6.6.1	<i>MLVS Level 1</i>	167
6.6.2	<i>MLVS Level 2</i>	169
6.6.3	<i>MLVS Level 3</i>	172
6.7	Results Evaluation with Different Baseline Images	174

6.7.1	<i>MLVS Level 1</i>	174
6.7.2	<i>MLVS Level 2</i>	176
6.7.3	<i>MLVS Level 3</i>	178
6.8	Conclusion	180
CHAPTER 7		
CONCLUSION AND SUMMARY		182
7.1	Conclusions Overview	182
7.2	Future Work	185
APPENDIX A: CAMERA CALIBRATION		187
APPENDIX B: PREDICTIVE CODING		194
APPENDIX C: MIDDLEBURY STEREO EVALUATION		200
AUTHOR'S PUBLICATION		203
REFERENCES		205

List of Figures

Figure 1.1: Overview of the proposed multi-view depth synthesis system that includes the acquisition of 3D video, calibration and rectification, stereo matching, view synthesis, layer based rendering, multi-view synthesis and compression sub-system	9
Figure 2.1: Capturing the plenoptic function from the still image camera to the video camera or multi-view imaging systems [3].....	14
Figure 2.2: Pinhole camera model [44].....	17
Figure 2.3: Rearranged pinhole camera model [44, 45].....	18
Figure 2.4: Converting from object to camera coordinate systems [37, 44].....	19
Figure 2.5: Epipolar geometry [37].....	21
Figure 2.6: Two-camera views based on pinhole camera model [46].....	23
Figure 2.7: Stereoscopic lenticular display [15]. (a) A lenticular display sheet precisely positioned onto an LCD. (b) Multi-view lenticular display with three pixels/views covered by micro-lens	25
Figure 2.8: Overview of stereo video system [56].....	30
Figure 2.9: Rectification of stereo camera to standard form [37].....	31
Figure 2.10: Prediction in H.262/MPEG 2 video multi-view profile [68].....	35
Figure 2.11: 3D data representation format consisting of regular 2D colour video and 8-bit depth images [43].....	36
Figure 2.12: Temporal/inter-view prediction structure for MVC [11].....	37
Figure 2.13: Matrix of Pictures (MOP) for $N=4$ image sequences, each comprising $K=4$ temporally successive pictures [11]	38
Figure 3.1: Classification of matching algorithms	46
Figure 3.2: The disparity estimated by searching the most similar block along the horizontal epipolar line [37]	48
Figure 3.3: Scene representation categories [136]	60
Figure 3.4: Light field image based representation [78]: (a) One parameterization of the light field. (b) A sample light field image array	63
Figure 3.5: A set of related pixels measured by a patch s on surface S [49].....	65
Figure 3.6: Virtual view synthesis [158].....	67
Figure 3.7: Spectrum of rendering representations [78].....	68
Figure 3.8: Sample of results comparison on PSNR and SSIM between the original image, images with noise and sharpened. The respective SSIM maps obtained and evaluated based on the standard parameters proposed by Wang [166]......	77
Figure 4.1: Aligned stereo rig and known correspondence [44].....	82
Figure 4.2: Relationship of depth and disparity [44].....	83

Figure 4.3: Disparity range and parallel planes (layers). (a) Disparity is higher for points closer to the camera. (b) Different disparity levels for disparity map	83
Figure 4.4: Inter-view synthesis: (a) The virtual camera view placed between camera 1 and 2. (b) Geometric stereoscopic camera model [167]	84
Figure 4.5: Block diagram of the proposed novel view synthesis based on depth map layers representation	86
Figure 4.6: Matching costs computation based on window size, $n \times n$, and disparity range, d , with left image as the reference and right as the target image	88
Figure 4.7: Occlusions in the left and right image.	90
Figure 4.8: Depth Image Layers Separation (DILS) algorithm on the view synthesis module.	91
Figure 4.9: Disparity depth map and the histogram distribution based on the ground truth image: (a) Disparity depth map and its corresponding histogram for Tsukuba image. (b) Disparity depth map and its corresponding histogram for Map image.	93
Figure 4.10: Histogram distribution of the disparity depth map and the matched pixels in binary after the quantization	95
Figure 4.11: Identify the layer on the threshold data samples from index k to d_{max}	96
Figure 4.12: Finding the i_F and i_L to distinguish the layer i in the threshold data sample p'	97
Figure 4.13: The zero run-length algorithm to determine the non-linear segment layer d into D layers	98
Figure 4.14: New zero run-length algorithm to determine the non-linear segment layer d into D layers for the data stream without single '0' occurrence	100
Figure 4.15: Example of layer masks based on layer separation process	101
Figure 4.16: Translation process for right image to the left by d value	102
Figure 4.17: Sample of inter-view interpolation at different layers.	104
Figure 4.18: The translation process to the right based on the disparity range value	105
Figure 4.19: Holes in the final virtual view images.	106
Figure 4.20: Middlebury data sets for the left image	106
Figure 4.21: Disparity depth maps for (from top to bottom) 'Teddy', 'Venus', 'Tsukuba' and 'Cones' image pairs based on the (a) Ground truth, (b) Left-to-right consistency check (LRCC) disparity maps and (c) Filtered LRCC disparity maps using 11x11 median filter	109
Figure 4.22: Histogram distribution of the disparity depth map using the LRCC. (a) Teddy; (b) Venus; (c) Tsukuba; (d) Cones.....	110
Figure 4.23: Original image view of the data sets at camera baseline ratio, $\beta=0.5$	111
Figure 4.24: Image view synthesis of Middlebury datasets at camera baseline ratio, $\beta=0.5$ obtained through (a) Conventional Linear Interpolation (LI) and (c) DILS algorithm; and SSIM map images respectively.	111
Figure 4.25: PSNR and SSIM index of Middlebury datasets based on LI and DILS algorithms	113
Figure 4.26: PSNR and SSIM index of Middlebury datasets based on DILS algorithm.....	114
Figure 4.27: Image view synthesis of Middlebury datasets at camera baseline ratio, $\beta=0.5$ through DILS algorithm based on disparity depth maps of (a) AD-Census and (c) Fixed Window (FW) SAD; and SSIM map images respectively.....	115

Figure 4.28: PSNR and SSIM index of Middlebury datasets through DILS based on based on AD-Census and FW-SAD disparity depth map	116
Figure 5.1: Overview DLR system.....	120
Figure 5.2: Block diagram of the proposed algorithm on disparity refinement based on DILS	122
Figure 5.3: Part 1 and 2 block diagram for the DLR that consist: a) stereo matching and layers extraction, and b) boundaries and edges identification	126
Figure 5.4: Boundaries and edge detection example.....	127
Figure 5.5: Part 3 block of the DLR algorithm, morphological process	128
Figure 5.6: Sample of boundary path for layer i and the disparity depth map	128
Figure 5.7: Mapping and diffusing for layer i with the border set by the segmented reference image	129
Figure 5.8: Layer extraction with edge map image. (a) Raw object mask layer i ; (b) Refined layer i through morphological process	130
Figure 5.9: Part 4 block of DLR algorithm, layers composition.....	131
Figure 5.10: Results of stereo matching based on different similarity metric	133
Figure 5.11: RMS error based on window size for all pixels and non-occluded pixels.....	134
Figure 5.12: Results of the proposed method by the Middlebury benchmark datasets. The first row images are the reference images of each set. The second row images are the ground truths. The third row images are the disparity maps by left-to-right matching using SAD metric. The fourth row images are the resulting disparity maps by DLR-SAD method	136
Figure 5.13: Analysis for non-occluded region based on bad pixel (absolute disparity error > 1). Non-occluded regions (white) with occluded and border regions (black)	139
Figure 5.14: Analysis for signed disparity error based on discontinuity. Regions near depth discontinuities (white), occluded and unknown regions (black) and other regions (gray)	140
Figure 6.1: Multi-camera array configuration by the Computer Graphic Laboratory, Stanford University	144
Figure 6.2: Sample of multi-camera configuration	145
Figure 6.3: System design architecture for MLVS algorithm	146
Figure 6.4: Image view synthesis module of MLVS algorithm.....	146
Figure 6.5: The image matching and synthesis based on the three different levels of MLVS..	147
Figure 6.6: Basic arrangement for the group of camera array, which consists of four images, obtained from four cameras views	148
Figure 6.7: The matching and synthesis for the Level 1 of MLVS.....	149
Figure 6.8: The matching and inter-view synthesis for the Level 2 of MLVS	150
Figure 6.9: Level 3 MLVS algorithm.....	152
Figure 6.10: The group of camera array can be expanded horizontally or vertically depending on the required number of virtual inter-view cameras	153
Figure 6.11: Expansion of group of camera array (GCA) with additional camera configuration along the horizontal lines	153
Figure 6.12: Sample of the first image camera from Stanford Multi-Camera Array	154

Figure 6.13: Multi-level view synthesis (MLVS) for Group of Camera Array (GCA) 1 in Cookies datasets	155
Figure 6.14: Cookies stereo image pair in the first row of the group camera array datasets. ...	157
Figure 6.15: The comparison of conventional interpolation and MLVS (Level 1) method. The sample of inter-view image synthesized at $\beta=0.4$ (camera 3) for the first row of Cookies datasets and the structural similarity index map.	158
Figure 6.16: PSNR results for the Level 1 of the MLVS using Left-to-Right (LR) matching DILS algorithm compared to the conventional inter-view interpolation for the first and fifth row of Cookies datasets.....	159
Figure 6.17: Comparison SSIM results for the Level 1 of the MLVS and the conventional inter-view interpolation for the first row of Cookies datasets.	159
Figure 6.18: Cookies stereo image pair in the first column of the group camera array datasets.	160
Figure 6.19: The comparison of conventional interpolation and MLVS (Level 2) method. The sample of inter-view image synthesized at $\beta=0.5$ (camera 3 in column 1) for the third row of Cookies datasets and the structural similarity index map.....	161
Figure 6.20: PSNR results for the Level 2 MLVS using Upper-to-Lower (UL) Matching DILS compared to the conventional linear interpolation PSNR of Cookies datasets.....	162
Figure 6.21: SSIM results of Level 2 MLVS compared with the SSIM results of the conventional linear interpolation for the first row of Cookies datasets.	163
Figure 6.22: The inter-view image synthesis along the third row in the MLVS Level 3 for the Cookies multi-camera datasets (group camera array 1).	164
Figure 6.23: The comparison of the MLVS (Level 3) algorithm by using Left-to-Right (LR) and Upper-to-Lower (UL) matching approaches. The sample images are for the third row and third column (3, 3) of the Cookies multi-camera datasets.....	164
Figure 6.24: The comparison of PSNR for the Level 3 of the MLVS by using Upper-to-Lower (UL) and Left-to-Right (LR) Matching in the group of array 1.....	165
Figure 6.25: Results on PSNR and SSIM for the MLVS at Level 3 compared with the conventional linear interpolation. The data sampled for the third row of Cookies multi-camera datasets.....	165
Figure 6.26: Lego stereo image pair in the first row of the group camera array datasets.....	166
Figure 6.27: The comparison of MLVS (Level 1) without/with hole-filling algorithms. The sample of inter-view image and the structural similarity index map synthesized at $\beta=0.5$ (camera in row 1 column 3) for the Lego datasets.	167
Figure 6.28: PSNR and SSIM of MLVS (Level 1) without and with using hole-filling algorithm in the image view synthesis at row 1 and 5 for Lego multi-camera datasets.....	169
Figure 6.29: Lego stereo image pair in the first column of the group camera array datasets ...	170
Figure 6.30: The comparison of MLVS (Level 2) with and without using the hole-filling method. The sample of inter-view image and the structural similarity index map synthesized at $\beta=0.5$ (camera in row 3 of column 1) for the Lego datasets.....	171
Figure 6.31: PSNR and SSIM of MLVS (Level 2) without and with hole-fillings in the image view synthesis at column 1 and 5 for Lego multi-camera datasets.....	172
Figure 6.32: The comparison of MLVS (Level 3) with and without hole-fillings method. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 7 of column 3) for the Lego datasets.....	173

Figure 6.33: PSNR and SSIM of MLVS (Level 3) without and with using hole-filling algorithm in the image view synthesis at row 7 and 11 for Lego multi-camera datasets.....	174
Figure 6.34: The comparison of the MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 9 of column 3) for the Lego datasets.	175
Figure 6.35: Comparison PSNR and SSIM results of the MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching in the image view synthesis at rows 1, 9 and 17 for Lego multi-camera datasets.	176
Figure 6.36: The comparison of the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 7 of column 1) for the Lego datasets.	177
Figure 6.37: PSNR and SSIM results for the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching at columns 1, 9 and 17 for Lego multi-camera datasets.	178
Figure 6.38: The comparison of the MLVS (Level 3) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 11 of column 7) for the Lego datasets.	179
Figure 6.39: PSNR and SSIM results of the MLVS (Level 3) by using 1-to-5 and 1-to-9 Left-Right (LR) baselines matching in the image view synthesis at row 7 and 11 for Lego multi-camera datasets.	180
Figure A.1: Pixel coordination system.....	187
Figure A.2: Radial distortion that causes straight lines to be bended	190
Figure A.3: Tangential camera lens distortion	191
Figure A.4: Converting from object to camera coordinate systems.....	191
Figure A.5: Rotating points by θ	192
Figure B.1: Hierarchical encoding of a matrix of pictures (MOP) with bi-predictive pictures	194
Figure B.2: Hierarchical reference picture structure for temporal prediction.....	195
Figure B.3: Temporal prediction using hierarchical B pictures.....	196
Figure B.4: IPP inter-view prediction for key pictures	196
Figure B.5: Reference modes in H.264/AVC multi-view extension codec	198
Figure B.6: Multi-view video coding parameter configuration file	199
Figure C.1: Middlebury stereo image pairs dataset for left images with their corresponding ground truth disparity maps [22].....	200
Figure C.2: Sample of Middlebury Stereo Evaluation Results, where ‘nonocc’ (for non-occluded regions), ‘all’ (for all regions) and ‘disc’ (for discontinuities regions).....	201

List of Tables

Table 2.1: Input Parameter of the H.264/AVC codec	40
Table 2.2: Simulation results for simulcast coding	40
Table 2.3: Simulation results for the multi-view coding.....	40
Table 3.1: Stereo Matching Algorithm Components	45
Table 3.2: Summary of Stereo Matching Algorithms	57
Table 3.3: Summary of Layered Image Techniques	73
Table 4.1: Sample of disparity range levels for different layers	96
Table 4.2: PSNR results of inter-view synthesis images based on conventional linear interpolation (LI) and DILS algorithms	112
Table 4.3: SSIM index results of inter-view synthesis images based on conventional linear interpolation (LI) and DILS algorithms	112
Table 4.4: PSNR results of inter-view synthesis images based on AD-Census and FW-SAD disparity depth map	115
Table 4.5: SSIM index results of inter-view synthesis images based on AD-Census and FW-SAD disparity depth map	115
Table 5.1: Processing time and RMS of Tsukuba and Map images.....	135
Table 5.2: Middlebury dataset ranking with the 1 pixel threshold. These values indicate the percentage of bad pixels whose errors are more than 1 pixel, where ‘N-o’ (non-occluded regions), ‘All’ (for the all regions), ‘Disc’ (near depth discontinuities regions) and ‘Avg.’ (average percentage of bad pixels over all datasets).	138
Table 6.1: Parameter of Cookies and Lego datasets for stereo matching and synthesis in DILS algorithm	155
Table 6.2: Comparison results for the conventional Linear Interpolation (LI) and the MLVS method (for Level 1) along the row 1 and 5 in a group of camera array.	159
Table 6.3: Results comparison between conventional Linear Interpolation (LI) and MLVS method (for level 2) along the column 1 in a group of camera array.....	162
Table 6.4: Comparison results between the use of Left-to-Right (LR) matching and the use of the Upper-to-Lower (UL) matching in the MLVS method (Level 3) for the row 3 in a group of camera array.	164
Table 6.5: Comparison results of MLVS (Level 1) with and without using the hole-filling algorithms along the row 1 and 5 in a group of camera array.....	168
Table 6.6: Results comparison of MLVS (Level 2) with and without using the hole-filling methods along the column 1 and 5 in a group of camera array.	171
Table 6.7: Results comparison of MLVS (Level 3) with and without the hole-filling methods along the row 7 and 11 in a group of camera array.....	173
Table 6.8: Comparison results of MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching along the rows 1 and 9	175

Table 6.9: Comparison results of the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching along the columns 1 and 9	177
Table 6.10: Results comparison of the MLVS (Level 3) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching along the row 7 and 11	179

Chapter 1

Introduction

1.1 Preface

Three-Dimensional (3D) video and imaging technologies is an emerging trend in the development of digital video systems. It has witnessed the appearance of 3D displays, coding systems and 3D camera setups by many global research groups [1]. Three-dimensional multi-view video is typically obtained from a set of synchronized cameras, which are capturing the same scene from different viewpoints. This technique enables applications such as free viewpoint video or 3DTV. Free viewpoint video applications provide the ability to the user to select any viewpoint in the video scene interactively. A 3D scene is obtained if the data representation and display enable to distinguish the depth within the scene. With 3DTV, the depth of the scene can be perceived using a multi-view display that renders simultaneously several views of the same scene through the special 3D glasses transmitter. There is a demand for 3D vision and multi-view application from small applications such as stereo-video acquisition, robotic navigation and video surveillance to highly sophisticated systems in entertainment and post-production works in video games and film industry. With a large amount of data, issues such as complexity, reliability and usability have become very important. In order to deal with this growing demand, research and development has been carried out in academic and commercial environments to find improvements or new solutions in signal processing, communications, computer vision and system engineering.

The numbers of cameras required are quite high to create dense immersive multi-view application. Therefore, an efficient transmission and compression is necessary to render multiple views on a remote display. However, one major problem of multi-view video is the large amount of data to be compressed, decompressed and rendered. The obstacles can be overcome with an efficient and flexible multi-view video system. The inter-view image synthesis can reduce the complexity of multi-camera configuration and number

of cameras through composing virtual views between the camera viewpoints. The algorithm developed for acquiring the depth signal from a multi-view video should be capable to obtain satisfactory and reliable new virtual synthesis images with a small number of cameras in the multi-view system. Research in academia is more focused to explore and improve multimedia signal processing tasks by generating more accurate, and robust algorithms in multi-view capturing, 3D representation, multi-view video compression and transmission, image-based rendering, multi-view image synthesis and multi-view display [2]. Additionally, research carried out is based on the study of exploiting structure in multi-view imaging system [3]. The creation of a high-quality reconstruction based on real-world scenes from a sparse set of multi-view video streams is also an active area of research [4, 5]. Realistic rendering for dynamic shape and motion, as well as the dynamic appearance and material properties of a real-world scene, are challenging engineering and algorithmic problems.

The 3D visual content representation is one of the fundamental challenges in the area of 3D signal processing. Various representations of 3D contents such as Lightfield [6], multi-view representation [7], 2D images with depth [8] and volumetric [9] requires particular and efficient compression techniques [10, 11]. The standard for the multi-view video compression developed by the Joint Video Team (JVT) in the Multi-view Video Coding (MVC) standard [12]. The description of the standard can be found in the Joint Draft 8.0 on Multi-view Video Coding [13]. The main challenge of the MVC standard is to define the efficient codec tools for the multi-view video due to the huge amount of data to be stored. As the multi-view video becoming the new generation of the interactive multimedia, it serves a wide variety of applications. The technologies and challenges for 3D video attracting much interest while it could provide not only a new viewing but also additional information such as depth. The development of signal processing algorithms for multi-view displays is also an active research field [14, 15], such as in the new 3D displays offer viewing of high-resolution stereoscopic images without glasses and volumetric displays [16] that provide new viewing experience to viewers. Volumetric displays produce volume-filling 3D imagery, where each volume element or voxel in a 3D scene emits visible light from the region in which it appears. Given their ability to project volume-filling autostereoscopic imagery, these displays are being adopted in fields as diverse as medical imaging, mechanical computer-aided design and military visualization [17].

This thesis is intended to cover tasks of image processing for multi-view and free viewpoint video systems and will be equally beneficial to commercial and academic researchers involved in developing techniques for the efficient utilization of multi-view images and video data. It focuses on the stereo matching and virtual view synthesis with application to 3D and free-viewpoint video.

1.2 Research Motivations

Due to the reducing cost of digital cameras, multi-view imaging has attracted attention. This opens a wide variety of interesting new research topics and applications, such as virtual view synthesis, high performance imaging, environmental surveillance, industrial inspection, remote education, entertainment, 3DTV and free-viewpoint video. Some of these tasks can be handled with conventional single view video. However, using multiple views of the scene significantly broadens the field of applications and at the same time enhances the visual performance and user experience.

The 3D films become as a new trend in the market for the past few years. James Cameron's 'Avatar' set new benchmarks on how the future of 3D films would look like. The film industry became particularly interested the 3D format since it could return huge profits compared to the standard format, which results in most of the new animation and Computer Graphic Images (CGI) films represented in 3D format. In addition, multi-view imaging has gained popularity among filmmakers with the unique views displayed in the action captured by the multi-camera in the film like 'The Matrix'. The freeze-effect shown in the film provides a new sense of viewing. Multiple cameras offer additional information to the viewer and can be used in 3D video and free-viewpoint video.

The positive responses on the 3D film from the audience provide a platform for the manufacturer to introduce the 3DTV to the public. The demand for multi-view video coding and free-viewpoint video is also driven by the development in new 3D display technologies and the growing use of multi-camera arrays. A variety of companies is starting to produce 3D display technologies that do not require glasses and can be viewed by multiple people simultaneously. This technology provides a good platform for new applications to emerge such as 3D scene communication [18]. This new application will adapt the multi-view video as the next generation high performance

video. Furthermore, even with 2D displays, multi-camera arrays are increasingly being used to capture a scene from many angles. The resulting multi-view data sets allow the viewer to observe a scene from any viewpoint and serve as another application of multi-view video compression. However, depending on the system, the number of cameras is limited and it can only describe 3D scenes from specific visual angles.

Autostereoscopic displays provide a platform for the future 3D technology. These displays spatially multiplex many views onto a screen that will give immersive experience by enabling users to look around the virtual scene from different angles. A parallax barrier that physically occludes certain pixels or a lenticular sheet that distributes light in different directions is fixed to the screen to ensure that each eye perceives pixels from two different views without the need of wearing 3D glasses [19]. For 3DTV and free-viewpoint video to become practical and acceptable on a wide scale, the added realism must outweigh any required increases in processing and system complexity. The stereoscopic information must be comfortable to view. Both of these goals can be achieved if the inter-views of the scene are available.

Multi-view screens commonly display a small number of images, such as nine views in Philips multi-view display [20] and it will increase with the development in the display technology. However, the main task for multi-view imagery is to capture content from many cameras. The camera configuration and density (number of cameras) impose practical limitations on navigation and quality of rendered views at a certain virtual position. Therefore, there is a classical trade-off to consider between costs and quality. Generally, the denser capturing of multi-view images with a larger number of cameras provides a more precise 3D representation resulting in higher quality views through the rendering and display processes. However, it also requires a higher compression rate in the coding process. The multi-camera arrays configuration is complicated and expensive, which requires bulky equipment, well-planned setup, camera calibration and rectification. Capturing imagery from two cameras for instance is much simpler and can be done with a low cost implementation than multiple cameras. Thus, the complexity of multi-camera configuration can be reduced if we can generate many views from a stereo image pair. This is the goal of view synthesis. The intermediate view synthesis composes an image that locates in the virtual viewpoint between source image viewpoints [21].

Binocular stereo is one of the most significant and active areas in the field of computer vision. Existing intermediate view synthesis algorithms emphasize mostly in disparity estimation [8] in the stereo matching algorithms. Recently, the number of publications on stereo is increasing due to the Middlebury Stereo Vision Page by Scharstein and Szelinski [22] with their taxonomy of stereo matching algorithms development. The Middlebury page provides some common benchmark datasets and evaluation systems that all researchers can utilize to examine their proposed methods objectively and universally. Based on the rank given by the website, the common techniques can be found and adopted in many sophisticated algorithms. The post-processing step for the stereo matching algorithm is the disparity refinement has received a lot of attention in recent years. In this step, raw disparity maps computed by correspondence algorithms contain outliers that must be identified and corrected. Several approaches aimed at improving the raw disparity maps computed by stereo correspondence algorithms such as sub-pixel interpolation [23], image filtering techniques, Bidirectional Matching [24] and Single Matching Phase [25]. Even though the proposed algorithm provides exceptional accurate disparity depth map, it has high complexity for the implementation particularly for real-time applications.

One of the main objectives of the multi-view video system is to create another dimension to the viewer and provide 3D information such as depth. Normally, the depth information of the scene can be obtained through stereo matching algorithms. Another depth acquisition method is based on the active range sensor that uses special equipment for measuring range of a scene, such as well-known time-of-flight (TOF) depth cameras [26-28]. The TOF depth camera emits light signals itself and then measuring the arriving back time of the signals to obtain the range data. However, in spite of its high price, it merely yields small spatial resolution images with noise. Another popular effort to obtain the depth map is by using the Kinect sensor by Microsoft, which consists of a projector-camera pair as the depth sensor that measure per pixel disparity. The Kinect sensor has gained much popularity in the scientific and the entertainment community lately [29]. Although it could produce good results on the depth, it is not flexible for multi-view camera arrays configuration to create dense image based rendering.

Researchers in academia and industry realize the need to perform efficient 3D video processing that includes not only for the computer-based visual effects but also for the real video and images captured by the multiple cameras. It also includes the entire

processing chain including camera configurations, acquisition, 3D content, processing, editing, coding, transmission, rendering and display that requires consideration [2]. There are correlations between all of the process involved. There are a number of efforts being made to perform particular block of processing in the multi-view video coding and step towards to create 3D video [11, 13]. This thesis serves the objective of bringing the gap between development of an efficient algorithm and practical implementation, by highlighting the problem issues in the stereo matching and image synthesis algorithm as the main interest in this research.

Due to the increased demand of efficient multi-view video for 3D and free-viewpoint video, the novel view synthesis is really important to provide additional view between the multiple cameras. This will reduce the number of cameras used to obtain the multi-view images and videos. To achieve multi-view video system, it is essential that algorithms executed with multi-camera arrays, which starting from the stereo. Stereo imaging is more intuitive and general because it is similar to human eyes system. However, when the application requires multiple camera array configurations, the cameras will not be fitted in stereo. For example, in dense camera and free-viewpoint television system, the cameras can be arranged in many directions. There are a number of sophisticated algorithms developed in recent years, but some of them need high computational demand for computer graphics [2, 30, 31]. The work presented in this thesis is part of an effort to develop efficient novel inter-view synthesis for a multi-camera system configuration. The new structures and design are shown to offer improved performance with fewer camera density compared with the conventional high volume camera configuration for multi-view and free-viewpoint applications.

1.3 Summary of Original Contributions

The major contributions of this work are in the field of image and video processing, specifically the development of new inter-view synthesis algorithms for 3D vision and free-viewpoint video applications. One of the main objectives in this research is to create inter-view image that locates in the virtual viewpoint between source image viewpoints based on the disparity depth map obtained from the stereo matching algorithms. The proposed techniques create the new virtual image through layered disparity depth map representation. Secondly, the refined disparity depth map image is

reproduced from the raw disparity map by using the depth image layers separation and refinement adaptation techniques. The focus of this research is to yield disparity depth map with low complexity and computation time with simple stereo correspondence matching algorithms. Lastly, the research is to develop novel multi-view synthesis for a limited number of cameras to produce arrays of multi-image for free-viewpoint applications such as in light field imaging. In the pursuit to this aim, three novel techniques are noted.

1. ***Depth Image Layers Separation (DILS) algorithms for layered depth image-based synthesis and rendering.*** The first novel technique is referred to as the Depth Image Layers Separation (DILS) algorithm. This technique is used to synthesize novel inter-view images based on disparity depth map layers representation. The depth layers are identified through histogram distribution and separated into several clusters of layer. Each layer is extracted with inter-view interpolation to create objects based on location and depth. DILS features a new paradigm that is not just a method to select interesting locations in the image based on the depth; it also produces a new image representation that allows objects or parts of the image to be described without the need of segmentation and identification. The image view synthesis can reduce the complexity of multi-camera array configuration for 3D imagery and free-viewpoint applications. It makes use of a disparity depth map layer separation for image based synthesis and rendering through multi-layer and overlapping techniques. With the selected layer of depth, disparity depth map can be refined independently and the layer can be composed onto different 3D scene. By exploiting the disparity depth information, it is possible to discriminate some background or foreground objects. This is useful for intelligent video tracking and image based rendering. The simulation results show that the concept of depth layers separation is able to create inter-view images integrated with other technique such as occlusion handling processes. The DILS algorithms can be performed from a simple to sophisticated stereo matching techniques to synthesize the inter-view images.
2. ***Depth Layers Refinement (DLR) for the disparity map with DILS algorithm.*** The second contribution is the development of a new disparity refinement of the disparity depth map based on the DILS algorithm, known as Depth Layers

Refinement (DLR). The core of the algorithm relies on a layer separation process based on different disparity range and the mapping between the layer and the segmented reference image. Each disparity layer and segmented reference image morphologically processed based on cross-path points from the mapping procedure. With this approach, the uniform areas and repetitive patterns can be grouped in a single layer of depth. The depth discontinuities can be improved significantly compared with the conventional block-based matching technique. A comparative analysis of existing stereo matching algorithms with the proposed algorithm is conducted based on the common benchmark datasets and evaluation system in the Middlebury Stereo Vision Page. The algorithm is shown to efficiently refine the disparity depth maps and improves the pixels matching between the two images with a basic similarity metric. The main difference between this algorithm and the sophisticated algorithms is that this approach refines the disparity map and detects the depth discontinuities based on the layers separation.

3. ***Multi-Level View Synthesis (MLVS) with DILS algorithm for sparse multi-view camera arrays.*** The third novelty is the multi-view synthesis for 3D vision and free viewpoint video applications. This method exploits the advantage of the new inter-view interpolation algorithms based on DILS algorithm by extending the stereo to multiple camera configurations. In this technique, novel multi-view synthesis created based on a limited number of cameras for a sparse camera arrays, for example four cameras used to create nine view images. This will reduce the camera usage required to create dense image. This method is known as the Multi-Level View Synthesis (MLVS), which finds the pixel matching correspondences and synthesis through three stages (levels) of process. The first stage identifies the pixel correspondences and view synthesis based on the left-right image pair. Meanwhile, the view synthesis image in the second stage is based on the upper-lower image pairs through vertical matching. The third stage will use the obtained output in the first or second stage for the new inter-view synthesis to create full virtual multi-camera array image views. The new structures and design are shown to offer improved performance and provide additional views with fewer cameras arrangement compared to the conventional high volume camera configurations for free-viewpoint video acquisition.

1.4 Thesis Organization

This thesis is organised into seven chapters. Chapter 1 provides an introduction of the research work presented in this thesis. Along with this introduction, this chapter also describes in brief the main research contributions of the thesis. Besides the introduction and conclusion chapters (Chapter 1 and Chapter 7), the core part of this thesis is divided into five chapters. Each individual chapter focuses on a sub-system of the 3D video and the proposed multi-view depth synthesis system. Figure 1.1 shows the proposed system architecture, which is composed of acquisition of 3D video system, calibration and rectification (reviewed and discussed in Chapter 2), stereo matching algorithm with post-processing and view synthesis based on disparity depth map (Chapter 3), depth layers extraction and novel inter-view synthesis rendering based on layers representation (Chapter 4), disparity refinement based on depth layers (Chapter 5) and lastly on multi-view synthesis (Chapter 6) and compression sub-system. In the following, we outline each of these sub-systems (chapters).

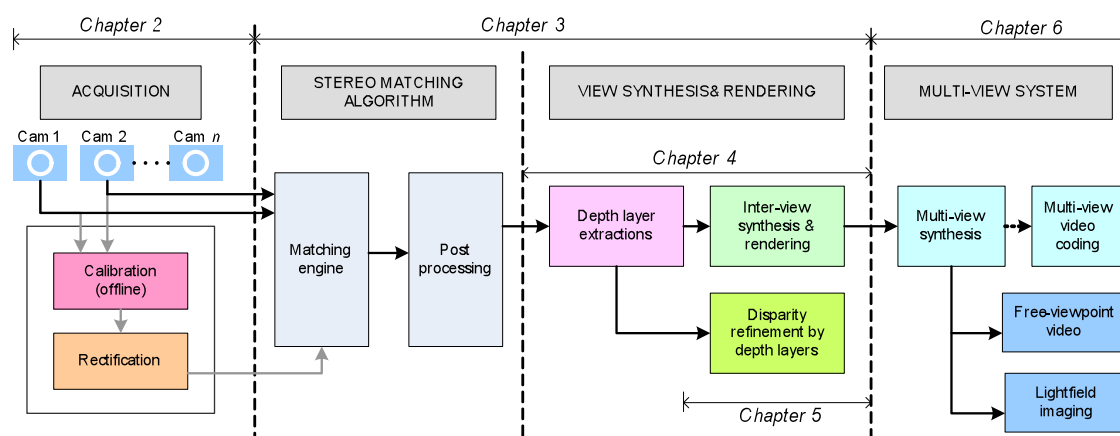


Figure 1.1: Overview of the proposed multi-view depth synthesis system that includes the acquisition of 3D video, calibration and rectification, stereo matching, view synthesis, layer based rendering, multi-view synthesis and compression sub-system

Chapter 2 contains introductory material and a review of the research work in the field of multi-view imaging starting from stereo to multi-view vision. This chapter presents the basic fundamental of stereo vision, which is the projective to the two-view geometry. It serves as the mathematical framework for the multi-view imaging and 3D vision. This chapter also surveys the multi-view imaging applications such as stereoscopic displays, free-viewpoint video applications and video editing and special effects. It will describe the overview of the well-known techniques and algorithms developed in the 3D video system and its main component, which includes the

acquisition, compression and rendering. Efficient compression algorithm is important in the multi-view video due to the huge amount of data for storage and transmission. Therefore, this chapter will also discuss the multi-view video coding algorithms. The redundancies between the camera views (inter-view) and temporal correlation exploited with the existing standards such as MPEG-2 and H.264/AVC.

Chapter 3 describes the fundamentals of the core components used in the development of multi-view and depth synthesis techniques. This chapter showcases the two main features that are stereo matching algorithms and view synthesis algorithms. It will describe the main building blocks of matching engine that is stereo matching classification and trends in finding the pixels correspondence between the stereo images. The evolution of stereo matching algorithms is described. From the stereo matching, the disparity depth map can be measured and used as the next stage of 3D image and video processing, which is the inter-view synthesis algorithm. The 3D scene representation discussed in this chapter will lead to the image synthesis and rendering algorithm. The related research works on layered depth image based rendering are also presented.

Chapter 4 presents the Depth Image Layers Separation (DILS) algorithm used to synthesize the virtual inter-view images based on layers representation. The main idea of this approach is to separate the depth map into several layers of depth based on the disparity distance of the corresponding points of the stereo pair images. The overall framework of the system design and algorithm presented in this chapter consist two main stages: matching algorithm and intermediate view synthesis. The novel view synthesis interpolated independently to each layer of depth from the left and right through masking the particular depth layer. The advantages of this approach are that any particular depth can be treated independently to make it more robust and accurate. The performance evaluation in terms of results and quality also presented in this chapter by using Middlebury datasets [32].

The Depth Layers Refinement (DLR) algorithm is presented in Chapter 5. The main objective of this algorithm is to improve the disparity maps in the disparity refinement stage for the stereo matching algorithms. This chapter provides an overview of the system design and also outlines the main features of the model that consist two main modules: stereo matching algorithm and disparity refinement module. It also covers the proposed algorithm for the disparity refinement by adapting the DILS algorithm that

presented in Chapter 4, to accomplish the objectives of this research work. The results and performance are discussed in this chapter by comparing the proposed algorithm with the state-of-the-art stereo matching algorithm in the Middlebury Ranking Stereo Page [22, 32].

Chapter 6 describes a new technique of obtaining dense camera array images by using a limited number of cameras. This method exploits the advantage of the inter-view interpolation technique presented in Chapter 4 to create novel view synthesis footage through multi-layer and depth synthesis algorithms by extending stereo to multiple camera configurations. The third novelty presented in this Chapter is known as the Multi-Level View Synthesis (MLVS) algorithm, which finds the pixel correspondences and syntheses through three levels of matching and synthesis processes. This chapter provides an overview of the proposed system design architecture and the MLVS algorithm. The experimental results and the parameters used in the algorithm will be reviewed by using the multi-camera datasets provided by Stanford University [33, 34]. Some essentials analysis of the selected multi-camera datasets and algorithm implementation issues are also discussed in this chapter.

Finally, Chapter 7 provides conclusions of the research in this thesis and outlines some future work directions.

Chapter 2

Fundamentals of Multi-View Imaging and 3D Video

2.1 Introduction

Recently, Multi-View Imaging (MVI) has attracted attention due to its increasingly wide range of applications and decreasing cost of digital cameras. This provides many opportunities to new and interesting research topics and applications, such as virtual view synthesis for 3DTV and free-viewpoint TV [35, 36], high performance imaging, video processing and analysis for surveillance, distance learning, industry inspection and so on. The availability of multiple views of a scene makes possible new and exciting applications ranging from 3D and free-viewpoint TV to robust scene interpretation and object tracking. The hardware for multi-camera systems is developing fast and is already being deployed for multimedia, security and industrial applications. However, there are still some challenging issues in terms of processing, primarily due to the sheer amount of data involved when the number of cameras becomes very large. Therefore, it is important to understand how the stereo and multi-view information is structured and how to take advantage of the inherent redundancy that results when the cameras capture the same scene.

This chapter will start by providing an insight on the nature of data in multi-view imaging systems and the fundamentals of the three-dimensional (3D) starting from stereo to multi-view. Then the basic fundamental of stereo vision presented in Section 2.3, which serves as the mathematical framework for the multi-view imaging and 3D graphics with the relationship of the 3D world and its corresponding position in 2D image. Section 2.4 surveys the multi-view imaging applications such as 3D television and free viewpoint video applications. 3D video systems and its main component are covered in Section 2.5, which includes the acquisition, compression and rendering. By using the basic structure of stereo, a multidimensional framework for the multi-view

video coding is derived. An efficient compression algorithm is important in the multi-view video due to the huge amount of data for storage and transmission. Section 2.6 will discuss the multi-view video coding algorithms. The redundancies between the camera views (inter-view) and temporal correlation exploited with the existing standards such as MPEG-2 and H.264/AVC. This includes overview of multi-view compression algorithms such as the conventional stereo video coding, video plus depth data and multi-view video coding. And finally the conclusion described in Section 2.7.

2.2 Stereo Vision to Multi-view

The word '*stereo*' derived from Greek and can be interpreted as 'solid' and 'hard'. This term gradually evolved in French as '*stere de bois*', which corresponds to a volumetric unit for a pile of wood. Meanwhile, 'stereo visualization' refers to the visual perception of the solid three-dimensional (3D) properties of some objects [37]. Since its initiation in 1838 by Sir Charles Wheatstone, stereoscopy has been widely used in photography and the film making industry. In stereo visualization, the binocular vision relates to the interpretation of two slightly different views of the same object seen by both human eyes. It is the ability of the human brain to process subtle differences between the images that are presented to the left and right eyes to perceive 3D outside world. Hence, stereo vision is the ability of the human brain to analyse the differences between the left and right eye views, determining whether objects are closer or further to the observer. The subtle difference between the left and right eye views is called disparity and is processed by the brain to yield three-dimensional perception of the scene being viewed. The concept has been illustrated with a device known as 'stereoscope', which paints two different images of the same object directly onto the reviewer retina. This early development in stereoscopic visualization leads to the invention of photography.

An artificially produced pair of so-called stereo images corresponding to the same scene seen by slightly different perspectives can be presented to an observer in a way so that the right image is seen by the right eye and the left image to be seen by the left eye. Then the human observer perceives the scene in depth by processing the relative difference between two images. This idea leads to the technique aimed at inferring depth using two or more cameras. There are wide research topics in computer vision that includes binocular stereo vision systems, dense stereo algorithms and stereo vision

application. Two separate cameras or binocular stereo camera can be used to obtain the stereo image and video. Although the basic principle of stereoscopic image acquisition seems simple, many pitfalls exist that can make editing of stereoscopic material a tedious task.

The principle that relates two 2D images to a 3D representation of an object can be extended to multi-view configurations. The 3D description of the object can be obtained accurately with more than two images, which is through the multiple views. Therefore, stereoscopic 3D properties of a scene can be derived from multiple views or multi-view images captured by a set of multiple cameras. For example, the background and foreground orientation and the relative positions of objects in the 3D scene can be extracted by analyzing the multi-view images. Research in multi-view camera systems from image processing point of view, means more dimensions are able to be visualized. The fundamental structure and coherent in multi-view images has been summarized by Berent in [3]. The number of dimensions goes up to seven when all the degrees of freedom are taken into account. The visual information captured depends on the viewing position (V_x, V_y, V_z) , the viewing direction (θ, ϕ) or (x, y) , the wavelength λ and the time t if dynamic scenes are considered. Adelson and Bergen [38] gather all these parameters into a single function called as *plenoptic function* and can be represented as:

$$P = P_7(x, y, \lambda, t, V_x, V_y, V_z) \quad (2.1)$$

where x and y are analogous to the Cartesian coordinates on the image plane.

Usually the wavelength is omitted by considering separate channels for colour images or one channel for greyscale images.

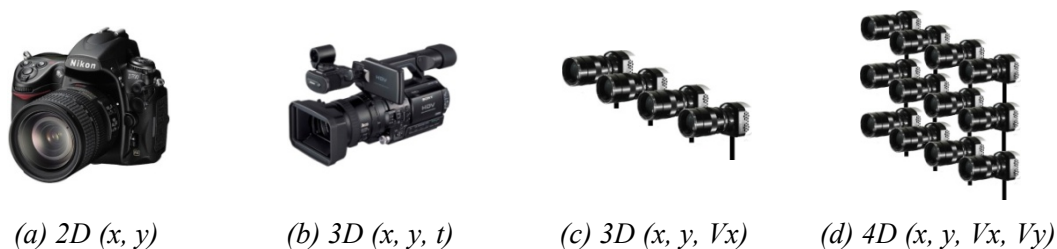


Figure 2.1: Capturing the plenoptic function from the still image camera to the video camera or multi-view imaging systems [3]

There are many different ways to capture the plenoptic function and most of the popular sensing devices do not necessarily sample all the dimensions. Figure 2.1 [3] illustrates a few techniques to capture the plenoptic function. For instance, the still image camera

fixes the viewing point and time. Only (x, y) dimensions remain. The video camera is able to capture images at different times and therefore captures the (x, y, t) dimensions. Another case of a three-dimensional plenoptic function can be obtained by giving one degree of freedom to the camera location such that (x, y, V_x) is sampled. Higher dimensional cases add more degrees of freedom to the viewing position such that (x, y, V_x, V_y) or even (x, y, V_x, V_y, V_z) can be captured.

Berent [3] introduced the concept of *plenoptic manifolds*, where the extraction of the plenoptic function in all dimensions can be very useful in numerous multi-view imaging applications such as layer-based representations [39], object based coding, disparity compensated [40], shape adaptive wavelet coding [41] and image based rendering (IBR) [42]. It also includes the case of occlusions, large depth variations, scene interpretation and understanding. With this fundamental concept, several applications for multi-view images can be outline such as the 3DTV and free-viewpoint video.

3D and free-viewpoint video are new types of natural video media that expand the user's experience far beyond what is offered by traditional media. The first offers a 3D depth impression of the observed scenery, while the second allows for interactive selection of viewpoint and direction within a certain operating range as known from computer graphics applications. Applications of stereo vision include robotics, automatic navigation systems, entertainment and machine aided surgery. From this range of applications, it is apparent that the output of a stereoscopic system might be viewed by a human being or taken as an input to a computer algorithm for further processing. Therefore, not all stereo pairs used to be viewed by human beings. As a result, several aspects of acquiring, processing and displaying stereo pairs to be viewed by human beings are determined by the limitation of the human visual system.

These applications are enabled through convergence of technologies from computer graphics, computer vision, multimedia and related fields. It also visualised by the rapid progress in research covered from the whole processing: acquisition, signal processing, data representation, compression, transmission, display and interaction. Some of these applications maybe based on particular systems, for example, in the post production of films and TV content [43]. The applications on multi-view imaging presented in the Section 2.4. The next section will describe the mathematical framework for the multi-view imaging system.

2.3 Camera Projective to Two-View Geometry

Projective geometry serves as a mathematical framework for 3D multi-view imaging and 3D graphics. It is used to model the image formation process, generate synthetic images or reconstruct 3D objects from multiple images. Besides the projective geometry, the Euclidean geometry also can be used to model lines, planes of points in a 3D space. However, the Euclidean geometry cannot model points at infinity. It is considered as a special case and this can be illustrated using a perspective drawing of two parallel lines. In perspective, two parallel lines such as a highway road, meet at infinity at the vanishing point. The intersection of the parallel lines at infinity is hard to model by the Euclidean geometry. Due to that, the projective geometry offers a solution.

The relation that maps the points Q_i in the physical world with coordinates (X_i, Y_i, Z_i) to the points on the projection screen with coordinates (x_i, y_i) is called the projective transform. When working with such transforms, it is convenient to use homogeneous coordinates. The homogeneous coordinates associated with a point in a projective space of dimension n are typically expressed as an $(n+1)$ -dimensional vector (for example x, y, z becomes x, y, z, w), with additional restriction that any two points whose values are proportional are equivalent.

In Euclidean space, a point defined in 3D is represented by a 3-element vector $(X, Y, Z)^T$. In the projective space, the same point is described using a 4-element vector $(X_1, X_2, X_3, X_4)^T$ such that [37],

$$X = \frac{X_1}{X_4}, \quad Y = \frac{X_2}{X_4}, \quad Z = \frac{X_3}{X_4} \quad (2.2)$$

where $X_4 \neq 0$.

Usually, the coordinates $(X, Y, Z)^T$ and $(X_1, X_2, X_3, X_4)^T$ are called inhomogeneous coordinates and homogeneous coordinates, respectively. As a generalization, the mapping from a point in the n -dimensional Euclidean space to an $(n+1)$ -dimensional projective space can be written as [37]:

$$\underbrace{(X_1, X_2, \dots, X_n)^T}_{\text{Euclidean space}} \rightarrow \underbrace{(\lambda X_1, \lambda X_2, \dots, \lambda X_n, \lambda)^T}_{\text{projective space}} \quad (2.3)$$

where $\lambda \neq 0$ corresponds to a free scaling parameter.

This free scaling parameter λ is used for the pinhole camera model.

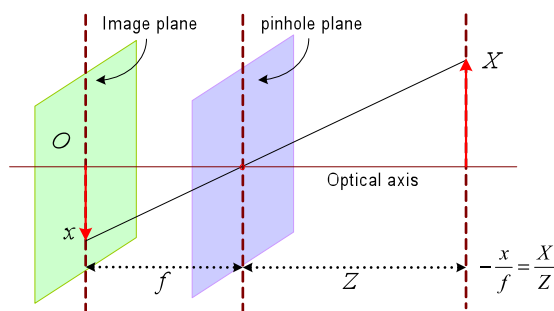


Figure 2.2: Pinhole camera model [44]

The basics of an image acquisition process start from the pinhole camera model. The model integrates with the internal (intrinsic) camera parameters, such as the focal length, CCD dimensions and the lens distortion. It also linked to the external (extrinsic) camera parameters corresponding to the position and orientation of the camera. The pinhole camera model is the simplest model of a camera as illustrated in Figure 2.2 [44]. In this model, light is envisioned as entering from the object but only a single ray enters from any particular point. In a physical pinhole camera, this point is then projected onto an imaging surface. As a result, the image on this image plane is always in focus, and the size of the image relative to the distance object is given by a single parameter of the camera, that is its focal length. For ideal pinhole camera, the distance from the pinhole aperture to the screen is precisely the focal length. In the actual image plane, the scene appears inverted. As shown in Figure 2.2, where f is the focal length of the camera, Z is the distance from the camera to the object, X is the length of the object, and x is the object's image on the imaging plane. In the figure, the similar triangles can be seen as $-x/f = X/Z$.

The pinhole camera model from Figure 2.2 is rearranged to an equivalent representation shown in Figure 2.3 [44, 45]. The main difference is that the object now appears right-side up. The new image plane is also known as the virtual image. The point in the pinhole is reinterpreted as the centre of projection, which is known as the optical centre or camera centre. The point at the intersection of the image plane and the optical axis is referred to as the principal point. On the new frontal image plane, the image of the distant object is exactly the same size as it was on the image plane in Figure 2.2.

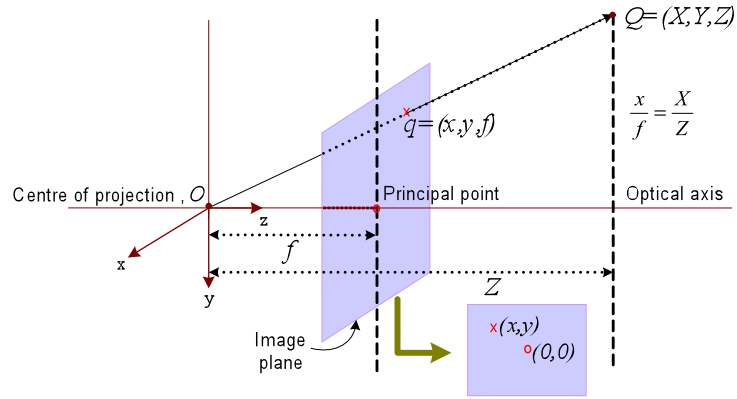


Figure 2.3: Rearranged pinhole camera model [44, 45]

The pinhole camera model as shown in Figure 2.3 defines the geometric relationship between a 3D point $Q=(X,Y,Z)$ and its 2D corresponding projection $q=(x, y)$ onto the virtual image plane. The geometric mapping from 3D to 2D is called a perspective projection. The image is generated by intersecting these rays with the image plane, which happens to be exactly a distance f from the centre of projection. This makes the similar triangles relationship $x/f=X/Z$ more directly evident than before and the point q can be simplified through rescaling as [44, 46],

$$q = \left(f \frac{X}{Z}, f \frac{Y}{Z}, f \right) \quad (2.4)$$

Consider a camera with the optical axis being collinear to the Z -axis and the optical centre being located at the origin of a 3D coordinate system as shown in the pixel position $q=(x, y)$. Therefore, the 2D coordinate of point q in the virtual image plane is given by [44],

$$(x, y) = \left(f \frac{X}{Z}, f \frac{Y}{Z} \right) \quad (2.5)$$

where the image centre is $(0, 0)$ and f denotes the focal length.

This formula can be transformed to the projective geometry framework as [37],

$$(\lambda x, \lambda y, \lambda)^T = (Xf, Yf, Z)^T \quad (2.6)$$

This relation can be expressed in matrix form as [37],

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.7)$$

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

Homogeneous coordinates are the key to all computer graphics and vision system. All standard transformations (rotation, translation, scaling) can be implemented by matrix multiplications with 4x4 matrices [37]. The projection can be implemented by multiplication with 3x4 matrices.

2.3.1 Homography

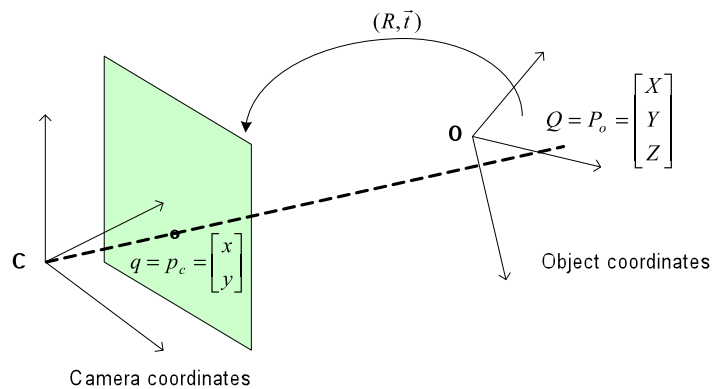


Figure 2.4: Converting from object to camera coordinate systems [37, 44]

In computer vision, planar homography can be defined as a projective mapping from one plane to another. Thus, the mapping of points on a two-dimensional planar surface to the imager of the camera is an example of planar homography. It is possible to express this mapping in terms of matrix multiplication if the homogeneous coordinates are used to express both the viewed point Q and the point q on the image plane to which Q is mapped [37, 44] (as illustrated in Figure 2.4). If we define,

$$\begin{aligned} \tilde{Q} &= [X \ Y \ Z \ 1]^T \\ \tilde{q} &= [x \ y \ 1]^T \end{aligned} \quad (2.8)$$

Then we can express the action of the homography as [37]:

$$\tilde{q} = sH\tilde{Q} \quad (2.9)$$

The parameter s is an arbitrary scaling factor. The transformation matrix can be solved

with some geometric and matrix algebra. The most important part is that H has two parts: the *physical transformation*, which essentially locates the object plane, and the *projection*, which introduces the camera intrinsics matrix. The physical transformation part is the sum of the effects of some rotation R and translation T that relate the plane viewed to the image plane. Due to homogeneous coordinates, the information of physical transformation can be combined within a single matrix as follows [37, 44]:

$$W = [R \quad T] \quad (2.10)$$

The action of the camera matrix M in projective coordinate is multiplied by $W\tilde{Q}$ that yields [37]:

$$\tilde{q} = sMW\tilde{Q}, \quad \text{where} \quad M = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

where c_x and c_y are possible displacement of the centre of coordinates parameter. Without loss of generality, the object plane can be defined so that $Z=0$. Therefore, the rotation matrix can be described into three 3-by-1 columns [37].

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.12)$$

The homography matrix H that maps a planar object's points onto the image plane is then described completely by [37],

$$H = sM \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad \text{where,} \quad \tilde{q} = sH\tilde{Q} \quad (2.13)$$

2.3.2 Two-View Geometry

Although the search for corresponding points in stereo imaging can be computationally expensive, the knowledge on the geometry of the system can be used to narrow down the search space as much as possible. In practice, stereo imaging involves four steps when using two cameras, and it can be described as follows:

- i. Mathematically remove radial and tangential lens distortion (described in Appendix A). The outputs of this step are undistorted images.

- ii. Adjust for the angles and distances between cameras, a process called *rectification*. The outputs of this step are images that are row-aligned and rectified.
- iii. Find the same features in the left and right camera views, a process known as *correspondence*. The output of this step is a disparity map, where the disparities are the differences in x -coordinates on the image planes of the same feature viewed in the left and right cameras.
- iv. With the known geometric arrangement of the cameras, the disparity map can be turned into distances by *triangulation*. This step is called *reprojection*.

The geometry of the two cameras relates to the respective position and orientation and internal geometry of each individual camera. The underlying geometry that describes the relationship between both cameras is known as the epipolar geometry. The epipolar geometry addresses the following two aspects [37]:

- Geometry of point-correspondence: considering a point in an image, the epipolar geometry provides a constraint on the position of the corresponding point.
- Scene geometry: given point-correspondences and the epipolar geometry of both cameras, a description of the scene structure can be recovered.

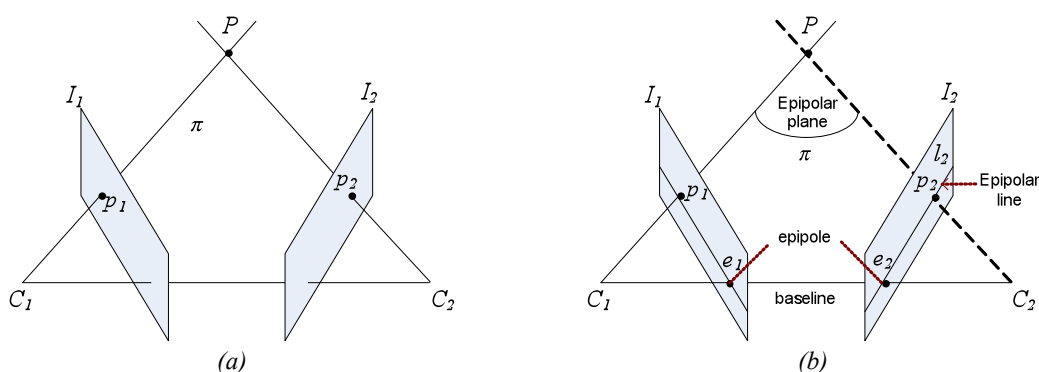


Figure 2.5: Epipolar geometry [37]

Figure 2.5 [37] shows epipolar geometry, which is defined by the point P and the two camera centres, C_1 and C_2 . The 3D point P is projected through the camera centres C_1 and C_2 onto two images at pixel positions p_1 and p_2 , respectively. Clearly, the 3D points P , C_1 , C_2 and the projected points p_1 and p_2 are all located within one common plane. This common plane denoted by π , is known as the epipolar plane.

The epipolar plane is fully determined by the back-projected ray going through C_1 and p_1 also with the camera centre C_2 . The property that the previously specified points belong to the epipolar plane provides a constraint for searching point correspondences.

Considering the image point p_1 , a point p_2 lies on the intersection of the plane π with the second image plane within I_2 in Figure 2.5(a). The intersection of both planes corresponds to a line known as the epipolar line. Therefore, the search of point-correspondences can be limited to a search along the epipolar line, instead of an exhaustive search in the image. Additionally, it is interesting to note that this constraint is independent of the scene structure and uniquely relies on the epipolar geometry.

The terminology related to the epipolar geometry based on Figure 2.5(b) can be summarized [37] as,

- i. The epipolar plane is the plane defined by a 3D point and the two cameras centres.
- ii. The epipolar line is the line determined by the intersection of the image plane with the epipolar plane.
- iii. The baseline is the line going through the two cameras centres.
- iv. The epipole is the image point determined by the intersection of the image plane with the baseline. Also, the epipole corresponds to the projection of the first camera centre (C_1) onto the second image plane (like I_2), or vice versa.

The 3D structure can be extracted by determining the correspondences in the two-views and that point-correspondences can be searched along the epipolar line only. In this case, the search of point-correspondences can be performed along the horizontal raster lines of both images.

However, it is difficult to accurately align and orient the two cameras such that epipolar lines are parallel and horizontal. Instead, an alternative approach is to capture two views (without alignment and orientation constraints) and transform both images to synthesize two novel views with parallel and horizontal epipolar lines. This procedure is called image rectification.

Image rectification is the process of transforming two images I_1 and I_2 so that their epipolar lines are horizontal and parallel. This procedure is useful for depth estimation

algorithms because the search correspondences can be performed along horizontal raster image lines. Practically, the image rectification operation corresponds to a virtual rotation of two cameras so that it would become aligned.

2.3.3 Triangulation

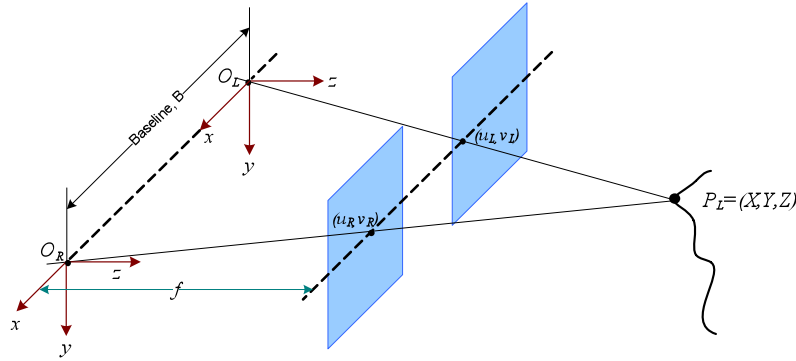


Figure 2.6: Two-camera views based on pinhole camera model [46]

In order to describe the relationship between two-view geometry is by referring to Figure 2.6 [46] that is derived from the pinhole camera model in the Figure 2.3, where O_L is the reference camera centre point (or the left camera), and O_R is the target camera centre point. The implementation of this system is based on parallel cameras, which are shifted along the same horizontal line or x -coordinate, known as the epipolar line. And therefore, $v_L = v_R$. The symbol f is the focal length of each camera's lens (the distance from camera centre point to the image plane) and B is the baseline distance (distance between the two optical centres, O_L and O_R). The points of the images can be described as the follows [46]:

$$(u_L, v_L) = \left(f \frac{X}{Z}, f \frac{Y}{Z} \right) \quad (2.14)$$

$$(u_R, v_R) = \left(f \frac{X - B}{Z}, f \frac{Y}{Z} \right) \quad (2.15)$$

The disparity of the stereo images is obtained as the difference between the two corresponding points, U_L and U_R [46]:

$$\text{disparity}, d = u_L - u_R = \left(f \frac{X}{Z} - f \frac{X - B}{Z} \right) \quad (2.16)$$

The location of correct projections of the same point P_L on the two image planes can determine the exact depth of P_L in the real world. From equation (2.16), the depth Z is

defined as [44-46]:

$$\text{depth, } Z = \frac{fB}{d} \quad (2.17)$$

The equations used to calculate the exact location of $P_L(X,Y,Z)$ for the 3D points are [44-46]:

$$X = \frac{Bx_L}{d}, \quad Y = \frac{By_L}{d}, \quad Z = \frac{Bf}{d} \quad (2.18)$$

The next section will discuss the applications of multi-view imaging based on technology and applications.

2.4 Applications of Multi-view Imaging

In this section, stereoscopic displays will be described as a technology to enable several specific applications. Then, some applications to a free-viewpoint video system are provided. Lastly, the usefulness of multi-view images for video editing will be illustrated.

2.4.1 Stereoscopic Displays

3DTV is becoming increasingly popular due to the rise of popular 3D feature films. Major television manufacturers began developing 3D home television technology in 2009. There are several methods that these manufacturers use to create 3D images on an LCD television; some are more expensive than others, and some are more feasible than others. In general, there are three primary methods of 3D home theatre technology: lenticular viewing, passive glass systems and active glass systems.

Stereoscopic displays allow the viewer to perceive the depth of the scene. It can be achieved by displaying a left and right image or view, as if it was individually seen by the left and right eyes. To obtain this result, several display technologies has been developed, which include the polarized displays, parallax barrier displays and lenticular displays. Stereoscopic lenticular displays or known as multi-view lenticular displays are based on a lenticular sheet, which is precisely positioned onto an LCD as shown in Figure 2.7(a). A lenticular sheet consists of an array of micro-lenses that directs the light of the underlying pixels in specific directions [20]. Consequently, the viewing

space in front of the display is divided into separate viewing zones, each of them showing a different view of the scene. Therefore, the perception of depth will be received by the viewer. This technology is different from the 3DTV displays that required viewer to wear special glasses to watch the 3D scene. The lenticular viewing technology has been pioneered by Philips [20].

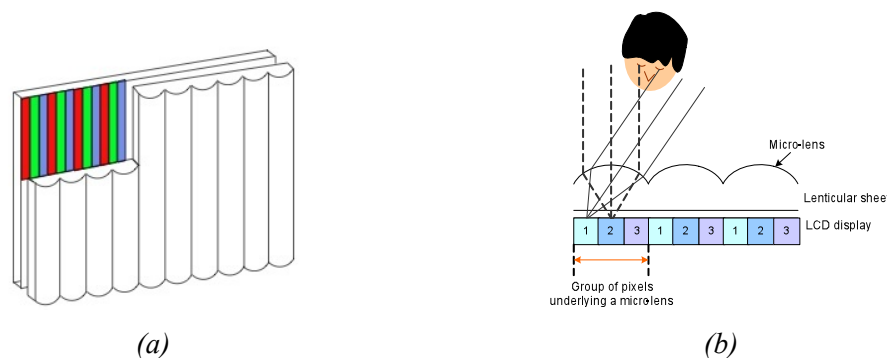


Figure 2.7: Stereoscopic lenticular display [15]. (a) A lenticular display sheet precisely positioned onto an LCD. (b) Multi-view lenticular display with three pixels/views covered by micro-lens

Figure 2.7(b) shows a three-view lenticular display that projects the light of three different pixels into three different viewing zones (two zones are drawn in the figure) [15]. Each view is projected into specific directions by the micro-lens, so that the left and the right eye see two different views. This method is quite different from the 3DTV aided by special transmitter glasses to view the 3D depth. Nine-view lenticular displays have been introduced to enable the viewer to watch the video scene from various viewpoints [9]. There exists a trade-off between the number of views supported by the display and the resolution of the image, where increasing the number of views involves a loss of image resolution. In the short term, two-view displays for stereo vision have gained strong popularity for 3D games and an early introduction to the 3DTV market. However, with this technology, the viewer must sit in a very specific spot in front of the TV. This means that only a couple of people would be able to watch the TV comfortably at once due to its small viewing angle.

Some manufacturers developed a passive glass system of LCD monitor that will allow both 2D and 3D images to be viewed. In order to watch the 3D images, viewers will need to wear the traditional glasses in order to watch 3D media. This technology is similar to watching 3D films in the cinema. The TV has two overlapping images and the glasses have polarized lenses. Each lens is polarized so that it can see only one of the two overlapping images.

The latest 3DTV technology uses active glass systems. Currently this is the dominant approach used by the manufacturers. The system is very similar to the passive glass system except that, rather than the TV doing all the work, the glasses do it instead (known as the active shutter glasses). The glasses synchronize with the refresh rate of the TV, and then alternate the polarization of each lens, making the wearers of the glasses see 3D images. With this technology, people could be watching a 2D or 3D movie comfortably. However, it requires the costly active shutter glasses to watch the 3D scene.

In term of applications, the stereoscopic display is well known for 3DTV as home entertainment. The 3DTV is expected to be the next revolution in television and display technology. It will become a key application of stereoscopic displays by providing the viewer with a new viewing experience to watch the 3D scene without wearing any special glasses. Besides that, the stereoscopic display will greatly enhance the gaming realism by showing a 3D representation of the virtual scene and characters in the video games. One main feature of video games is that a 3D description of the scene is provided by the game.

Another application for stereoscopic displays is the training and serious gaming systems [37]. The usage of stereoscopic displays simplifies the teaching of several important subjects, such as surgery to junior medical doctors. Junior surgeons are able to perceive the depth as seen by the senior operating surgeon. The stereoscopic displays can be employed as a generic medium or simulation tools for training junior professionals such as medical doctors, pilots, engineers or in military.

2.4.2 Free-Viewpoint Video

Free-Viewpoint Video (FVV) applications provide the ability for users to select and control a viewpoint of the video scene interactively. FVV offers the same functionality that is known from 3D computer graphics [47]. The user can choose their own viewpoint and viewing direction within a visual scene, meaning interactive free navigation [48]. In contrast to pure computer graphics applications, FVV targets real world scenes as captured by real cameras. It can be performed by capturing the video scene from multiple viewpoints. The target applications of FVV include broadcast television and other forms of video entertainment, as well as surveillance. These applications are enabled through convergence of technologies from computer graphics,

computer vision, multimedia and related fields with rapid progress in research covering the whole processing chains from capturing, signal processing, data representation, compression, transmission, display and interaction.

A FVV system not only shows an event from the existing camera viewpoint, but it also allows free navigation within the 3D scene. The ability to generate an arbitrary viewpoint for a particular scene provides an attractive scheme to the free-viewpoint video. Some interesting applications include the selection of arbitrary viewpoint for visualizing and analyzing sports or dynamic scenes or actions [49]. For example, in a football match, it is often required for the referee to know the position of the players to ensure fair play. By rendering an appropriate viewpoint of the playing field, the player positions can be derived and illustrated using the virtual view [37]. It is also fascinating for user applications like in the opera or concert where the user can freely choose the viewpoint, as well as for post-production. Systems for the post-production are already being used for sports and movies. Free-viewpoint video technologies also simplify video training activities. For example, the training of dynamic activities such as martial arts and dancing can be simplified by allowing the trainee to select a viewpoint of the scene.

Tanimoto [35, 36] developed a new type of display named as FTV (Free-Viewpoint TV). It is known as an innovative visual media that enables the viewer to view a 3D scene by freely changing their viewpoints to any virtual view perspective. It is easy to realize the free viewpoint for virtual scenes made by computer graphics. However, it is quite challenging for the real scenes. The concept and idea has been proposed and implemented on a single PC and a mobile player [35]. FTV is closely related to 3DTV. While 2DTV generates a single view and 3DTV generates 2 or more views for display, FTV generates infinite number of views since the viewpoint can be placed anywhere. FTV captures and transmits the information of all rays in a 3D scene. It is also regarded as a natural interface between human and environment as well as an innovative tool to create new types of content and art.

MPEG regarded FTV as the most challenging 3D media and started the international standardization activities for FTV. The first phase of FTV was MVC (Multi-View Video Coding), which enables the efficient coding of multiple camera views. MVC was completed in May 2009. The second phase of FTV is 3DV (3D Video), which is a standard that targets serving a variety of 3D displays.

2.4.3 Video Editing and Special Effects

In traditional 3D computer graphics (CG), artists create 3D models of a scene, animate it and render the 3D scene into a 2D image. It is a 3D-to-2D process where the artists have full control of how the scene should look like and how the characters should move [50]. Unfortunately, it is also a labour intensive and costly procedure. In computer vision, scientists deduce 3D information of a scene using 2D image or images. The processes of real-life image acquisition and 3D reconstruction are cheaper than acquiring the real-life 3D model, but the reconstructed model contains noise and is complicated.

For the past decade, a new field between computer vision and computer graphics has emerged, known as Image-Based Modelling and Rendering (IBMR) [51, 52]. It is a field that utilizes computer vision techniques to render graphics directly from images. Instead of going through the manual intensive 3D modelling, animating and computationally expensive 3D to 2D rendering processes, IBMR starts with 2D images, calculates the underlying 3D structure of the scene and renders new views of the scene as 2D images. It is a 2D-to-2D process with some knowledge of the 3D structure. The result is a faster rendering processes but with larger data sets. Depending on the application, IBMR is a powerful alternative to a traditional computer graphics (CG) approach.

Video-Based Rendering (VBR) is a sub category of IBMR that takes IBMR further into the temporal domain for the video editing and production [53-55], which supports the traditional CG animation. It has the advantages that IBMR offer; that is faster rendering for complex scene, reduced labour in animation and modelling. However, the disadvantages of this approach which include a lower degree of freedom in animation and a larger dataset are also inherited. Whether they come from the computer graphics or computer vision in the video editing and production, the multi-view images provide a good platform to utilize both fields for 3D video and free-viewpoint applications.

The viewpoint of the multi-view images can be manipulated in time and place, which can be done by the professional video editors. The multi-view image technology simplifies the production of special effects such as the ‘freeze’ effect that can be seen in the film of The Matrix. This effect provides the illusion to the viewer of freezing the time and gradually modifying the viewpoint of the scene. By exploiting the 3D

information, it is possible to discriminate some background or foreground objects. The video objects can be easily removed and reinserted into the video elsewhere by using the 3D information. It is also possible to insert synthetic 3D objects in the scene to obtain an augmented reality video scene.

2.5 3D Video System

All the applications presented in the previous section rely on multiple views of the scene. Therefore 3D video technologies enabling these various applications do not exclude each other and can be integrated into a single 3D video system. Several 3D video systems have been introduced to enable the 3DTV and free-viewpoint video applications. They can be classified into three types with respect to the amount of employed 3D geometry: N-texture representation format, partial 3D representation of the scene (depth map) and hybrid between N-texture with depth map.

The first type of 3D video system is based on multiple texture views of the video scene or known as N-texture representation format. The N-texture representation format become the basic for the Multi-view Video Coding (MVC) standard developed by the Joint Video Team (JVT) [12]. The description of the standard can be found in the Joint Draft 8.0 on Multi-view Video Coding [13]. The main challenge of the MVC standard is to define the efficient codec tools for the multi-view video due to the huge amount of data to be stored. A first coding tool exploits the similarities between the views by multiplexing the captured views and encoding with the H.264/AVC standard. A second coding tool equalizes the inter-view illumination to compensate for mismatches across the views captured by different cameras.

The second type of 3D video system relies on a partial 3D geometric description of the scene. The scene geometry described by a depth map or depth image, which specifies the distance between a point in the 3D world and the camera. Normally, a depth image is estimated from two images by identifying matched pixels in the multiple views, or known point-correspondence that represents the same 3D scene point. With the depth map, the new views can be rendered by using Depth Image Based Rendering (DIBR) algorithms. The DIBR corresponds to a class of rendering algorithms that use depth and texture images simultaneously to synthesize virtual images. Consider a 3DTV application, where it is assumed that the scene is observed from a narrow field of view

(short baseline distance between cameras). As a result, the combination of only one texture and one depth video sequence is sufficient to provide appropriate rendering quality (1-depth/1-texture). The 1-depth/1-texture approach is standardized through Part 3 of the MPEG-C Video.

The last type of 3D video system addresses the occlusion problem by combining the depth image and texture image, N-depth/N-texture. The problem of occluded regions can be addressed by combining multiple reference images that cover all regions seen by the virtual camera. The N-depth/N-texture representation format is compatible with different types of multi-view displays supporting a variable number of views.

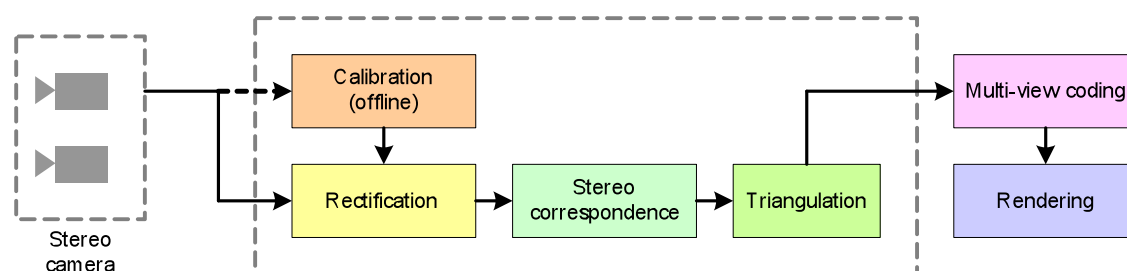


Figure 2.8: Overview of stereo video system [56]

In multi-view system, multiple cameras capture the same scene. For stereo system, two cameras will be used for the acquisition. Figure 2.8 [56] shows the overview of a stereo vision system. This section will describe the overview of stereo video system that consists of calibration, rectification, stereo matching, triangulation, coding and rendering.

2.5.1 Camera Acquisition, Calibration and Rectification

One of the problems dealing with multiple camera views is the signal ambiguity while identifying the correspondence points. With multiple cameras, the internal settings like the contrast setting can vary, which may result in dissimilar intensity values at the correspondence points. This will contribute to an unreliable identification of the point-correspondences and thus inaccurate depth values. Light reflection on the surface in different directions with varying intensity might occur, known as specular reflection phenomenon. As a result, object surfaces appear differently depending on the viewpoint. Such surface is identified as a non-Lambertian surface. Therefore, camera calibration is quite important, which includes special methods for calculating the internal and external parameters. Internal parameters describe the perspective projection, the lens and sensor

(chip) distortion and the digitization process. External parameters describe the rigid-body transformations between the main camera frame and the world frame, such as rotations and translations. A detailed description of calibration can be found in [45, 57] and Appendix A.

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. The purpose of the calibration is to establish the relationship between 3D-world coordinates and their corresponding 2D image coordinates. Once this relationship is established, 3D information can be inferred from 2D information and vice versa. In an application involving multiple cameras, this step is necessary to guarantee geometric consistency across different terminals. Calibration is carried out by acquiring and processing more than 10 stereo pairs of a known pattern, typically a checkerboard. The calibration procedure is available in OpenCV [44] and Matlab [58].

Rectification is a process used to facilitate the analysis of a stereo pair of images by making it simple to enforce the two-view geometric constraint as discussed in Section 2.3.2. This procedure is particularly useful for depth-estimation algorithms because the search for point correspondences can be performed along horizontal raster image lines. Many stereo algorithms assume this simplified form because subsequent processing becomes much easier if differences between matched points in one direction only. Practically, the image-rectification operation corresponds to a virtual rotation of two cameras, so that they would become aligned. By using the output of calibration, lens distortions can be removed and turns the stereo pair in standard form as shown in Figure 2.9. The rectified images can be often regarded as acquired by cameras rotated with respect to the original ones or images of these cameras projected onto the same plane.

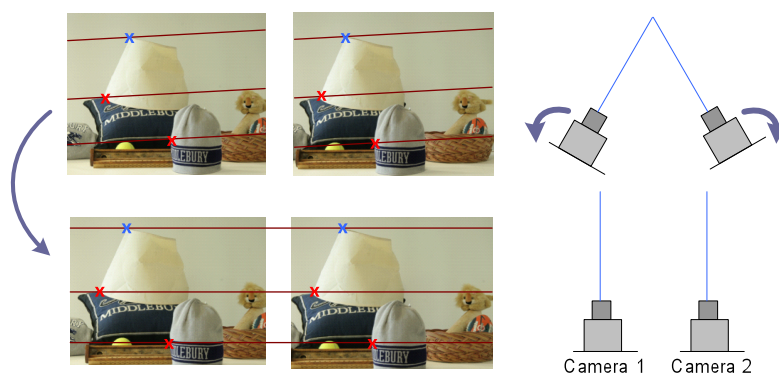


Figure 2.9: Rectification of stereo camera to standard form [37]

2.5.2 Stereo Correspondence and Depth Map

Stereo correspondence or stereo view matching is the fundamental problem of determining which parts of two images (views) are projections of the same scene element. The main aim of stereo matching algorithms is to find homologous points in the stereo pair. The output is a disparity map for each pair of cameras, giving the relative displacement (disparity) of corresponding image elements. Disparity maps allow estimating the 3D structure of the scene and the geometry of the cameras in space. The search for matches between two images is simplified and speed up if the two images are warped in such a way that correspondence points lie on the same scan line in both images based on the rectification process.

Scharstein and Szelinski [22] provided a valuable taxonomy and evaluation of dense stereo matching algorithms for rectified image pairs, arguing that most algorithms perform four steps: matching cost computation, cost aggregation, disparity computation/optimization and disparity refinement. The topic will be extensively discussed in Chapter 3. With the given disparity map, baseline and focal length from calibration, the position of the correspondence in the 3D space can be computed. This process is known as triangulation.

Generally the depth information of the scene can be obtained through stereo matching algorithms. However, the depth can also be obtained with a different depth acquisition method that is based on the active range sensor. This uses special equipment for measuring range of a scene, such as well-known Time-Of-Flight (TOF) depth cameras [26-28, 59]. The TOF depth camera emits light signals itself and then measuring the arriving back time of the signals to obtain the range data. In spite of its high price, it merely yields small spatial resolution images with noise. Zhu [26] describes a method for fusing depth from stereo cameras and TOF cameras. The performance review of 3D TOF system in comparison to stereo vision with the system has been discussed by Hussman [60]. The major advantage of the TOF technology is the delivery of an evenly distributed range and intensity images because each pixel calculates a range and intensity value. Hence the correspondence problem of conventional stereo vision system does not exist. However, the range resolution depends on the chosen modulation frequency and the power rating of the used illumination source.

Another popular effort to obtain the depth map is by using the Kinect sensor by Microsoft, which consists of a projector-camera pair as the depth sensor that measures per pixel disparity. Lately, the Kinect sensor has gained much popularity in the scientific and the entertainment community [61]. There are so many existing and on-going project using the Kinect as the framework to obtain the depth maps and used in many applications [29]. Although it could produce good results on the depth, the system is not flexible for multi-view systems due to the sensor interference problem when interacting with multiple Kinect sensors.

2.5.3 Multi-view Compression

Multi-view video requires a large amount of data for storage and transmission. Therefore, an efficient compression algorithm is vital. Each of the camera views can be coded independently with the state-of-the-art H.264/AVC standard [11]. However, it does not exploit the redundancies between the camera views. The correlation between the camera views is usually referred to as inter-view correlation. The correlation exists both for texture and depth signals. For each view, the succeeding frames have a correlation over time, i.e. a temporal correlation. This correlation is similar with the normal single view video signals. The temporal correlation within a single view is already exploited by the existing standards such as MPEG-2 and H.264/AVC by employing motion compensated transform coding.

The inter-view correlation can be exploited for compression, such as with predictive coding technique since the neighbouring views show most of the similar scene from a different viewpoint. The description of multi-view video coding will be discussed in Section 2.6.

2.5.4 Rendering for 3D

In general, rendering involves the read-out and presentation process of images. In a multi-view coding system, image rendering refers to the process of generating synthetic images. Synthetic images can be rendered by combining the multiple texture images with their corresponding depth maps. Significant progress in the field of image rendering for multimedia applications has been achieved over the past few years [52].

There are two aspects for rendering: visualization and compression. Both aspects can be simultaneously addressed when aiming at high quality rendering using algorithms such as 3D image warping [62, 63] and mesh-based rendering [64] techniques. Each of these two techniques suffers from either low image rendering quality or high computational complexity. An alternative formula proposed by Morvan [65] shows an improvement in rendering quality with a relief texture algorithm and an inverse mapping rendering technique. Chapter 3 will discuss the rendering algorithms in detail, which include the model-based rendering, image-based rendering and layered image based rendering.

2.6 Multi-view Video Coding Algorithms

Many 3DTV systems are based on scenarios, where a 3D scene is captured by a number of cameras. The simplest case is stereo video with two videos. For more advanced systems, it could apply 8, 16 or more cameras. Some systems traditionally apply per sample depth data that can also be treated as video signals. This section gives an overview of compression algorithms and standards for such data, which includes the conventional stereo video coding, video plus depth data and multi-view video coding. Overview of this field can be found in Shum [30]. Depending on the degree of common content, shared by a subset of the cameras, a coding gain can be achieved in comparison to single-view coding. In multi-view coding, correlations between adjacent cameras are exploited in addition to temporal correlations within each sequence in the inter-view direction.

2.6.1 Conventional Stereo Video Coding

A conventional stereo pair consists of two images showing the same scene from two slightly different viewpoints corresponding to distance of human eyes [66], which is the basic case for multi-view system. The images are in general very similar, which makes them well suited for compression, where one image predicts the other. For instance, one of them can be predicted from the already encoded one, just like temporally related images that can be motion compensated in video compression.

The displacement or disparity of each sample in one image with respect to the other is equivalent to a dense motion field in between two consecutive images of a video sequence. Therefore, it is justified to use the same principles of motion estimation and

compensation for disparity estimation and compensation. The prediction error or residual error will be then encoded. Smolic [67] states that some specific differences between motion compensation and disparity compensation need to be considered. The statistics of disparity vector fields is different from the motion vector fields. Disparities are biased and relatively large. Zero disparity means a very large depth of the corresponding point in 3D, while 3D points close to the camera may have a very large disparity value. In general, temporally adjacent images of a video sequence tend to be more similar than views of a stereo pair at practical frame rates. Some other differences are caused by the disocclusion effects, where the content that is visible in one image is occluded in the other and cannot be predicted. The incorrect white and colour balance between the stereo pair caused by the scene lighting and surface reflectance effects also contributes to the differences.

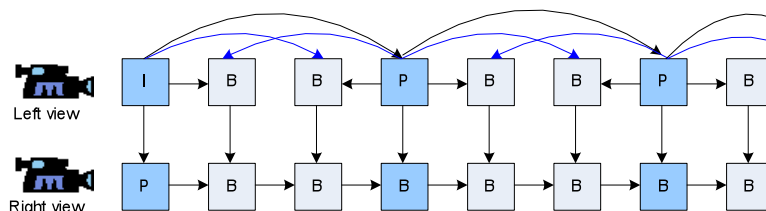


Figure 2.10: Prediction in H.262/MPEG 2 video multi-view profile [68]

The combination of inter-view and temporal prediction is the basic principle for efficient compression of conventional stereo videos. A corresponding standard specification has been defined in ISO/IEC Technical Report [68]. The multi-view Profile Standard is shown in Figure 2.10, where I is the intra-coded pictures and P or B are the inter-coded pictures. The left view is encoded without reference to the right view by using the standard MPEG 2 to ensure the compatibility with Main Profile of H.262/MPEG 2 Video. For the right view, inter-view prediction is allowed in addition to temporal prediction. A significant increment of compression efficiency is achieved with the inter-view prediction in H.262/MPEG 2 multi-view video coding. Research on compression of conventional stereo video has continued into several directions, including designing better and efficient inter-view prediction structures. Algorithms have been designed on current video codec such as H.264/AVC.

2.6.2 Video Plus Depth Data

Another option to classical stereo video is to transmit a video signal and a per sample depth map. From the video and depth information, a stereo pair can be rendered at the decoder [69]. The functionality can be extended since it enables head motion parallax viewing if the user's head motion is tracked. Furthermore, this format is interesting from the compression efficiency point of view. Per sample depth data can be regarded as monochromatic and luminance video signal.

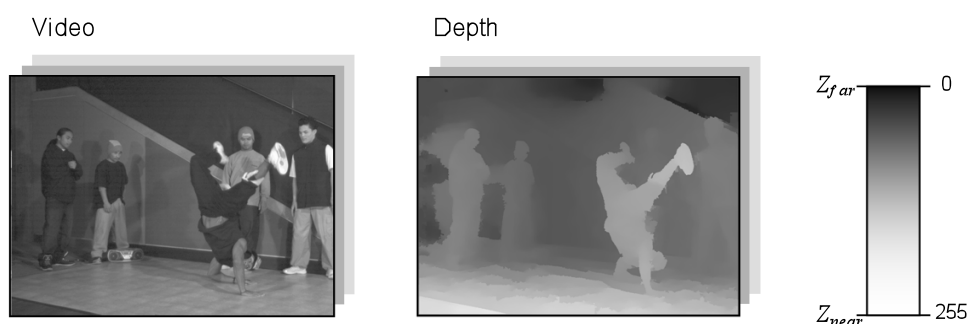


Figure 2.11: 3D data representation format consisting of regular 2D colour video and 8-bit depth images [43]

Figure 2.11 illustrates the video plus depth format with an image and its associated per sample depth map [43]. The depth is restricted to a range between two extremes Z_{near} and Z_{far} indicating the minimum and maximum distance of the corresponding 3D point from the camera respectively. The depth range is linearly quantized with 8-bit, with the value of 255 for the closest point and the value of 0 for the most distant point. With that, the depth map in the right of Figure 2.11 is specified, resulting in a grey scale image. These grey scale images can be converted into either YUV 4:0:0 format video signal or YUV 4:2:0 format, where the luminance component corresponds to the grey scale depth values and the chrominance is set to a constant value. The resulting standard video signal can then be processed by any state-of-the-art video codec. Results from the European Advanced Three-Dimensional Television System Technologies (ATTEST) project [69] have shown that depth data can be very efficiently compressed with several video codecs like MPEG 2, MPEG 4 and H.264/AVC.

A common problem of the video plus depth format is content creation known as the generation of depth information. Cameras that automatically capture per pixel depth with the video are available and are being further enhanced. However, the current quality of the captured depth fields is still limited. Algorithms for depth and disparity

estimation have gained research interest in computer vision literature and powerful solutions are needed. A fully automatic, accurate and reliable depth capturing system is still to be developed.

2.6.3 Multi-view Video Coding

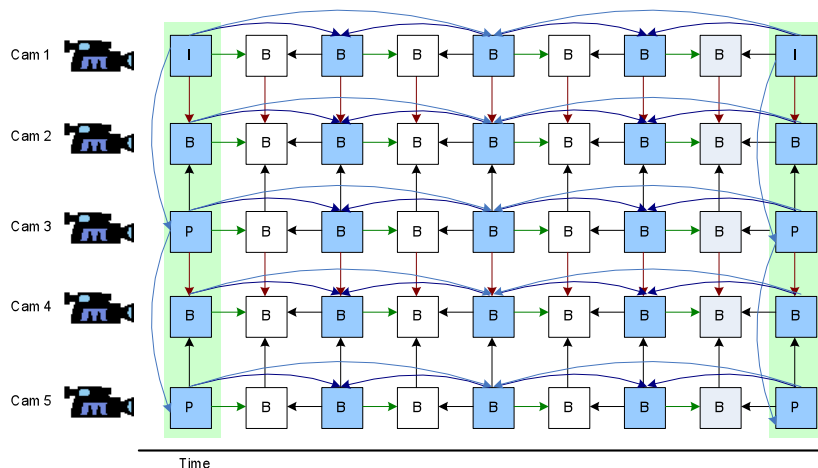


Figure 2.12: Temporal/inter-view prediction structure for MVC [11]

Multi-view video compression algorithms should reduce redundancy in information from multiple views as much as possible to provide a high degree of compression. As the multi-view video captures the same dynamic 3D scene, the similarities or redundancies exist among the images within temporal and inter-view video images. Exploiting redundancies among the multi-view video images is the key for efficient compression. The redundancy in multi-view video streams consists of inter-view redundancy (between adjacent camera views) and temporal redundancy between temporally successive images of each video [11]. These redundancies can be exploited for combined temporal/inter-view prediction as shown in Figure 2.12. Images are not only predicted from temporally neighbouring images, but also from corresponding images in adjacent views. Other type of redundancies includes the transform redundancy and the redundancy of the human visual system [1]. The temporal redundancies can be exploited with motion compensated techniques like a normal one view video streams such as block matching, adaptive block size and bidirectional predicted picture techniques.

A simpler version of temporal and inter-view prediction structure is shown in Figure 2.13. The classification of the redundancies is based on the normal arrangement of multi-view video images into a Matrix of Pictures (MOP) [70]. Each row holds

temporally successive pictures of one view, and each column consists of spatially neighbouring views captured at the same time. It depicts a matrix of pictures for $N=4$ image sequences, each composed of $K=4$ temporally successive pictures. $N=4$ views form a Group of Views (GOV), and $K=4$ temporally successive pictures form a temporal Group of Pictures (GOP). For example, the images of the first view sequence are denoted by $x_{l,k}$ with $k = 1, 2, \dots, K$.

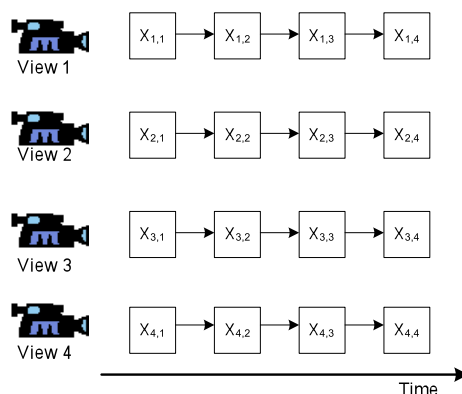


Figure 2.13: Matrix of Pictures (MOP) for $N=4$ image sequences, each comprising $K=4$ temporally successive pictures [11]

Encoding and decoding each view of a multi-view video data separately, referred to as simulcast coding; can be done with any existing standard, such as H.264/AVC, where each camera view of the sequence is coded independently like a normal video stream [43, 71-73]. The process of simulcast coding can be illustrated from Figure 2.13. The first frame of each view is coded as an *I-frame*, while the remaining frames are predicted as *P-frames*. This would be a simple, but inefficient way to compress multi-view video sequences, due to not exploiting the inter-view redundancies.

Disparity compensation is the most popular approach and the most straightforward extension of general single view video coding. In this approach, pictures of other views are treated in the same way with encoding target view, used as reference pictures for predictive coding. The difference in terms of motion compensation is only the domain treated; motion is handled in the temporal domain whilst disparity is handled in the inter-view domain. Therefore, in this approach, disparity information like motion vectors and prediction error, are encoded and transmitted to the decoder side. Many prediction structures have been proposed such as a group of GOP prediction schemes, hierarchical B pictures, checkerboard decomposition, sequential view prediction and so on [13]. Some of the predictive coding is presented in Appendix B.

The simplest approach to disparity compensation is the use of block matching algorithms similar to those for motion compensation. These techniques offer the advantage of not requiring knowledge of the geometry of the underlying 3D objects. However, this approach fails to compensate correctly if the cameras are sparsely distributed. More advanced approaches to disparity compensation are Depth Image Based Rendering (DIBR) algorithms [74]. With this technique, the given viewpoint image is compensated more accurately even when the cameras are sparsely distributed. These techniques rely on accurate depth images. The hybrid techniques that combine the advantages of both approaches are effectively exploiting inter-view redundancies [75].

2.6.4 MVC Test and Analysis

In this section, a multi-view video coding simulation based on H.264/AVC will be presented. The coding scheme processed the frames of sequences captured by multiple cameras from a scene. The codec is based on the JM H.264/AVC Version 10 [12], which is the reference software for MVC. It uses prediction structure of hierarchical B pictures for each view in the temporal direction as shown in Appendix B. The simulcast coding is achieved by coding each view sequence separately using H.264/AVC standard. The multi-view video coding is based on five modes of operation in MMRG H.264 Multi-view Extension [76] (described in Appendix B).

In order to evaluate the performance of the coding schemes, the multi-view video sequences were captured using four cameras positioned in a regularly spaced linear array. It contains four views of 50 frames each at 15 fps. The sequences were set to produce CIF size sequences. The quality of the encoded sequences was measured by the average PSNR of their frames.

The parameters set for the encoding process are shown in Table 2.1. The results based on simulcast coding are given in Table 2.2. Meanwhile, Table 2.3 provides the simulation results by using different reference modes of the multi-view video coding H.264/AVC. The multi-view images with YUV 4:2:0 formats are coded with the parameter of macroblock search range 16 and the total number of frames is 200. From the results, it shows that the different reference mode produced a difference in performance. The bit rate for Mode 4 is higher compared to the remaining modes even though the SNR of each mode is almost similar. The total encoding time for the Mode 5

gives faster encoding time, which is 9 fps. The total bits for the simulcast coding seem to be smaller compared to the output in inter-view coding because it was only for a single sequence. The total number of bits will increase due to the summation of all the compressed data 4 views to transmit or store. This is an inefficient way to compress the multi-view video because it does not exploit the redundancies between the multiple views. Based on this test evaluation, it is shown that the MVC outperforms the simulcast coding while maintaining the quality of the reconstructed pictures. The performance of a new H.264/AVC based multi-view video coding scheme with different four modes of operation also has been presented by Akbari [77], where the experimental results have been shown that the proposed coding scheme outperforms the simulcast H.264/AVC coding.

Table 2.1: Input Parameter of the H.264/AVC codec

Parameter	Input
Image format	352 x 288
Search range	16
Sequence type	IPPP
Motion Estimation Scheme Search	Full
Search range restrictions	None

Table 2.2: Simulation results for simulcast coding

Camera Views	View 1	View 2	View 3	View 4
Total encoding time (fps)	15	15	15	14
Total ME time for sequence (sec)	737.63	750.38	764.66	705.77
SNR Y (dB)	39.82	40.75	41.14	41.66
Total bits	325,192	211,160	187,752	150,312
Bit rate (kbit/s)	195.12	126.70	112.65	90.19

Table 2.3: Simulation results for the multi-view coding

Reference Mode	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Total encoding time (fps)	7	3	4	4	9
Total ME time for sequence (sec)	1343.47	549.73	807.38	698.49	1768.55
SNR Y (dB)	40.53	40.51	40.50	40.92	40.49
Total bits	905,528	939,104	936,696	1,140,888	937,120
Bit rate (kbit/s)	135.83	140.87	140.50	171.13	140.57

2.7 Conclusion

This chapter has introduced the multi-view imaging from the basic stereo point of view. Stereoscopic concepts define two images corresponding to the same scene with slightly different perspectives. The subtle differences between the left and right images known as disparity, perceived by the human brain as the three-dimensional view of the scene. This idea leads to the technique of inferring depth by using two or more cameras that include the stereo vision, dense stereo algorithms and stereo vision applications. The principle that relates two 2D images to a 3D representation can be extended to multiple views.

Then the basic fundamental of stereo vision serves as the mathematical framework for the multi-view imaging and 3D graphics with the relationship to the 3D world and its corresponding position in a 2D image. The pinhole camera model provides the platform for a camera calibration process to determine the internal and external camera parameters that are useful in camera rectification and triangulation processes.

The applications of multi-view imaging include the stereoscopic displays, free-viewpoint video, video editing and special effects. Stereoscopic display allows the viewer to perceive the depth of the scene. Several display technologies has been developed including the polarized displays, parallax barrier displays and lenticular displays. Current 3DTV enables the viewer to watch the 3D scene by using an active glass system. Meanwhile, the free-viewpoint video application provides the ability for users to select and control any viewpoint of the video scene interactively. It offers the same functionality that is known in the 3D computer graphics but targeted on real world scenes as captured by real cameras.

The 3D video technologies enable various applications that can be integrated into a single 3D video system. Several 3D video systems have been introduced to enable the 3DTV and free-viewpoint video applications. The 3D video system consisted of several components includes the camera acquisition, calibration, rectification, stereo correspondence, compression and rendering. Camera calibration used to establish the relationship between 3D-world coordinates and their corresponding 2D image coordinates by defining the external and internal camera parameters. The rectification is a process to facilitate the analysis of a stereo pair of images by removing the lens distortion and turns the stereo pair in standard form. This process is useful for disparity

and depth estimation process in the stereo matching algorithm.

An efficient compression algorithm is important in the multi-view video due to the huge amount of data for storage and transmission. This chapter has discussed an overview of multi-view compression algorithms, which includes the conventional stereo video coding, video plus depth data and multi-view video coding. Multi-view video compression algorithms should reduce redundancy in information from multiple views as much as possible to provide a high degree of compression. As the multi-view video captures the same dynamic 3D scene, the similarities or redundancies existed among the images. The redundancy in multi-view video streams consists of inter-view redundancy (between adjacent camera views) and temporal redundancy between temporally successive images of each video. The redundancies between the camera views (inter-view) and temporal correlation exploited with the existing standards such as MPEG-2 and H.264/AVC.

In the next chapter, stereo matching and view synthesis algorithms will be reviewed. From the stereo matching, the disparity depth map can be measured and used as the next stage of 3D image and video processing, which is the inter-view synthesis algorithm. The inter-view synthesis refers to the generation of a view of a scene from an arbitrary or novel viewpoint.

Chapter 3

Stereo Matching and View Synthesis Algorithms

3.1 Introduction

In the state-of-the-art research in 3D vision, stereoscopy is one of the most significant and active research areas and has been widely used in photography and the film making industry [37]. Recently, it has received more attention because the necessary technology has matured significantly, allowing both stereoscopic recording and playback within reasonable constraints. Disparity estimation in the stereo matching can be used in the intermediate view synthesis algorithms implementation. The view synthesis composes a new image located in the virtual viewpoint between the stereo image pairs. The chapter describes the related research works and current trends in stereo matching and view synthesis algorithms. Also in here we will cover in detail the main building blocks of the algorithms, which includes cost computation, cost aggregation, intermediate view synthesis and rendering that will be used in layered depth map in the next chapter.

This chapter is divided into two main parts: stereo matching algorithms and novel view synthesis algorithms, which are organized in seven sections. In Section 3.2, an introduction to the fundamentals of the three-dimensional (3D) images based on the stereo matching algorithms will be described. It presents the stereo matching classification to obtain the stereo correspondence pixels between the stereo images based on the sparse or dense disparities techniques. The stereo matching system and its main components will be introduced through stereo disparity estimation computation. From the stereo matching, the disparity depth map can be measured and used as the next stage of the 3D image and video processing, such as in the inter-view synthesis algorithm.

Section 3.3 surveys the stereoscopic creations that are differentiated by the underlying models of the 3D scene representation, where two major groups can be identified: based on a geometric scene representation and image-based systems. Between the two major groups, there is a range of methods with varying proportions of geometric and image information such as the layered depth image-based representations. View synthesis algorithms are discussed in Section 3.4. Section 3.5 covers the image-based rendering algorithm to generate new views of scenes from an arbitrary or novel viewpoint. The layered image-based rendering will be discussed in Section 3.6. Section 3.7 provides the performance evaluation used for the image view synthesis and finally Section 3.8 concludes the chapter.

3.2 Stereo Matching Algorithms

The main aim of stereo matching algorithms is to find homologous points in the stereo pair [46]. It is concerned with the matching of points between a pair of pictures of the same scene. The matching points reside on corresponding horizontal lines upon calibrated stereo setup. The disparity is calculated as the distance of these points when one of the two images is projected onto the other. The disparity values for all the image pixels comprise the disparity map. Once the stereo correspondence problem is solved, the depth of the scenery can be estimated. The disparity and depth are required in applications such as 3D reconstruction, virtual reality, robot navigation and many other aspects of production, security, defence, exploration and entertainment.

Matching methods can be classified into two approaches: sparse and dense. The sparse outputs can be obtained with feature based matching methods, which are matching the two images based on matching segments or edges. The disadvantage of this approach is counterbalanced by the accuracy and speed obtained. In feature based matching, the algorithms select feature points independently in the two images, then match them using tree searching, relaxation, maximal detection or string matching [78]. Template matching also provides a sparse output, which selects templates in one image, usually patches with some texture information and then searches for corresponding points in the other image using some similarity measure. The algorithms in this class tend to be slower as the search is less constrained. The search can be simplified with the rectification process when the images are rotated and projected onto the same plane.

In order to categorize and evaluate the stereo correspondence algorithms that produce dense output, a taxonomy and evaluation has been proposed by Scharstein and Szeliski [22]. In general, the stereo matching algorithms consist of four steps, which are (a) matching cost computation, (b) cost aggregation, (c) disparity computation or optimization and (d) disparity refinement, which can be summarized in Table 3.1. However, not all stereo algorithms take all four steps depending on individual implementation and requirements. With respect to the combination of these steps, stereo algorithms that generate dense depth measurements can be divided into two classes, namely global and local algorithms.

Table 3.1: Stereo Matching Algorithm Components

Step	Components
a	Matching cost computation
b	Cost aggregation
c	Disparity computation or optimization
d	Disparity refinement

A global algorithm (energy-based) determines the optimal disparity map by minimizing a global energy function defined by pixel matching and a smoothness constraint. Global algorithms rely on iterative schemes that carry out disparity assignments on the basis of the minimization of a global cost function [25]. Many algorithms in this category consist of steps (a), (c) and (d). In order to solve the optimization problem, many algorithms adopt Graph Cuts [79], Belief Propagation [80, 81] and Dynamic Programming [82-84]. Since the energy function is defined on all the pixels of the image, global methods are less sensitive to local ambiguities (occlusions, textures) than local methods. Although these algorithms yield accurate and dense disparity measurements, they exhibit very high computational and time costs, which render them unsuitable to real-time applications.

On the other hand, local algorithms, also known as area-based algorithms [85] are typically faster than the global approaches and have a lower memory footprint. However, they also have reduced accuracy compared to global state-of-the-art algorithms. The local methods calculate the disparity at each pixel on the basis of the photometric properties of the neighbouring pixels. This approach utilizes the window matching techniques to determine the optimal disparity map [32, 86, 87]. In local methods, the correspondence of a pixel is decided by the relationships between the

pixels in its neighbourhood (usually defined as a rectangular window) and those in the neighbourhood of the corresponding pixel in the other image.

Typically, the area-approach based algorithms have all four steps. The disparity computation at a given point depends only on intensity values within a finite window. The disadvantage of this method is that it easily affected by local ambiguous regions such as occlusions, textures and homogenous regions. Compared to global algorithms, local algorithms yield significantly less accurate disparity maps but can run fast enough to be deployed in many real-time applications [85]. The summary of matching algorithms can be illustrated in Figure 3.1.

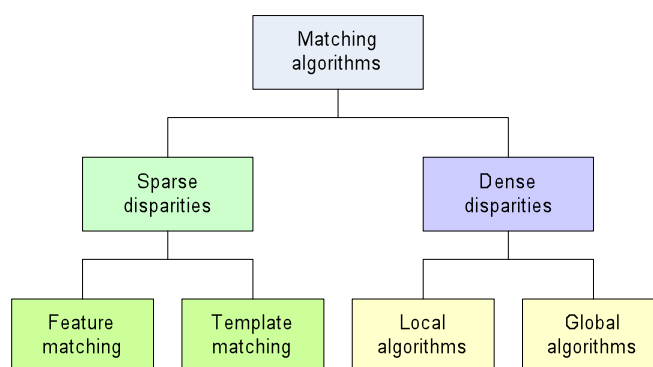


Figure 3.1: Classification of matching algorithms

There are many other methods that do not fall under these categories, such as the depth disparity estimation through wavelet transform [88], neural networks and cellular automata. Most of the work done involves a theoretical description of the algorithm, a software development stage and finally the testing of the algorithm with the use of a general-purpose personal computer. This methodology results in considerable running times. However, this is not the case when the objective is the development of Simultaneous Localization Automatic Mapping (SLAM) or virtual reality systems. Such tasks require real-time, efficient performance and demand dedicated hardware and consequently specially developed and optimized algorithms.

The Scale-Invariant Feature Transform (SIFT) [89] and Speeded-Up Robust Features (SURF) [90] operators also has been used to find the point correspondences between two images of the same scene or object [91]. For example, a novel framework for matching video sequences using the spatiotemporal segmentation of videos is presented by Basharat [92], where the point trajectories are computed using the SIFT operator. In radiometric applications as presented by [93], the mutual information and SIFT

descriptor are combined to devise a robust and accurate stereo system. Due to the ability of SIFT to extract feature points, it has been used in the stereo vision by using two Point-Tilt-Zoom (PTZ) cameras system [94] to obtain multi-view angle and multi-resolution information. The calibration and configuration are really complicated and require further research to improve the depth map estimation in the system.

Meanwhile, the SURF operators present a novel scale and rotation invariant detector and descriptor, which not only can be very efficiently computed but also has comparable performance compared to other existing schemes with respect to repeatability, distinctiveness, and robustness. The framework is tested in two challenging applications: camera calibration treated as a special case of image registration and object recognition. However, the length of the descriptor is a major obstacle for real-time applications and mobile platforms where the computation time and storage capacity is limited. It has also been shown that the high dimensional SIFT and SURF descriptors suffer from a numerical instability known as the curse of dimensionality [95]. SURF can be computed efficiently at every pixel but introduce artefacts that degrade the matching performance [96].

In large baseline matching, the algorithms generate significantly different images. It is quite challenging to determine correspondences between different images because the cameras' relative displacement or rotation is large. As a consequence of the significant differences between the images, direct correlation-based matching fails at many more locations than in small-baseline stereo. The images of large baseline stereo pair lead to significant disparities and tend to present considerable amounts of relative distortions and occlusions. The next sections (Section 3.2.1, 3.2.2 and 3.2.3) will discuss each step of the stereo matching algorithm in detail.

3.2.1 Matching Cost Computation and Aggregation

Cost computation or depth estimation aims at calculating the structure and depth of objects in a scene from a set of multiple views or images [37]. The main challenge is to localize corresponding pixels or the point-correspondences in the multiple views that identify the same 3D scene point. The most common pixel-based matching costs include Absolute intensity Differences (AD) and Square intensity Differences (SD). In the video processing community, these matching criteria are referred to as the Mean Absolute Difference (MAD) and Mean Squared Error (MSE) measures [22]. The pixel-based

matching costs for AD and SD are given by the following equations [97, 98] respectively:

$$AD = e(x, y, d) = |I_R(x, y) - I_T(x + d, y)| \quad (3.1)$$

$$SD = e(x, y, d) = (I_R(x, y) - I_T(x + d, y))^2 \quad (3.2)$$

where I_R and I_T are the reference and target images respectively for the location of x and y , and d is the disparity.

Other conventional matching costs, also known as the area-based matching are determined based on matching windows of pixels by using similarity metrics such as the Sum of Absolute Differences (SAD), Sum of Square Differences (SSD) or normalized correlation techniques. In order to determine the correspondence of a pixel in the left image using a similarity metric function, the window costs are computed for all candidate pixels in the right image within the search range. The pixel in the right image that gives the minimum window cost is the corresponding pixel of the left image. Local and window-based methods aggregate the matching cost by summing or averaging over a support region in the Disparity Space Image (DSI) [22].

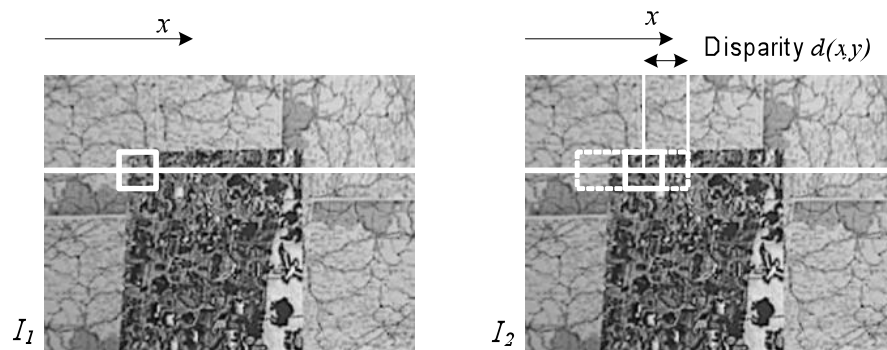


Figure 3.2: The disparity estimated by searching the most similar block along the horizontal epipolar line [37]

A simple disparity estimation algorithm can be described as follows. Consider a left and right rectified image denoted by I_1 and I_2 respectively. In order to perform disparity estimation, it is necessary to establish the point-correspondences (p_1, p_2) for each pixel. By selecting the pixel p_1 as a reference, a straightforward strategy consists of searching for the pixel p_2 that corresponds to pixel p_1 along the epipolar line. The searching process is simplified because the images are rectified and the search for the point-correspondence is only along the horizontal raster scanlines. In order to limit the search area, a maximum disparity value d_{max} is defined. The similarity between pixels p_1 and p_2

is measured using a matching block (window) W surrounding the pixels as illustrated in Figure 3.2. By employing a similarity measure like SAD for block comparison, the disparity d of a pixel at position (x, y) in view I_l can be written as [99]:

$$d(x,y) = \arg \min_{d_{\min} \leq d \leq d_{\max}} \sum_{(i,j) \in W} |I_1(x+i, y+j) - I_2(x+i-d, y+j)| \quad (3.3)$$

The previous operation is repeated for each pixel so that a dense disparity map is obtained. The main attraction of this approach is its computational simplicity since the technique relies on a block matching procedure. However, this simple technique results in inaccurately estimated disparity values. For example, a change of illumination across the views introduces ambiguities. The problem of depth estimation has been intensively investigated in the computer vision research community [22]. Some approaches to overcome this problem will be discussed in Section 3.2.2 through different component of stereo matching algorithms. In general, the basic stereo matching algorithm can be summarized in the following steps:

For each epipolar line

For each pixel in the left image

- compare with every pixel on the same epipolar line in the right

- find pixel with minimum match cost

In brief, the matching process involves computation of the similarity measures for each disparity value, followed by an aggregation and optimization step. The images can be matched by taking either left image as the reference (left-to-right matching, also known as direct matching) or right image as the reference (right-to-left matching, also known as reverse matching) [22]. The computation of window cost is given by the following equation [98, 100, 101]:

$$\text{Sum of Absolute Differences (SAD)} = \sum_{(i,j) \in W} |I_1(x,y) - I_2(x+i, y+j)| \quad (3.4)$$

$$\text{Sum of Squared Differences (SSD)} = \sum_{(i,j) \in W} (I_1(x,y) - I_2(x+i, y+j))^2 \quad (3.5)$$

$$\text{Normalized Cross Correlation (NCC)} = \frac{\sum_{(i,j) \in W} I_1(x,y) \cdot I_2(x+i, y+j)}{\sqrt{\sum_{(i,j) \in W} I_1^2(x,y) \cdot \sum_{(i,j) \in W} I_2^2(x+i, y+j)}} \quad (3.6)$$

$$\text{Sum of Hamming Distances (SHD)} = \sum_{(i,j) \in W} I_1(i,j) \text{ bitwise XOR } I_2(x+i, y+j) \quad (3.7)$$

SAD is one of the simplest of the similarity measures which is calculated by subtracting pixels within a square neighbourhood between the reference image I_1 and the target image I_2 followed by the aggregation of absolute differences within the square window, and optimization with the Winner-Take-All (WTA) strategy [102]. If the left and right images exactly match, the resultant will be zero.

In SSD, the differences are squared and aggregated within a square window and later optimized by WTA strategy. This measure has a higher computational complexity compared to SAD algorithm as it involves numerous multiplication operations. Normalized Cross Correlation is even more complex to both SAD and SSD algorithms as it involves numerous multiplication, division and square root operations. SHD is normally employed for matching census-transformed images (can be used on images that have not been census transformed) by computing bitwise-XOR of the values in the left and right images, within a square window. This step is usually followed by a bit-counting operation, which results in the final Hamming distance score.

The pixel in the right image that gives the best window cost that is the minimum SSD or SAD value or the maximum correlation value indicates the corresponding pixel of the pixel in the left image. In this research, window cost calculation is performed based on SAD and SSD algorithm. In direct search, it requires to compute the window costs, with the SAD or SSD values for all candidate pixels within the search range, $-d_{max}$ to $+d_{max}$. Another matching costs including non-parametric [103] and Mutual Information (MI).

If the correlation is separable in x and y , with all pixels within the correlation window equally weighted, an efficient implementation is possible. The Sum of Absolute Differences (SAD) is chosen because it performs better than the sum of squared differences in the presence of outliers and it has a smaller computational complexity than normalized correlation measures [104]. In this application, the exposure and white balance of the cameras is controlled to minimize the difference of brightness and contrast between the images.

Correlation based matching typically produces dense depth maps by calculating the disparity at each pixel within a neighbourhood. This is achieved by taking a square window of a certain size around the pixel of interest in the reference image and finding the homologous pixel within the window in the target image, while moving along the corresponding scanline. The goal is to find the corresponding (correlated) pixel within a

certain disparity range d that minimizes the associated error and maximizes the similarity.

Window-based stereo matching technique is widely used due to its efficiency and ease of implementation. The simplest cost aggregation method is by using Fixed Window (FW). However, Barnard and Fischler [105] point out a problem in the selection of a window with fixed size and shape. The fixed window cost aggregation method ignores the depth discontinuities and does not deal with uniform areas and repetitive patterns. The bigger the window size, the higher the chance for a correct match but with a drawback of quality loss at disparity discontinuities such as object borders (borders become broader). Small windows increase the quality at borders and the localizing of matches is more accurate, but they can cause more false matches at difficult areas. Many researchers proposed adaptive window methods using windows of different shapes and size depending on local variations of intensity and disparity [106]. However in adaptive window algorithms, the computation time is relatively higher than the fixed window algorithms. To overcome this problem and to achieve high gain in accuracy with less computation time, Chowdhury [107] proposed an average disparity estimation method.

Several cost aggregation methods aimed at improving the robustness of stereo correspondence within local and global algorithms have been proposed. In [108], the classification and evaluation of cost aggregation strategies for stereo correspondence has been presented. Most of the techniques compute the stereo matching pixels based on position, shape and weights strategy. The evaluation comprises fixed window, shiftable window [22], multiple window, variable window [109], adaptive weights, bilateral filtering and segmentation-based [110] strategy. From the qualitative evaluation, it shows that the segmentation-based cost aggregation strategy adapts very well as it supports along depth borders as well as in presence of low-textured regions.

In spite of its limitation, fixed window cost aggregation is widely adopted in practice for real time applications. According to Mattoccia [23], the fixed window is easy to implement, faster execution, runs in real-time on standard processors [97], has limited memory requirements and low power consumption in hardware implementation. The fast stereo matching using general purpose processor by Stefano [25] was performed and developed by using SAD computational optimization. Stefano outlines the optimization techniques to avoid redundant calculations. The computation scheme can

be used in any similarity metric functions such as SSD and NCC. Therefore, the basic fixed window-based similarity metric matching cost is the best approach for faster execution and it can be easily adapted in parallel processor instruction technology for future implementation.

3.2.2 Disparity Computation and Optimization

In local methods, the emphasis is on the matching cost computation and on the cost aggregation steps [22]. In order to compute the final disparity one chooses at each pixel the disparity associated with the minimum cost value. The most popular disparity optimization is the simple Winner-Take-All (WTA) strategy. It is normally implemented by using a window-based cost computation and aggregation method such as SAD. A limitation of this approach is that uniqueness of matches is only enforced for one image (the reference image), while points in the other image might get matched to multiple points [22].

In contrast, global methods perform almost all of their work during the disparity computation phase and often skip the aggregation step [22]. Many global methods are formulated in an energy minimization framework. The objective is to find a disparity function d that minimizes a global energy. Typically, the cost function takes the following form [22, 98]:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (3.8)$$

The first term is the data term, $E_{data}(d)$ which evaluates the pixel matching with disparity configuration d and λ is a parameter that adjusts smoothness of the result. By using the disparity space formulation [22, 98]:

$$E_{data}(d) = \sum_{(x,y)} C(x,y,d(x,y)) \quad (3.9)$$

where C is the initial or aggregated matching cost Disparity Space Image (DSI).

The second term in (3.8), $E_{smooth}(d)$ is the smoothness term that takes a low value when the disparity value at a pixel is similar to those of its neighbours. The term can be defined as follows [22, 98]:

$$E_{smooth}(d) = \sum_{(x,y)} \rho(d(x,y) - d(x+1,y)) + \rho(d(x,y) - d(x,y+1)) \quad (3.10)$$

where ρ is some monotonically increasing function of disparity difference.

In regularization-based vision ρ is also defined as a quadratic function, which makes d smooth everywhere and may produce poor results at object boundaries. The terms in E_{smooth} can also be made to depend on the intensity differences as defined in the following [22]:

$$\rho_d(d(x,y) - d(x+1,y)) \cdot \rho_I(\|I(x,y) - I(x+1,y)\|) \quad (3.11)$$

where ρ_I is some monotonically decreasing function of intensity differences that lowers smoothness costs at high intensity gradients.

This idea encourages disparity discontinuities to coincide with intensity or colour edges and appears to account for some of the good performance of global optimization approaches.

Some relevant approaches in the disparity computation and optimization are Graph Cuts [111], Belief Propagation [81] and Cooperative Optimization [112]. A detailed comparison of relevant energy minimization methods can be found in [113]. Finding stereo correspondence with Graph Cuts formulates the correspondence problem as the search for the maximum flow of a weighted graph. This graph has two special vertices, the source and the sink. Between those are nodes, which are connected by weighted edges. Each node represents a pixel at disparity level and is associated with the according matching costs. Each edge has an associated flow capacity that is defined as a function of the costs of the node it connects. This capacity defines the amount of flow that can be sent from source to sink. The maximum flow is comparable to the optimal path along a scanline in dynamic programming, with the difference that it is consistent in three dimensions. The computation of the maximum flow is extensive and cannot be used for real time applications.

Another global disparity optimization approach is Belief Propagation [80, 81, 113-115]. This iterative strategy uses rectangular Markov random fields for assigning the best matching disparities to the pixels. Each node is assigned to a disparity level and holds its matching costs. The belief (probability) that this disparity is the optimum arises from the matching costs and the belief values from the neighbouring pixels. At every iteration cycle, each node sends its belief value to all four connected nodes. The belief value is the sum of the matching costs and the received belief values. The new belief value,

which is the sum of the actual and the received values, is saved for each direction separately. This is done for each disparity level. Finally, the best match is the one with the lowest belief values defined by a sum over all four directions.

A different class of global optimization algorithms is that based on Dynamic Programming (DP) [82-84]. Dynamic programming was first used for stereo vision in sparse and edge-based methods. The approaches expanded on the dense scanline optimization problem. The strategy of DP is to find the optimal path (minimum cost path) through all possible matching costs between two corresponding scanlines. The ordering constraint, which means that pixels in the reference image have the same order as their correspondences in the matching image, specifies the possible predecessors of all matches. The path with the lowest matching and joining costs is chosen recursively. This leads to a path through the possible matches that implies a left/right consistency check. Partial occlusion is handled explicitly by assigning a group of pixels in one image to a single pixel in the other image. The problem with DP includes the selection of the right cost for occluded pixels and the difficulty of enforcing inter-scanline consistency. The main drawback of dynamic programming is that it only considers horizontal smoothness constraints. Therefore, the disparity maps obtained based on DP suffer from horizontal streaking artefacts but they are computationally inexpensive.

3.2.3 Disparity Refinement

The post-processing step for the stereo matching algorithm is the disparity refinement has received a lot of attention in recent years. Most global-based matching algorithms compute disparities as integer values and need to be refined. In this step, raw disparity maps computed by correspondence algorithms contain outliers that must be identified and corrected. Moreover, if the disparity maps computed at discrete pixel level, disparity refinement step is necessary to remove errors in the disparity maps. Several approaches aimed at improving the raw disparity maps computed by stereo correspondence algorithms such as sub-pixel interpolation [116], image-filtering techniques [117], Bidirectional Matching (BM) [24] and Single Matching Phase (SMP) [85]. Sub-pixel disparity is obtained by interpolating the three matching costs with a parabolic function. This technique is reasonably accurate but computationally expensive if performed directly at matching cost computation stage. The disparity map can be simply refined by means of image filtering techniques without explicitly enforcing any

constraint about the underlining disparity maps. The common image filtering operators are median filtering, morphological operators and bilateral filtering [117].

Bidirectional matching [24] is widely used to detect outliers in stereo based on local or global approaches. It is also known as the Left-Right Consistency Check (LRCC). The correspondence problem is solved in two steps: firstly, by assuming left image as reference, $d_{LR}(x, y)$ and secondly by assuming right image as reference, $d_{RL}(x, y)$. The disparity values that are not consistent between the two maps are classified as outliers enforcing, with threshold T typically set to 1 in the following equation [24, 104]:

$$|d_{LR}(x,y) - d_{RL}(x + d_{LR}(x,y),y)| < T \quad (3.12)$$

The advantages of LRCC are that it is useful for detecting occlusions, preserves depth discontinuities and also effective for detecting outliers in ambiguous regions. However, this approach is computationally expensive because of two matching phases required. The Single Matching Phase (SMP) approach [85] aims at detecting unreliable disparity assignments using a more computationally efficient technique. It uses a single matching phase that explicitly enforces the uniqueness constraint. The algorithm dynamically updates the disparity map when the uniqueness constraint is violated. Therefore, it is quite effective compared to the LRCC and suitable for efficient real-time standard processor implementation. Even though the proposed algorithm provides exceptional accurate disparity depth map, some of it suffered with complexity for the implementation particularly for real-time implementation.

Middlebury Stereo Evaluation Page developed by Scharstein and Szelinski [22] provides some common benchmark datasets and evaluation systems for all the researchers to analyze and examine their methods objectively with a standard parameters. Based on the Middlebury ranking, it shows that many stereo algorithms adopt the segment-based method [81, 118-120]. Segment-based methods are widely accepted for effectiveness of disparity map refinement. Two fundamental assumptions for segmentation-based outliers identification and replacement are that the disparity within each segment varies smoothly and that each segment can be approximated with a plane. The robust plane fitting of disparity measurements can be performed within a global energy minimization framework, such as by using Random Sample Consensus (RANSAC) [121] and histogram voting [112]. Almost all stereo algorithms adopt a Mean-Shift [110] method as their colour segmentation strategy.

Other methods of disparity refinement to remove large errors and to enforce local consistency in the disparity results are the Intensity Consistent (IC) and Locally Consistent (LC) [122] disparity selection techniques. The IC technique relies on segmentation and is particularly effective to solve the problem of propagation of disparities from textured foregrounds to non-textured regions, one of the major problems in local stereo matching algorithms. The locally consistent disparity selection technique, by enforcing local consistency between neighbouring points, has proven to be effective in recovering wrong disparity assignments in uniform regions as well near depth discontinuities. Nevertheless, this technique is unable to recover from large erroneous areas. The erroneous patches are typically caused by homogeneous regions in the stereo pair.

Therefore, we propose a disparity refinement pipeline in which the resulting disparity maps of local-based stereo matching are refined with the colour segmentation method and morphological techniques to solve large erroneous areas and typically enforcing local consistency by mean of morphological. A detailed description of the proposed disparity refinement technique is described in Chapter 5.

3.2.4 Summary of Stereo Matching Algorithms

In this section, a summary of the stereo matching algorithms is presented. Researchers are making efforts in all fields of stereo and image view synthesis that include stereo correspondence matching, 3D scene representation and rendering. Beside the taxonomy and evaluation proposed by Scharstein and Szelinski [22], the image matching algorithms has also been discussed intensively by Cyganek [98]. The compilation of the research works in this field indicates the important and necessities of stereo matching in the image processing and 3D vision.

Currently, the proposed stereo matching methods in the Middlebury Stereo Page contains more than a hundred submissions. The list is updated constantly and provides the state-of-the-art on the latest algorithm on the stereo matching. It is hard to select and define which is the best algorithm since each algorithm is developed for a particular application with a trade-off between accuracy and speed. Therefore, the selected approaches presented in Table 3.2 are based on some representative stereo matching algorithm and their corresponding taxonomy: the matching cost, aggregation, optimization and refinement techniques used. Not all the presented algorithms take all

four stereo matching components (given in Table 3.1, Section 3.2). As discussed in Section 3.2, the selection of steps determines the class of dense disparities either locally or globally. For example, many algorithms classified into global-based methods, do not contain the cost aggregation step.

Table 3.2: Summary of Stereo Matching Algorithms

Method	Matching cost	Aggregation	Optimization	Refinement
SAD, SSD	Squared difference	Square window	WTA	Any approaches
Non-Parametric, Zabih [103]	Rank transform	Square window	WTA	None
Pixel-to-pixel Stereo, Birchfield [123]	Shifted absolute difference	None	DP	None
Locally adaptive, Kanade [87]	Squared difference	Adaptive window	WTA	None
Cooperative algorithm, Zitnick Kanade [106]	Squared difference	Iterative aggregation	WTA	None
Maximum likelihood, Cox [124]	Squared difference	None	DP	None
Graph cut, Boykov [79]	Squared difference	None	Graph cut	None
SMP, Stefano [85]	Squared difference	Square window	LC	Subpixel interpolation
Fast correlation-based, Yoon [99]	Squared difference	Square window	WTA	LRCC
Multiple windowing, Fusiello [125]	Squared difference	Multiple windowing	WTA	LRCC
Segment-based GC, Hong [119]	Absolute difference	Colour segmentation	Graph cut	Segment plane
Anisotropic diffusion, Banno [126]	Mutual information	None	Belief propagation	Anisotropic diffusion
Linear stereo matching, De-Maezto [127]	Absolute difference or Mutual information	Adaptive weight	WTA	IC-LC
Layered, Bleyer [118]	Absolute difference	Colour segmentation	Graph cuts and Plane	None
Segment-based BP, Klaus [81]	Squared difference and gradient	Square window & segmentation	WTA & Belief propagation	LRCC
Sliding window, Muhlmann [104]	Squared difference	Sliding window	WTA	Subpixel and filtering
Fast stereo matching, Humenberger [100]	Census transform	Square window	WTA	LRCC
MEVSV, Khaleghi [128]	Census transform	Square window	WTA	LRCC and intensity
Fast Bilateral Stereo (FBS), Mattoccia [117]	Absolute difference	Adaptive weight bilateral filtering	WTA	None
Different Array (DA), Zhang [129]	Absolute difference	Colour segmentation	Gaussian	None
AD-Census, Mei [130]	Absolute difference & Census	Cross-based aggregation	Scanline optimization	Multi-step and sub-pixel interpolation

The squared and absolute differences are common methods of selecting a matching cost step, where the latter is usually used in the global-based approach. The cost aggregation in Table 3.2 comprises various techniques such as square window, adaptive window, multiple window, adaptive weight and colour segmentation. In the optimization and refinement disparity stage of Table 3.2, some of the proposed algorithms use Winner-Take-All (WTA) to cater the window-based approaches in a cost aggregation step. Other methods include Dynamic Programming (DP), Graph Cuts, Locally Consistent (LC), Belief Propagation, filtering and Left-Right Consistency Check (LRCC). From this table, it is obvious that there is a quite large subset of possible algorithm design spaces that have been explored over the years.

In the state-of-the-art research of 3D vision, most image processing and stereo vision based approaches use image pairs captured by left-to-right within the horizontal of the epipolar line. The image pairs will be rectified before being processed with stereo matching algorithms to obtain the disparity depth map and matching correspondence pixels. As described in the Table 3.2, almost all the represented stereo matching in this table are left-to-right matching. Computational approaches to stereo matching have often taken advantage of geometric constraints which state that matching elements in the left and right eyes are on the epipolar lines. However, experiments with dynamic random element stereogram carried out by Stevenson [131] revealed that human stereopsis can detect and identify the depth of matches over the range of both vertical and horizontal disparities.

Most of the stereo matching algorithms presented in the Middlebury Stereo Ranking [22] mainly focus on finding the matching between stereo pair based on the left-right scene. Not many submissions have been done to find the corresponding pixels points for the scene captured based on vertical disparity although the concept is similar for the top-down (upper-lower) camera configuration. Normally, this type of matching is implemented for a specific type of application, such as vertical stereo system for range-finding for vehicles system by Miyazaki [132]. The stereo matching for this system is calculated based on characteristic features of horizontal lines in the upper and lower images. A mobile robot for localization and mapping developed by Caron [133] uses an omnidirectional stereo vision sensor. It uses four parabolic mirrors and orthographic camera to produce four images of the same scene. Neither of the systems adapt any stereo matching algorithms proposed in [22], nor create any virtual view synthesis

images based on the disparity depth map. Although Yang [134] presented a stereo matching algorithm utilizing vertical disparity based on neural networks, analysis and performance are mainly for left-right image pairs.

The system developed by Yang [135] for video conferencing employs the matching for vertical disparity based on upper and lower images. The system synthesizes virtual views for eye gaze correction with correlation-based stereo feature and template matching of a human face. This system was designed specifically for video conferencing by utilizing two cameras mounted vertically and limited by one virtual view synthesis only. It also works only for faces and not for dense stereo camera configurations. Therefore in Chapter 6, this thesis proposes an image view synthesis framework based on multi-view camera arrays configuration by matching and synthesizing inter-view images algorithms horizontally and vertically. The Section 3.3 will discuss the 3D scene representation that will lead to the novel view synthesis implementation.

3.3 3D Scene Representation

In computer graphics literature, the 3D scene representation can be classified as a continuum between two extremes. The one extreme is represented by classical 3D computer graphics, which is also known as the geometry-based modelling. The opposite of this extreme is called image-based modelling and does not use any 3D geometry at all. In between the two extremes, there are a number of methods that make more or less use of both approaches and combined their advantages in some way. The choice of a certain 3D scene representation format is of central importance for the design of any 3D video and free-viewpoint system [2]. On the one side, it sets the requirements for acquisition and multi-view signal processing. For instance using an image-based representation implies using a dense camera setting. A relatively sparse camera setting would only give poor rendering results of virtual views. In contrast, using a geometry-based representation implies the need for sophisticated and error prone image processing algorithms such as object segmentation and 3D geometry reconstruction. The other side of the 3D scene representation determines the rendering algorithms interactivity, as well as compression and transmission if necessary. The 3D scene representations of this two approaches are illustrated in Figure 3.3 [136].

In a geometry-based approach, multi-view videos are acquired from randomly distributed network of cameras and then a 3D model is extracted based on the captured images. The images will be rendered with new views using advanced rendering techniques. This approach is also called model-based technique. In most cases, scene geometry is described on the basis of 3D meshes. Real world objects are reproduced using geometric 3D surfaces with an associated texture mapped onto them. More sophisticated attributes can be assigned as well. For instance, appearance properties such as opacity and reflectance can enhance the realism of the models significantly.

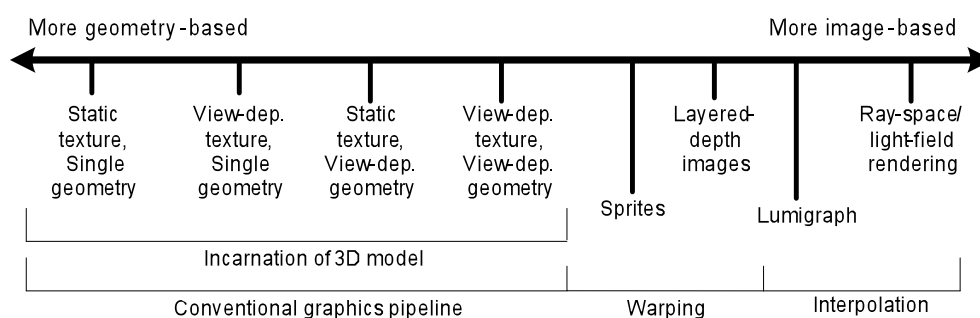


Figure 3.3: Scene representation categories [136]

Geometry-based modelling is used in applications such as games, graphic design and movies. The technology for both production and rendering has been optimized over the last few years, especially in the case of common 3D mesh representations. Typically, the scenes of purely computer generated with these models performed very well in comparison to the image-based approach. In addition, the state-of-the-art PC graphic cards are able to render highly complex scenes with an impressive quality in terms of refresh rate, levels of detail, spatial resolution, reproduction of motion and accuracy of textures [2].

Despite the advances in 3D reconstruction algorithms, reliable computation of 3D scene models remains difficult with most of existing systems are restricted to foreground objects only [4, 137, 138]. Furthermore, volumetric representations such as voxels (from volume elements) can be used instead of a complete 3D mesh model to describe 3D geometry. Prior knowledge of the object model can be used to improve the reconstruction quality such as in voxel based representations, where the human body is the object of interest in the scene [139]. It can easily integrate information from multiple cameras but it is limited in resolution. The work of Vedula [139] is based on the explicit recovery of the 3D scene properties that uses the voxel colouring algorithm to recover a 3D voxel model of the scene at each time instant, and 3D scene flow algorithms to

obtain the 3D non-rigid motion of the scene between consecutive time instants. The voxel models and scene flow become inputs to a spatio-temporal view interpolation algorithm. Some research mainly focused on the human object because of the scenes involving humans are amongst difficult tasks to reconstruct into the 3D model. A triangle mesh representation is employed in the work of Carranza [137] and Theobalt [4] because it offers a closed and detailed surface representation. It composed of multiple rigid body parts that are linked by a kinematic chain through geometry model of a human body.

The model-based approach has the advantage of reducing the acquisition cost by using fewer cameras. However, a drawback of this approach is the typical high cost and human assistance required for 3D content creation. The algorithmic complexity increases in order to capture the real-world scene models in real-time processing [48]. The 3D scene and object modelling are often complex and time consuming, and it will become even more complex when dynamically changing scenes are considered. Furthermore, an automatic 3D object and scene reconstruction implies an estimation of camera geometry, depth structures and 3D shapes. With some possibility, all these estimation processes generate errors in the geometric model. These errors then have an impact on the rendered images. Therefore, high-quality production of geometry models is typically user assisted, for example in film productions.

Another known technique is the image-based approach. This uses a densely distributed network of cameras to acquire high-resolution light fields and then uses image-based rendering algorithms to generate images at the new view-points. The image-based approach has the advantage of reconstructing new views without the need of a 3D scene model. It has the potential to produce high quality of virtual view synthesis images without any 3D scene reconstruction through dense sampling of the real world with a sufficiently large number of natural camera view images [2]. In general, the synthesis quality increases with the number of available views. Hence, typically a large number of cameras have to be set up to achieve high performance rendering with a huge amount of image data needs to be processed. If the number of used cameras is too low, interpolation and occlusion artefacts will appear in the synthesized images, which affecting the image quality. The image-based method also demands more storage and transmission bandwidth for video data [140]. One of the challenges for the image-based approach is to capture and transmit dense light field in a cost-effective way.

Some examples of image-based representations are ray space or Light Field [6, 141], Lumigraph [142, 143] and panoramic configurations, including concentric and cylindrical mosaics [144]. The basic idea in these methods is capturing the complete flow of light in a region of the environment. Such a flow is described by a plenoptic function. The plenoptic function was introduced by Adelson and Bergen [38] in order to describe the visual information available from any point space and has been introduced in the Section 2.2. It is characterized by seven dimensions, namely the viewing position (V_x, V_y, V_z) , the viewing direction (θ, ϕ) or (x, y) in Cartesian coordinates, the time and the wavelength for dynamic scene. It can be summarized as $P=P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t)$. The image-based representation stage is a sampling stage, where the samples are taken from the plenoptic function for representation and storage [145].

Research on image-based modelling is mostly on how to make reasonable assumptions to reduce the sample data size while keeping the rendering quality [78]. One of the main strategies to reduce the data size is to restrain the viewing space of the viewers. By ignoring the wavelength and time dimensions, McMillan and Bishop [146] introduced plenoptic modelling in 5D function, which is $P=P_5(V_x, V_y, V_z, \theta, \phi)$. A static scene recorded by positioning cameras in the 3D viewing space, each on tripod capable of continuous panning. At each position, a cylindrical projected image was composed from the captured images during the panning. This forms a 5D image-based representation: 3D for the camera position and 2D for the cylindrical image. In order to render a novel view from the 5D representation, the nearby cylindrical projected images are warped to the viewing position based on their epipolar relationship and visibility tests [145].

The Light Field and Lumigraph image-based representations ignored the wavelength and time dimensions with the assumption the radiance does not change along a line in free space. Both approaches parameterized the space of oriented lines with the light rays recorded by their intersections with two planes. One of the planes is indexed with coordinate (u, v) and the other with (s, t) . Figure 3.4(a) [78] shows an example where the two planes, camera plane and focal plane are parallel. An example of light ray is shown and indexed as (u_0, v_0, s_0, t_0) . The two planes are then discretized so that a finite number of light rays are recorded. If all the discretized points from the focal plane are connected to one discretized point on the camera plane, an image for 2D array of light rays is obtained. Therefore, the 4D representation is also a 2D image array as shown in Figure 3.4(b) [78].

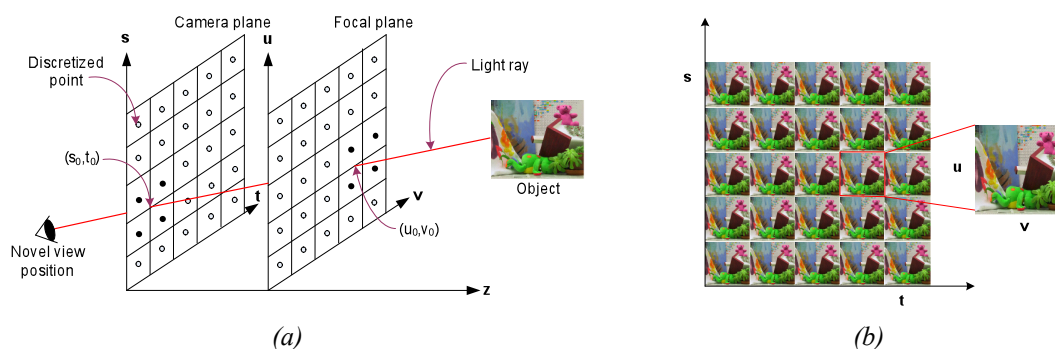


Figure 3.4: Light Field image based representation [78]: (a) One parameterization of the Light Field. (b) A sample Light Field image array

The difference between Light Field and Lumigraph is that the Light Field assumes no knowledge about the scene geometry. As a result, the number of sample images required in the light field for capturing a normal scene is huge. On the other hand, Lumigraph reconstructs a rough geometry for the scene to facilitate the rendering with a small amount of images. For this reason, Lumigraph is also classified as a hybrid and not a pure image-based modelling technique.

Between these two extremes lies a range of methods with varying proportions of geometric and image input information that has been explored, for example in [147]. Using a complete 3D model allows freedom in the final rendering but requires more computation and often creates noticeably artificial output images. In contrast, the Image-Based Rendering (IBR) approach requires little geometric information and can give potentially photo-realistic results but requires many more input images. A layer-based representation of the scene geometry [39, 148] represents a compromise that has low geometric complexity, while allowing view synthesis from a moderate number of input images [149].

Other representations do not use explicit 3D models but they use depth or disparity maps. Such maps assign a depth value to each sample of an image. The original 2D image can be combined with the depth map to build a 3D-like representation, also known as 2.5D [69]. This can be extended to layered depth images, where multiple colour and depth values are stored in consecutively ordered depth layers. A different extension is to use multi-view video plus depth, where multiple depth maps are assigned to the multiple colour images [8, 150], whereas the Advanced Three-Dimensional Television System Technology (ATTEST) [69] project proposal is based on the distribution of video-plus-depth data corresponding to a single central viewing position. Some of the algorithms will be discussed again in Section 3.4 and 3.5 since they are

related to the view synthesis and image based rendering algorithms.

3.4 View Synthesis Algorithms

Stereo images provide simple means of perceiving the relative depth information in a real world scene. However, 3D television, which probably uses stereoscopic videos, leads to increase visual strain because of imbalance between accommodation and convergence of the eyes. The view synthesis technique can be used to overcome this problem [151]. The look-around capability of the view synthesis makes viewers comfortable and produces photo-realistic images [152].

Novel view synthesis covers a broad set of computer vision techniques which have been developed to solve the problem of generating a novel view from a set of measurements of a scene (typically a set of images). It is concerned with two things: determining correspondences between images and interpolating or extrapolating from these correspondences to form a new image. A typical solution to the problem of free viewpoint video is to capture a set of video sequences instead of single images and then apply novel view synthesis algorithms on a frame-by-frame basis. Another approach is to capture the same scene with a number of synchronized cameras (such as stereo).

The basic underlying principle in novel view synthesis algorithms are based on the plenoptic function, introduced in the commonly seen 5-dimensional form by McMillan and Bishop [146], also as 7-dimensional by Adelson and Bergen [38]. This states that a single function in terms of a position in (V_x, V_y, V_z) and a direction with azimuth and elevation (θ, ϕ) can describe all possible images of a scene. An image is then a discrete sample of the plenoptic function, with each pixel being an integration over a small range of θ and ϕ . The problem of novel view synthesis can then be expressed as an attempt to generate a continuous representation of an entire plenoptic function given a small set of discrete samples.

A second principle that is often relied upon is that images are formed by objects, specifically that a region of an image is formed by light reflecting off a patch s on a surface S as shown in Figure 3.5 [49]. By determining the properties of s , and with a knowledge of the camera calibration and geometry, the pixels relating to s can be determined across a set of images M . S is often referred to as the ‘scene geometry’ [49].

As illustrated in Figure 3.5, p is a real pixel on real image I and p' is a synthesized pixel on synthetic image I' . M is the set of real images.

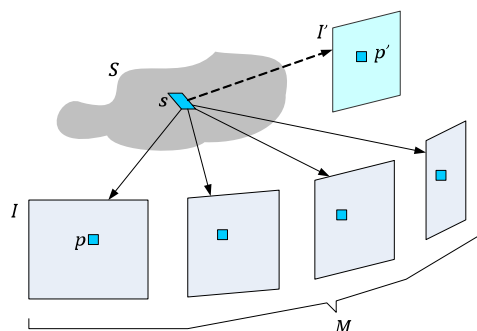


Figure 3.5: A set of related pixels measured by a patch s on surface S [49]

There are generally two main categories of view synthesis algorithms: reconstruction-based methods and interpolation-based methods [153]. Reconstruction-based methods use explicit or implicit 3D structures of the scene to synthesize new views through a fundamental matrix or a trilinear tensor. However, interpolation-based methods do not require 3D structures or camera parameters. The interpolation-based methods are able to generate smooth transitions between reference images by simple interpolation from two stereoscopic images. In considering of compression for multi-view image sequences, whichever approach we choose, we must consider coding efficiency, computational complexity and ease of generating the intermediate-view images. In most cases, the view interpolation can be divided into two processes: disparity estimation and intermediate-view generation.

The view interpolation of the stereo images may be thought of the extended version of the motion estimation/compensation problem since the disparity of the stereo images can be considered as the motion vector of the monocular moving pictures. Therefore, the motion estimation algorithm of two successive frames in the monocular video could also be applicable to finding disparity values of two stereo images [154]. Those approaches of various motion estimation methods can be directly applied to stereo image coding since their main focus is to minimize coded bits and prediction errors. However, the approaches of minimizing prediction errors have limitations in generating intermediate view images, and the interpolated images usually have visible artefacts since those are mainly focused on coding. Therefore, finding exact disparities of stereo images is an important work in view interpolation. Stereo image pairs are obtained using simultaneously recording a scene with two cameras at different positions. The

relative position, orientation, and some additional parameters of the cameras generate the disparity vectors of the stereo image pairs [154]. When two pinhole cameras are placed with equal orientation while their positions differ only in the direction of the scan line, it is called parallel setup of stereo cameras. This case makes the disparity vector a one-dimensional value. Therefore, finding disparity vector in stereo image pair is different from finding motion vector in monocular video sequences according to the camera setup.

Fan and Ngan [155] proposed a coding method of disparity map based on adaptive triangular surface modelling. This algorithm consists of two stages: to find a smooth disparity map using block-based hierarchical disparity estimation and to model the acquired disparity map by Delaunay triangulation on a set of nodes [152]. It compresses the set of nodes with disparities by the differential pulse coded modulation and variable length coding. Because the disparity map is modelled by a finite number of nodes, the acquired disparity map must be smooth so that the disparity error can be neglected. If the disparity map is not smooth enough, i.e., there exist some disparity discontinuities; large number of triangulation nodes must be placed very heavily around the discontinuity or noisy area to reduce the disparity error.

Sethuraman [156] proposed a compression method of multi-view image sequences using a generalized quad-tree. By partitioning a reference image successively by the generalized quad-tree decomposition and finding a disparity value for the partitioned rectangular patch, he achieved very high compression efficiency. However, the drawback of this algorithm stems from having only one disparity value on every rectangular patch. If input stereo images have a continuously varying disparity map, the strategy of just one disparity value per rectangular patch makes the disparity error large.

Wang and Wang proposed mesh-based analysis and coding of multi-view video sequence [157]. In their work, disparity estimation and compensation were performed in such a way that the compensation error of full frame should be minimized. Node points were iteratively moved in the direction of minimizing the prediction error. They proposed a full search method and computationally efficient fast search method. However, the computational complexity is high because of the iterative procedure. In addition, they did not consider the occlusion problem.

Hong Park [152] proposed a new view interpolation method for stereoscopic images that bring together computational efficiency and high PSNR of the intermediate-view image compared to the previous view interpolation methods. In spite of this, the proposed algorithm contains high complexity in the implementation point of view. Further description on the image view synthesis described in Section 3.5 on image based rendering.

3.5 Image Based Rendering

Virtual view synthesis refers to the generation of a view of a scene from an arbitrary or novel viewpoint. Image Based Rendering (IBR) techniques generate a novel view from a set of available images or key views. Unlike traditional 3D computer graphics, in which 3D geometry of the scene is known, IBR techniques render novel views directly from input images [30]. Figure 3.6 illustrates this idea [158]. Camera number 1 and camera number 2 are real cameras that capture the same scene from different viewpoints. Also shown, is a virtual camera placed at a viewpoint, which is between the two real cameras. The goal is to render a novel view observed by this virtual camera.

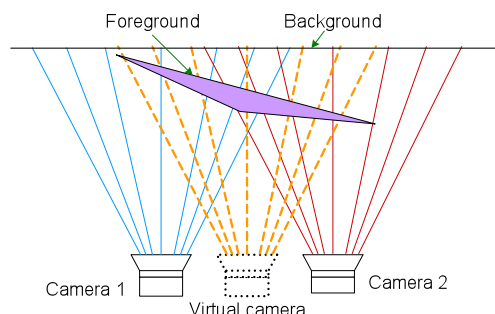


Figure 3.6: Virtual view synthesis [158]

In general, new views of scene are reconstructed in the rendering stage. There is a spectrum of rendering techniques depending on the functionality and technologies required in the system, due to the amount of geometry information of the scenes/objects being used. A survey of IBR techniques is presented by Shum [30]. This survey classifies IBR techniques into three categories according to how much geometric information is used based on the scene representation illustrated in Figure 3.7 [78]:

- Rendering with explicit geometry (either with approximate or accurate geometry).

- Rendering with implicit geometry (with correspondence).
- Rendering without geometry.

Based on Figure 3.7, at one end of the spectrum, there are very accurate geometric models of the scenes and objects, for instance generated by animation techniques with only a few images required to generate the textures. Novel views can be rendered using conventional graphic techniques with given 3D models. Interactive rendering with movable objects and light sources can be supported using advanced graphics hardware.

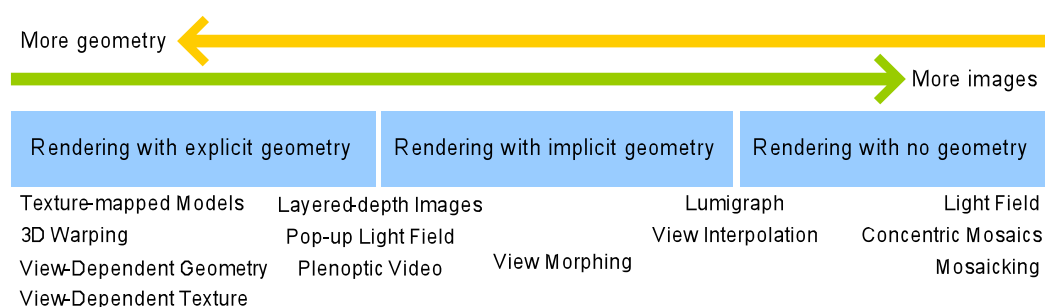


Figure 3.7: Spectrum of rendering representations [78]

At the other extreme, Light Field [7] or Lumigraph [8] rendering relies on dense sampling by capturing more images with very small geometry information for rendering. The advantage of this approach is its superior image quality compared with 3D model for complicated real-world scenes. It also requires less computational resources for rendering regardless of the scene complexity because of most of the quantities involved are pre-computed. The description of the IBR classification is discussed in Section 3.5.1 to 3.5.3.

3.5.1 Rendering With Explicit Geometry

This category is represented by techniques such as 3D warping, Layered Depth Image (LDI) rendering and view-dependent texture mapping. The 3D warping [62, 63] assumes that the depth information is available for every point in one or more images. LDI rendering [39] is an improvement over 3D warping since it treats the disocclusion artefacts in 3D warping. LDI rendering, however, assumes the knowledge of what is behind the visible surface. The texture maps generated by applying computer vision techniques to capture images. View-dependent texture mapping [57] blends the textures from different viewpoints after warping them all to a common surface.

3.5.2 Rendering With Implicit Geometry

In this category, the techniques rely on positional correspondences across a small number of images to render new views [30]. Typically, the positional correspondences are generated from the input images by the user. This class of rendering has the term *implicit* to express the fact that geometry is not directly available [52]. However, the 3D information is computed using the usual projection calculations. Methods in this category rely on positional correspondences across a small number of images to render new views.

A classical approach for generating a synthetic view is image view interpolation [151, 152]. View interpolation uses general dense optic flow to generate the intermediate views directly. Hence, the intermediate view may not necessarily be geometrically correct. The drawback of this method is that it can only produce images that are intermediate views between two original images, where the virtual camera lies on the baseline between the two real cameras. Another representative of this category is view morphing [52], a special version of view interpolation. In this method, the interpolated views are always geometrically correct.

3.5.3 Rendering Without Geometry

Light Field rendering [6, 159] and Lumigraph systems [142] are the main techniques in this category. These techniques do not rely on any geometric information, but they rely on oversampling to counter undesirable aliasing effects in the output display. The purpose of this research is to come up with a rendering technique that requires depth information based on the correspondence input and works well when the disparity between two adjacent views is not too high. These two key images depict the same objects from slightly different view-points. The computational complexity is bounded by the image resolution (spatial size of the image), rather than the scene complexity. The viewpoint for the novel view can be anywhere on the line joining the two camera centres.

The IBR technique can then be used to generate a video of viewpoint traversal in a static natural scene. The effect of inserting novel views on the viewing experience will then be observed. One of the examples on viewpoint traversal is from the film ‘The Matrix’, where the actor, ducks to avoid bullets. At this instant, the video freezes in

time, so that the dynamic scene becomes static, and then the viewpoint is changed smoothly to a completely different angle. Once the viewpoint is changed, the scene becomes dynamic in time again.

As described earlier, there are several methods for generating an arbitrary new view of a scene from a set of existing views. One approach is to create a textured 3D model of the entire scene and to use this for synthesizing new views through model-based rendering. Meanwhile, in Image Based Rendering (IBR), the new images are generated by combining the individual pixels with a densely sample set of input images.

IBR is a technique for generating arbitrary views or synthesizing free-viewpoint images of a scene that differs from the conventional computer graphic approach. It is an interesting alternative for generating novel views compared to model based rendering due to its lower complexity and can produce photo-realistic images. Instead of rendering views of 3D scenes by projecting objects and their textures, new views are rendered by interpolating available nearby images. The scene is not represented by its objects but it is represented by the light rays that are captured by the cameras, for example in the Light Field [6] and Lumigraph [142]. New views are simply generated by interpolating the sampled light rays. The advantage of such a method is that little or no geometry of the scene is required, as opposed to a full geometric model, which can be very difficult to obtain for natural images [42]. In addition, the rendering algorithms produce convincing photorealistic results since the interpolated viewpoints are generated by combining real images. Due to this advantage, IBR is expected to become a fundamental technology in many applications, such as 3D content production, telecommunications and virtual reality systems [160]. The main drawback of such a representation is the fact that huge amounts of data (typically hundreds of thousands of images) need to be captured, stored and transmitted.

IBR can be defined as a sampling and interpolation problem. Thus, it is interesting to study this problem in a traditional sampling and interpolation framework. That is, to estimate the spectrum of the signal at hand and determine the sampling frequency necessary for an aliasing free reconstruction. All the visual information can be characterized with a single seven dimensional function called the plenoptic function. In the research work by Chai et. al. [147], the authors showed that the spectrum of the plenoptic function is approximately band limited by the maximum and minimum depths of the scene and has a distinctive bow-tie shape. From this spectrum, the authors are

able to deduce the number of samples (images) is required for an aliasing-free rendering. They also showed that the interpolation filter can be steered with an angle that depends on the depth of the scene in order to reduce aliasing. Therefore, there is a clear trade-off between the number of images, the number of layers and the depth variation.

3.6 Layered Image Based Rendering

Layers have been used for many applications in multi-view images. Several layer based representation techniques have been proposed such as the layer depth images [39]. They have been successfully used in free-viewpoint video [8] as well. However, these methods are designed to produce an accurate depth map of the scene. New views of the scene are rendered through warping of the layers. These techniques are very sensitive to errors in the depth reconstruction [42].

Several other layer based representation techniques are designed for image based rendering such as the coherent layers in Pop-up Light Field [148] and plenoptic layers [3]. These representation techniques are based on approximate geometry rather than exact depth. In [147], Chai has shown that a certain number of layers are optimal for a given scene and number of cameras. Therefore, extracting more layers is unnecessary. Some scenes do not require advanced layer extraction methods. In fact, the layer extraction should be tailored to the scene and samples of the light field in an adaptive manner. That is, there is a relation between the complexity of the scene (depth variation, occlusion and non-Lambertian) and the layer extraction.

A simple scene with small depth variation only requires very few depth layers, which can be extracted very quickly, for example, two different depths. A scene with large depth variations requires many different rendering depths and therefore, the layer extraction must be tested for more depths. Following this analysis, Li [161] and Takahashi [160] reconstructed an approximate depth map with different constant depth filters and fusing in-focus regions. In [148], the user manually extracts layers until becomes satisfy with the rendered result. Table 3.3 summarizes the selected layer based image techniques for the image based rendering. Most of the featured techniques in Table 3.3 deal with the layers using a different approach. In order to render an image, some of the techniques require a known geometry [42, 147], user interaction [148, 160]

and multiple image views [149]. In addition, some of the algorithms are restricted on a limited number of layers [162, 163], high computational requirements [8, 118] and do not take occlusion into consideration in rendering process [160, 161].

One of the main advantages of layer based representation is that the method enables to extract a 3D object or layers from one real scene and to superimpose it onto another 3D, either real or computer graphic scene. Since the composition of the layer is obtained in a view-dependent way, both the object and new background move naturally along with the viewpoint changes as if they existed together in the same space. The experimental results on the object and layer superimposition onto different images, which have presented by Ishii [163] show the effectiveness of this scheme.

The layer based representation techniques are mostly based on planar disparity, foreground/background layers and colour segmentation. Bleyer in [118] proposed a technique that divides one single surface that contains texture into several segments using a colour segmentation algorithm. In general, the concept of layers and planar presented by the algorithms in Table 3.3 are similar, but their main differences are on the way the layers are assigned and extracted from their disparity map. The basic idea behind this is that if the disparity map is correct, then the synthesis image is very similar to the real image from that viewpoint. Therefore, the quality of the reconstructed image through layers depends on the stereo matching algorithms. Sjostrom [164] outlined that possible errors occurring in 3D image synthesis for a layered Depth Image Based Rendering (DIBR) algorithm such as empty cracks, translucent cracks, corona-like effects, unnatural contours and empty regions. The quality analysis of inter-view images including the layer based image technique will be described in Section 3.7.

Table 3.3: Summary of Layered Image Techniques

Publication	Techniques	Features	Comments
Shade et. al., 1998 [39]	Layered Depth Images (LDI)	IBR with smooth varying surface with depth. Contains complex geometries for LDI.	Combine traditional graphics elements and planar sprites for 3D computer graphics modelling. Use only single input camera and not stereo images.
Chai et. al., 2000 [147]	Plenoptic Sampling	Based on sampling rate light field rendering	Assume known geometry.
Shum et. al., 2004 [148]	Pop-up Light Field rendering	User interaction that specifies how many coherent layers needed	Representations are based on approximate geometry rather than exact depth. Require user interactions.
Zitnick et. al., 2004 [8]	High quality video view interpolation	Video based rendering of dynamic scenes using multiple video streams combined with Image Based Modelling Rendering (IBMR) algorithms.	Applying colour segmentation to generate high quality photo consistent correspondences across all cameras. High complexity technique and computational demanding.
Li et. al., 2003 [161]	1D Light Field	Rendering driven depth recovery.	Block-based multi-layer depth representation. Does not take occlusions and yields artefacts in the boundaries of layers.
Bleyer and Gelautz, 2005 [118]	Layer stereo matching with image segmentation and global visibility	Global stereo matching with collection of planar layers based on colour segmentation and layer assignment.	Image segmentation with window-based approach that exploits the result of segmentation. Addresses the problems of non-textured regions and occlusions. High computational requirements.
Takahashi and Naemura, 2006 [160]	Layered Light Field rendering	Focus on measurement scheme.	Does not take into account occlusions and relies on the user for the number of layers.
Smolic et. al., 2008 [162]	Intermediate view interpolation based on multi-view video plus depth	Two boundary layers and one reliable layer are used (separate foreground and background boundary layers).	Does not rely on 3D graphics support but uses image-based 3D warping. Suitable for advanced 3D video system but restricted into two layers only.
Berent et. al., 2009 [42]	Adaptive layer extraction for IBR	Automatically adapts the number of depth layers to extract the scene itself and the spacing between the sample views.	Extracting depth layers in the presence of occlusions for IBR based on the spectral analysis of the plenoptic function. Require geometrical information.
Pearson et. al., 2011 [149]	Accurate non-iterative depth layer extraction	Fast-unsupervised method for synthesizing viewpoints of a scene with hierarchical approach to assign depths.	Includes optimising placement of the depth layers. Building geometric model maximise its accuracy. Requires multiple image views.
Ishii et. al., 2010 [163]	Joint rendering and segmentation	Robust methods that share a calculation process between the synthesis and segmentation.	Exploits the segmentation and graph cut to extract layers. The algorithm assumes no other object around the depth of the target object.

3.7 Performance Evaluation

In this research, both subjective and objective quality measurements are used to evaluate the quality of the synthesized images, which are based on the Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) Index.

3.7.1 Peak Signal-to-Noise Ratio (PSNR)

Subjective approaches to evaluate the quality of the inter-view synthesized images are too general (either good or bad), which do not define the image diagnosis reliability with the original image. Mean Squared Error (MSE) and PSNR are widely used for the objective quality measurement of the images. However, MSE and PSNR do not correspond well with the subjective visual quality. The failure of the MSE is partially due to the following: spatial relationships between the samples of the signal, and the relationships between the original and the distorted image are ignored [165]. So, which quality measurement should be used that best corresponds to the visual and diagnostic quality?

The MSE is the cumulative squared error between the synthesis image $g(i, j)$, and the original image $f(i, j)$, which can be described in the following equation:

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f(i, j) - g(i, j))^2 \quad (3.13)$$

where M and N are the size of the image.

The Peak Signal-to-Noise Ratio (PSNR) is calculated and derived from the MSE using the following equation:

$$PSNR(f, g) = 10 \log_{10} \left(\frac{R^2}{MSE(f, g)} \right) \quad (3.14)$$

where R is the dynamic range of the pixels.

As described earlier, the problem of MSE is that it is independent of temporal or spatial relationships between pixels and it ignores the correlation error between the reconstructed and the original signals. All signal samples (structured and smooth areas) are treated equally.

3.7.2 Structural SIMilarity (SSIM) Index

A newer image fidelity measure called the Structural SIMilarity (SSIM) index, introduced by Wang and Bovik [166], assumes that images are highly structured and there exist strong neighbouring dependencies among the pixels. The human visual system is highly sensitive to structural information or distortions in an image and SSIM is automatically adjusted to mark the non-structural ones. The SSIM index measures the differences and similarities between two images by combining three components of the human visual system, which are luminance, contrast and structure. The local SSIM index is computed within a sliding window of $m \times n$ neighbourhood pixels. The resulting quality map SSIM reveals local image quality. The total SSIM score is computed by averaging the local SSIM values.

The SSIM index proposed by Wang [165, 166] measures the distance between two images f and g , by combining three components of the human visual system (HVS), which are luminance, contrast and structure.

The luminance $l(f, g)$, is estimated using the following equation:

$$\mu_f = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m f(i, j) \quad (3.15)$$

The contrast $c(f, g)$ can be measured by variance, as follows:

$$\sigma_f^2 = \frac{1}{(mn-1)} \sum_{i=1}^n \sum_{j=1}^m (f(i, j) - \mu_f)^2 \quad (3.16)$$

The final component, structure $s(f, g)$, is measured by covariance using the following equation:

$$\sigma_{fg} = \frac{1}{(mn-1)} \sum_{i=1}^n \sum_{j=1}^m (f(i, j) - \mu_f)(g(i, j) - \mu_g) \quad (3.17)$$

The three components are finally combined to calculate the SSIM Index between the two image f and g , using the following equation:

$$SSIM(f, g) = \left(\frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \right) \left(\frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \right) \left(\frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \right) \quad (3.18)$$

over $m \times n$ neighbourhood pixels. The notations μ_f and μ_g are (local) sample means of f

and g respectively, σ_f and σ_g are (local) sample standard deviations of f and g correspondingly, and σ_{fg} is the (local) sample correlation coefficient between f and g . The items C_1 , C_2 , and C_3 are small positive constants that stabilize each term, so that near zero sample means, variances, or correlations do not lead to a numerical instability.

In [166], the SSIM index is computed using an 11x11 sliding windows of circular-symmetric Gaussian weighting function. The constants chosen for the SSIM evaluation are defined as $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$ and $C_3 = C_2/2$, where L is the dynamic range of the pixel values (255 for 8-bit greyscale images) with $K_1 = 0.01$ and $K_2 = 0.03$. The SSIM evaluations in this research computed based on these parameters.

Figure 3.8 shows samples of evaluation using both PSNR and SSIM indices. The original image of ‘Najla’ has been contaminated with ‘salt and pepper’ noise with 0.001 noise density as shown in the Figure 3.8(b). The noise is not noticeable in the image, but the SSIM map image reveals the errors contained in the image Figure 3.8(c) with random dot regions. The PSNR of this image is 56.74 dB, with the SSIM index at 0.97. The higher SSIM index indicates the better quality of the image. The noise density of the ‘salt and pepper’ was increased to 0.02 and the new image can be seen in Figure 3.8(d). The errors between this image and the original image are relatively huge compared to the previous example, where the PSNR is 44.01 dB, and the SSIM index is 0.68. The dark regions in the SSIM image map show the errors created by the added-noise (Figure 3.8(e)).

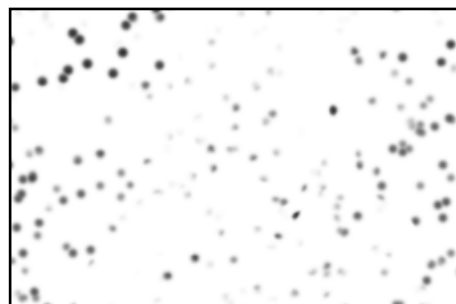
Lastly, the image is sharpened by contrast enhancement Laplacian filter as shown in Figure 3.8(f). Subjectively, the sharpened image quality is better in the human visual system compared to the previous noise-added images. However, the PSNR indicates differently with the computed PSNR of 32.96 dB, which is lower compared to the previous example. The SSIM index provides the additional measurement with the SSIM index calculated as 0.92. The SSIM quality map image in Figure 3.8(g) signifies the degradation occurs rather uniformly on the objects regions (the girl). The grey regions show that there are differences in the pixel values of the image in term of contrast and luminance but these differences do not affect the quality of the tested image. Therefore, the SSIM index provides additional quality measurement for the new synthesis images along with the PSNR and MSE.



(a) Original image 'Najla'



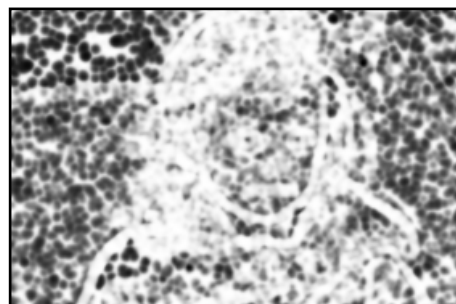
(b) Image with 0.001 noise density;
PSNR = 56.74 dB



(c) SSIM map image with SSIM = 0.97



(d) Image with 0.02 noise density;
PSNR = 44.01 dB



(e) SSIM map image with SSIM = 0.68



(f) Image sharpened; PSNR = 32.96 dB



(g) SSIM map image with SSIM = 0.92

Figure 3.8: Sample of results comparison on PSNR and SSIM between the original image, images with noise and sharpened. The respective SSIM maps obtained and evaluated based on the standard parameters proposed by Wang [166].

3.8 Conclusion

This chapter provided the core of the stereo matching and novel view synthesis algorithms. The fundamentals of stereo matching algorithms were described for sparse and dense disparities maps, which may contain matching cost computation, cost aggregation, disparity computation or optimization and lastly disparity refinement components. Depending on the implementation and requirement, not all stereo algorithms take all four components steps. The review of selected algorithms and approaches were discussed in this chapter. The compilation of the research works in stereo matching algorithms provided in the Middlebury Stereo Page. For the stereo matching, the disparity depth map can be measured and used as the next stage of the 3D image and video processing, such as the inter-view synthesis algorithm.

This chapter also described the stereoscopic creations that differentiated by the underlying models of the 3D scene representation, where two major groups can be identified: based on the geometry scene representation and image-based systems. Between these two major groups, there is a range of methods with varying proportions of geometry and image information such as the layer depth image-based representations. The view synthesis algorithms are discussed. Virtual view synthesis refers to the generation of a view of a scene from an arbitrary or novel viewpoint. Image Based Rendering (IBR) techniques generate a novel view from a set of available images or key views. IBR techniques render novel views directly from the input images contrarily to traditional 3D computer graphics, in which 3D geometry of the scene is known.

Finally, the chapter also outlined the image-based rendering algorithm to generate new views of the scenes from an arbitrary or novel viewpoint including the layered image-based rendering. A layer-based representation of the scene geometry represents a compromise that has low geometry complexity, while allowing view synthesis from a moderate number of input images. Several layered representation techniques have been proposed and implemented in free-viewpoint video applications. However, most of the methods are designed to produce an accurate depth map of the scene and new views of the scene are rendered through warping of layers, which are very sensitive to errors in the depth reconstruction.

In the next chapter, we will present the development of a novel Depth Image Layers Separation (DILS) method that is based on the stereo matching and image view synthesis algorithms.

Chapter 4

Virtual View Synthesis Based on Depth Image Layers Separation (DILS)

4.1 Introduction

This chapter presents a novel method for virtual view image synthesis referred to as the Depth Image Layers Separation (DILS). This technique is used to synthesize novel inter-view images based on disparity depth map layers representation. The depth layers are identified by using histogram distribution and separated into several clusters of layers. Each layer is extracted with inter-view interpolation to create objects based on location and depth. DILS features a new paradigm that is not just a method to select interesting locations in the image based on the depth, but it is also a new image representation that allows the description of the objects or parts of the image without the need of segmentation and identification. The image view synthesis can reduce the complexity of multi-camera array configuration for 3D imagery, free-viewpoint applications and light fields imaging. It makes use of a disparity depth map layer separation for image based synthesis and rendering through multi-layer and overlapping techniques. With the selected layer of depth, disparity depth map can be refined independently and the layer can be composed onto different 3D scenes. By exploiting the 3D information, it is possible to discriminate some background or foreground objects of the scene. This is useful for intelligent video tracking and image based rendering. The DILS algorithm can be performed from a simple to sophisticated stereo matching techniques to synthesize the inter-view images.

The rendering system presented in this research contains novelty with respect to Chai [147], Li [161], Takahashi [160], Shum [148] and Berent [42] in several ways. First, any known geometry is not assumed as in Chai [147]. Second, user interaction is not required as in Shum [148]. Third, the depth estimation in Li [161] is block-based which may cause reconstruction artefacts in the boundaries of layers and does not take into

account occlusions. Fourth, Takahashi [160] does not take into account occlusions and relies on the user for the number of layers. Finally, the proposed method of image rendering process does not require geometrical information as in Berent [42]. In contrast, in the proposed method, both the depth estimation and interpolation are taking occlusions into account and the number of layers is estimated based on the histogram distribution.

The chapter is organized into five sections. Section 4.2 provides the background and idea of the layers representation for this research. In Section 4.3, the overall framework of the system design and algorithm is presented that consists of two main stages: matching algorithm and intermediate view synthesis. Section 4.3.1 and 4.3.2 describe the two respective stages in detail. Section 4.4 provides test results for the algorithm and lastly concluding remarks are given in Section 4.5.

4.2 Depth Map Layers Representation

4.2.1 Depth Map

The basic concept of view synthesis with stereo matching data is to use pairs of neighbouring original camera views in order to create and render arbitrary virtual views on a specified camera path between them. Instead of transforming with the Image Based Rendering (IBR) geometry technique, this approach will use the basic idea of range field (horopter) from the stereo rig of the camera. In order to calculate the 3D location or the range field of the scene, basic geometry rules are used. The projection of a 3D physical point on the two image planes requires finding the exact location of the object as described in Chapter 2. The simplest geometry of a stereo system formed by two parallel cameras with horizontal displacement is shown in Figure 4.1, which is derived from the pinhole camera model [44]. The disparity can be determined by finding the difference between the x coordinate of two correspondence points.

Assume a perfectly undistorted, aligned and measured stereo rig is obtained as shown in Figure 4.1, where two camera image planes are exactly coplanar with each other, and their optical axes are exactly parallel. The optical axis is the ray from the centre of projection O through the principal point C is also known as the principal ray that is known distance apart, with equal focal length $f_L = f_R$. Also assume that the principal

points C_L and C_R have been calibrated to have the same pixel coordinates in their respective left and right images. The principal point is where the principal ray intersects the imaging plane. This intersection depends on the optical axis of the lens.

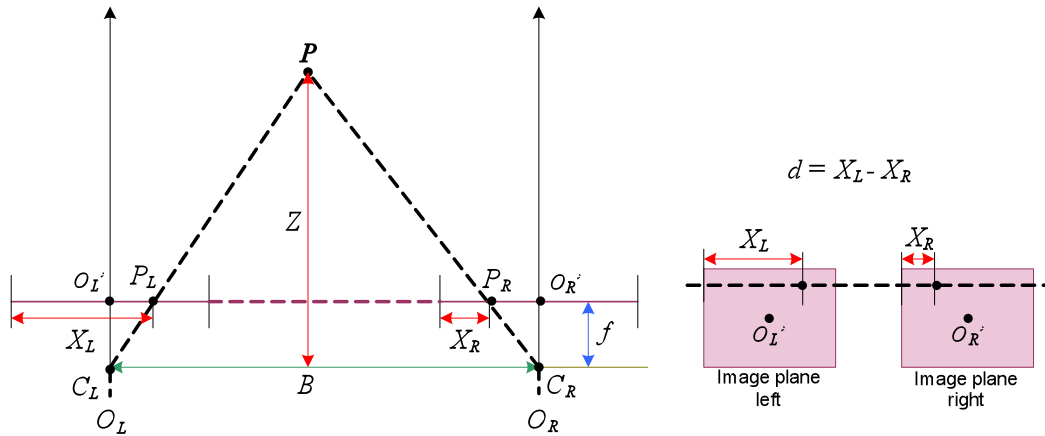


Figure 4.1: Aligned stereo rig and known correspondence [44]

Another assumption is that the images are row-aligned (every pixel row of one camera aligns exactly with the corresponding row in the other camera). This camera arrangement is known as the frontal parallel. A point P in 3D world mapped in the left and the right image views at points P_L and P_R will have the respective horizontal coordinates X_L and X_R .

In this simplified case, X_L and X_R are the horizontal positions of the points in the left and right images, respectively. This allows us to show that the depth is inversely proportional to the disparity between these two views, where the disparity is simply defined by $d = X_L - X_R$. This situation is shown in Figure 4.1, where using the similar triangles ($PO_L O_R$ and $PP_L P_R$) the depth, Z , can be derived as follows[44]:

$$\frac{B - (X_L - X_R)}{Z - f} = \frac{B}{Z} \Rightarrow Z = \frac{fB}{X_L - X_R} = \frac{fB}{d} \quad (4.1)$$

Since depth is inversely proportional to disparity, there is obviously a nonlinear relationship between these two terms. When disparity is near to 0, small disparity differences represent depth differences. When disparity is large, small disparity differences do not change the depth that much. The consequence is that stereo vision systems have high depth resolution only for objects relatively near the camera, as clearly shown in Figure 4.2.

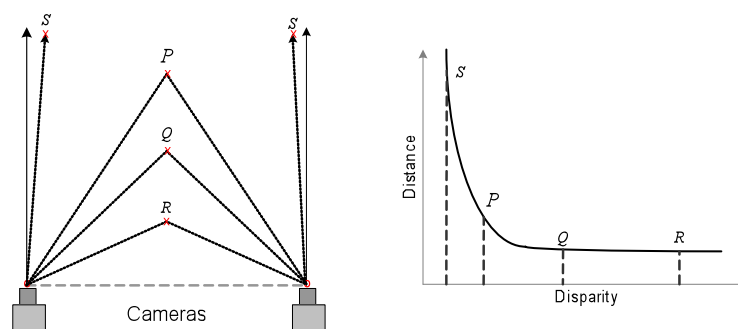


Figure 4.2: Relationship of depth and disparity [44]

The disparity is higher for points closer to the cameras. The disparity varies for objects at different depths and distances. The range field of the system is constrained by the disparity range $[d_{min}, d_{max}]$ with the baseline B and focal length f . The depth measured by a stereo vision system is discretized into parallel planes or layers (one for each disparity value). A better virtual discretization can be achieved with subpixel techniques.

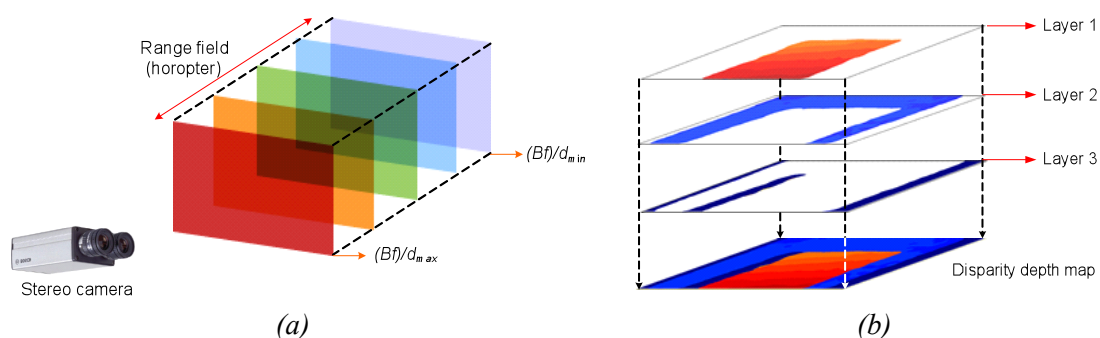


Figure 4.3: Disparity range and parallel planes (layers). (a) Disparity is higher for points closer to the camera. (b) Different disparity levels for disparity map.

The range field for different disparity and depth is shown in Figure 4.3(a). Based on the different layers of depth, the layers can be separated to focus on particular points or objects in the scene. The layers of disparity through the disparity range are shown in Figure 4.3(b), where the disparity depth map consists of three layers. Layer 1 in Figure 4.3(b) represents the range field of the scene that is the nearest to the camera that is with $(Bf)/d_{max}$. Meanwhile, Layer 3 is the farthest object from the camera with $(Bf)/d_{min}$ and Layer 2 lies between the two layers. With the disparity depth information, the new view synthesis based on layered representation can be performed as would be described in the next section. Disparity information plays an important role in synthesizing intermediate views from stereo images. The synthesized view quality depends mainly on the accuracy of disparity map. In this research, area-based method is used because disparity

information for every pixel is required.

4.2.2 Layers Based Disparity Depth Map Separation

Image Based Rendering (IBR) is known as an efficient way of generating novel views of real and synthetic objects [39]. Layered depth image is one of the scene representation categories from the geometry-based to more image-based approaches for the IBR. Table 3.3 in Section 3.6 summarizes the selected layered depth image techniques for the image-based rendering.

The layer extraction depends on the complexity of the scene that is related to the Light Field rendering. There is a relation between the complexity of the scene and the layer extraction. The complexity includes the variation, textures and occlusions. A simple scene with small depth variation only requires very few depth layers, which can be extracted very quickly, for example, testing for two different depths only. A scene with large depth variations requires more different rendering depths and therefore the layer extraction must test more depths layers.

4.2.3 Inter-view Synthesis Based on Disparity Depth Map

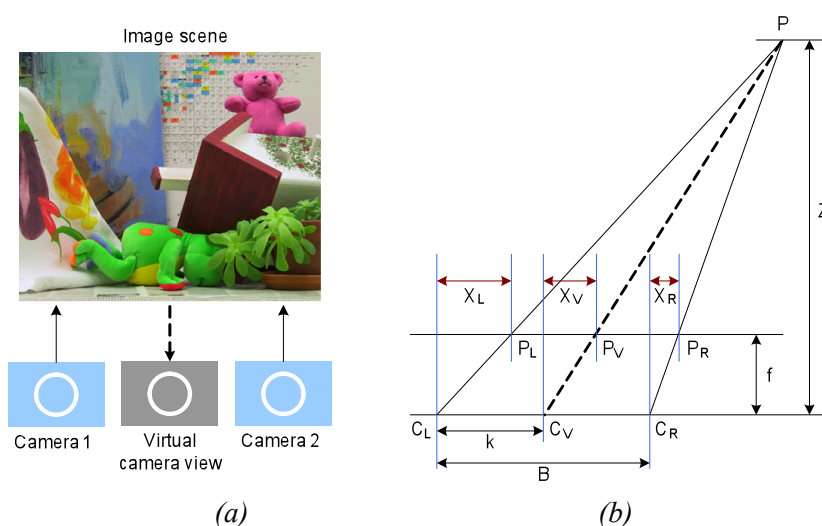


Figure 4.4: Inter-view synthesis: (a) The virtual camera view placed between camera 1 and 2. (b) Geometric stereoscopic camera model [167]

The main idea of view synthesis algorithm is to obtain the inter-view synthesis as shown in Figure 4.4(a) by using the disparity depth map layers representation. The virtual camera view for the image scene synthesized to be located between camera 1 and camera 2. The geometry model is shown in Figure 4.4(b), where two identical cameras

C_L and C_R on the same parallel coordinates are used and their image planes are coplanar [167, 168]. Let B denote the baseline distance and f denote the focal length. In this model, a 3D object point is simultaneously captured on both left and right viewing planes.

Disparity estimation should satisfy epipolar geometry constraint, for parallel camera configuration, the vertical component is equal to zero, so that only a one-dimensional search along the scan line is necessary. Assume that the scene point P from Figure 4.4(b) is projected into the left and right image planes at points P_L and P_R respectively. The disparity from left to right, where left image is the reference image, is given by:

$$d_{LR} = |X_L - X_R| = \frac{Bf}{Z} \quad (4.2)$$

where Z denotes the depth of point P . Now assume that the virtual camera C_V , corresponds to the intermediate camera, which is located between the left and right cameras and is related to the left camera by a distance of k . The disparity from left to intermediate is given by [167]:

$$d_{LV} = |X_L - X_V| = \frac{kf}{Z} = \left(\frac{k}{B}\right)d_{LR} = \beta d_{LR} \quad (4.3)$$

where $\beta = k/B$, $0 \leq \beta \leq 1$, indicates the baseline ratio between the left and right cameras. If the virtual camera C_V is assumed to be located in the exact middle of left and right image, then the value of β is 0.5. The C_V can be created along the disparity range position. In this chapter, the disparity that will be used is d_{LRC} , which is the disparity from left to right with the left as the reference and has been processed with left-right consistency check to determine the unmatched pixels.

Based on the calculated disparity maps over stereo image sequences, virtual views can be synthesized at any virtual camera position that is represented by the ratio of a baseline, β , which is the distance between the left and right cameras. The most popular method is Linear Interpolation (LI) [169-171], which can be written as [169]:

$$I'(x') = (1 - \beta)I_L(x + \beta d) + \beta I_R(x + (1 - \beta)d) \quad (4.4)$$

where I' is the virtual view image, I_L and I_R are the left and right images respectively, x and x' are pixel positions, d is disparity vector, and β was defined in Equation (4.3).

This method has been adapted into several view synthesis systems proposed by Lu [167, 168], Wang [21] and Jain [158]. A new virtual view synthesis method is proposed in the DILS algorithm, which will be described in the following section.

4.3 System Design Architecture

In this section, a novel intermediate view synthesis method based on disparity estimation depth map layers is presented. It contains two stages: stereo matching engine and a view synthesis module. In the first stage, disparity estimation, through area-based stereo matching algorithm, is used to obtain the disparity depth map. However, the stereo matching engine can be replaced by any stereo matching algorithms. Then, it will undergo the stereo matching computation and disparity refinement process. In the second stage, a new strategy for view synthesis is presented. It separates the depth layer of the disparity depth map based on the disparity range. The layers are divided into two main regions: non-occluded and occluded regions using their image histogram distribution. The non-occluded regions contain several layers depending on the complexity of the disparity depth map. Linear interpolation is used on the regions in different modes according to the characteristics of each region. After each of the layers has been interpolated, the layers are flattened into single novel view images.

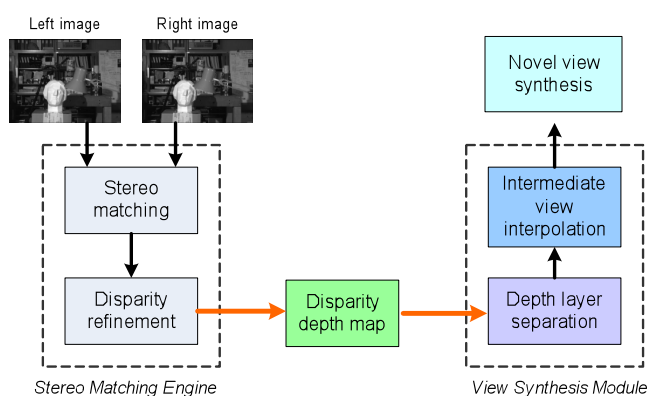


Figure 4.5: Block diagram of the proposed novel view synthesis based on depth map layers representation

One of the main advantages of the Depth Image Layers Separation (DILS) algorithm is that it can be performed with different stereo matching algorithms. In practice, view synthesis performed using a better disparity depth map. The proposed system design for matching and view synthesis is shown in Figure 4.5. It consists of two modules, which are the stereo matching engine and view synthesis modules. The system requires a

stereo pair using two synchronized cameras to acquire images or videos. Equivalent stereo imaging system may be built on two relatively inexpensive digital cameras provided with an external trigger capability and connected to a PC through a high-speed digital interface. The algorithm requires pairs of rectified images, so that corresponding epipolar lines are horizontal and on the same height.

The view synthesis module in Figure 4.5 consists of two main phases for the DILS algorithm; the depth layer separation and the intermediate view interpolation. In the first phase, the depth layer separation involves the histogram distribution, layers identification and layer separation. For the second phase of the DILS algorithm, the intermediate view interpolation synthesis includes the layer translation, mask layers, intermediate view interpolation and view synthesis. The stereo matching engine and view synthesis module will be described in Section 4.3.1 and 4.3.2, respectively.

4.3.1 Stereo Matching Engine

This section will discuss the stereo matching engine proposed in the novel views synthesis system architecture. It consists of two main stages: stereo matching and disparity refinement process. The main aim of stereo matching algorithms is to find homologous points in the stereo pair [46]. In stereo correspondence matching, the two images of the same scene are taken from slightly different viewpoints using two cameras that placed in the same lateral plane.

The matching pixels can be found by searching the element in the right image according to the similarity metric to a given element in the left image (a point, region or generic feature). In order to determine the correspondence of a pixel in the left image using a similarity metric, the window costs are computed for all candidate pixels in the right image within the search range. The pixel in the right image that gives the best window cost is the corresponding pixel of the left image. In this research, the SAD metric is selected for faster execution and low consumption. SAD metric also requires simpler arithmetic operation and performs better than the SSD approach in the presence of outliers [104].

Disparity information plays a crucial role in synthesizing intermediate views from stereoscopic images. The synthesized view quality depends mainly on the accuracy of disparity estimation [125]. The area-based method for disparity estimation is selected

because the disparity information for every pixel is required to synthesize the new virtual view. The block matching is applied where correspondence analysis is carried out on squared blocks of pixels. The disparity estimation process based on SAD correlation, left-right consistency check and the disparity refinement is described in Section 4.3.1.1 and 4.3.1.2.

4.3.1.1 Stereo Disparity Estimation

Assuming the stereo pair is in the same epipolar line, the disparity estimation is performed by using a fixed-size window. The SAD function is defined as follows:

$$SAD(x,y,d) = \sum_{i,j=-n}^n |I_R(x+i,y+j) - I_T(x+d+i,y+j)| \quad (4.5)$$

where $I_R(x, y)$ and $I_T(x, y)$ are the gray-level intensities of the reference (left) and target (right) image respectively, window size of $n \times n$, and d is the disparity.

The best disparity value is determined using the minimum SAD value. The conventional way to calculate the matching correspondence point is to fix a point and vary d in the disparity range to calculate the matching costs. Then simply select d with the smallest matching cost as the final disparity at this point. This method is also known as Winner-Take-All (WTA).

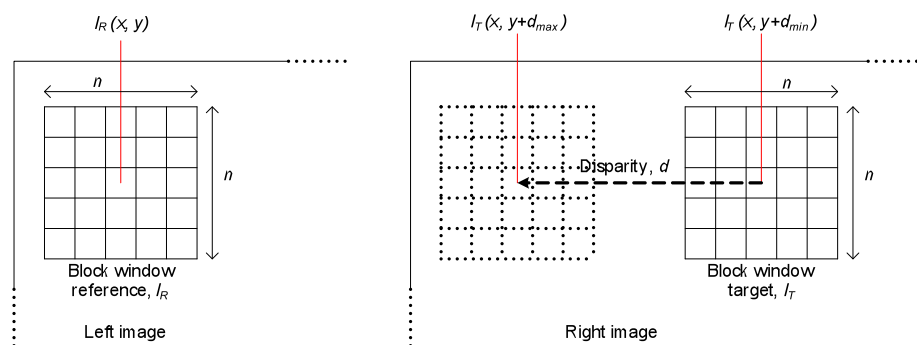


Figure 4.6: Matching costs computation based on window size, $n \times n$, and disparity range, d , with left image as the reference and right as the target image

As illustrated in Figure 4.6, the algorithm firstly sets one particular fixed value for d for all the points, and the matching costs are calculated for each image row. Then by varying d , the process is repeated until the value of d has iterated through the complete disparity range. Consequently a two-dimensional matrix containing the SAD values for each image row is obtained. The width of the matrix is the same as the length of image row, and the height of the matrix is the disparity range. The disparity value at (x, y) ,

$d(x, y)$ is the point where the correlation value of SAD is the smallest. Therefore, the disparity value may be expressed as follows:

$$d(x, y) = \arg \min_{d_{\min} \leq d \leq d_{\max}} SAD(x, y, d) \quad (4.6)$$

To ensure the consistency and accuracy of the disparity map, the matching process is performed in both directions. At the first stage, the left image is selected as the reference image and the right image as the target. The disparity map for this matching referred to as left to right disparity map, d_{LR} . A similar process is performed by having the right image as the reference, and the left image as the target image, to obtain the disparity map, d_{RL} . The result from both matching will be used for the next stage in the left-right consistency check.

4.3.1.2 Stereo Disparity Refinement

After the first stage in the stereo matching engine, the disparity map obtained through the stereo disparity estimation contains occlusions. Occlusions can create points that do not belong to any corresponding pixels. In many cases occlusions occur at depth discontinuities, where the occlusions on one image correspond to disparity jumps on the other. In the human visual system occlusions can help to detect object boundaries. However in computational stereo processing it is a major source of errors.

A typical method to deal with occlusions is Bidirectional Matching (BM) [106]. We apply a Left-Right Consistency Check (LRCC) proposed by Fua [24] based on two disparity maps that are created relative to each image: one for left-to-right (d_{LR}) and another for right-to-left (d_{RL}), as described previously. A valid correspondence should match in both directions. This operation is executed by taking the computed disparity value in one image and re-projecting it on the other image. The $d_{LR}(x, y)$ and $d_{RL}(x, y)$ disparities should satisfy Equation (4.7). If this is not satisfied, the $d_{RL}(x, y)$ is invalid and assigned the value of -1.

$$d_{RL}(x + d_{LR}(x, y), y) = -d_{LR}(x, y) \quad (4.7)$$

The LRCC operation can be illustrated in Figure 4.7 [125]. When searching for conjugate pairs, only the visible points on the image are matched. If the role of left and right images is reversed, new conjugate pairs can be found. The LRCC states that feasible conjugate pairs are those found with both direct and reverse matching. In

reverse matching, the conjugate pairs are equivalent to the uniqueness constraint, which states that each point on one image can match at most one point on the other image. Consider for instance an occluded point B , in the left image of Figure 4.7. Although it has no corresponding point in the right image, the matching cost minimization matches it to some point C' . With reverse matching, it corresponds to a different point in the left image. However, this information is available only when searching from right to left.

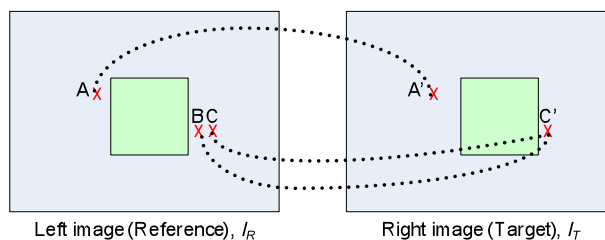


Figure 4.7: Occlusions in the left and right image.

Although the disparity map has been improved with the LRCC operation, it still suffers from noises. The disparity maps can be refined using image filtering techniques without explicitly enforcing any constraint about the underlining disparity maps. A common image filtering operator used is the median filter due to the fact that it preserves edges whilst removing noise [104]. The filtering of the disparity map can improve the results in weakly textured regions, where the signal to noise ratio is low and often some pixels are rejected although the disparity can correctly be estimated in the neighbourhood. The filtering significantly reduces the disparity depth map noise while it smooths the depth map. In this research, the median filter parameter used is 11×11 . The resulting disparity depth map is known as the d_{LRC} , and is used to generate the novel view synthesis in Section 4.3.2.

4.3.2 Inter-View Synthesis with DILS

The main idea of inter-view synthesis is to separate the depth map into several layers of depth based on the disparity distance of the corresponding points. The new view synthesis is interpolated independently for each layer of depth from the left and right part of the image by masking the particular depth layer. The separation process of the layers is carried out after identifying the number of depth layers on the disparity depth map. By having the result in the image form, the subject can be easily known through different tones of greyscale or colours. The disparity distribution is obtained using the histogram plot.

A block diagram of the Depth Image Layers Separation (DILS) is illustrated in Figure 4.8. It consists of several important steps to obtain the inter-view synthesis of the stereo images. In the stereo matching engine, the raw disparity depth map is obtained from the stereo matching algorithm processes with the stereo disparity refinement. In this stage, the depth map addresses the presence of occlusions through left-right consistency to compute disparity and its associated uncertainty to eliminate false matches. A fast median filter was applied to the resulting map to further remove outliers and produce a smooth depth map. After this stage, the disparity depth map, defined as d_{LRC} , which signifies the left-right consistency check is based on the left image as the reference. The d_{LRC} is the main input for the DILS algorithm as shown in Figure 4.8.

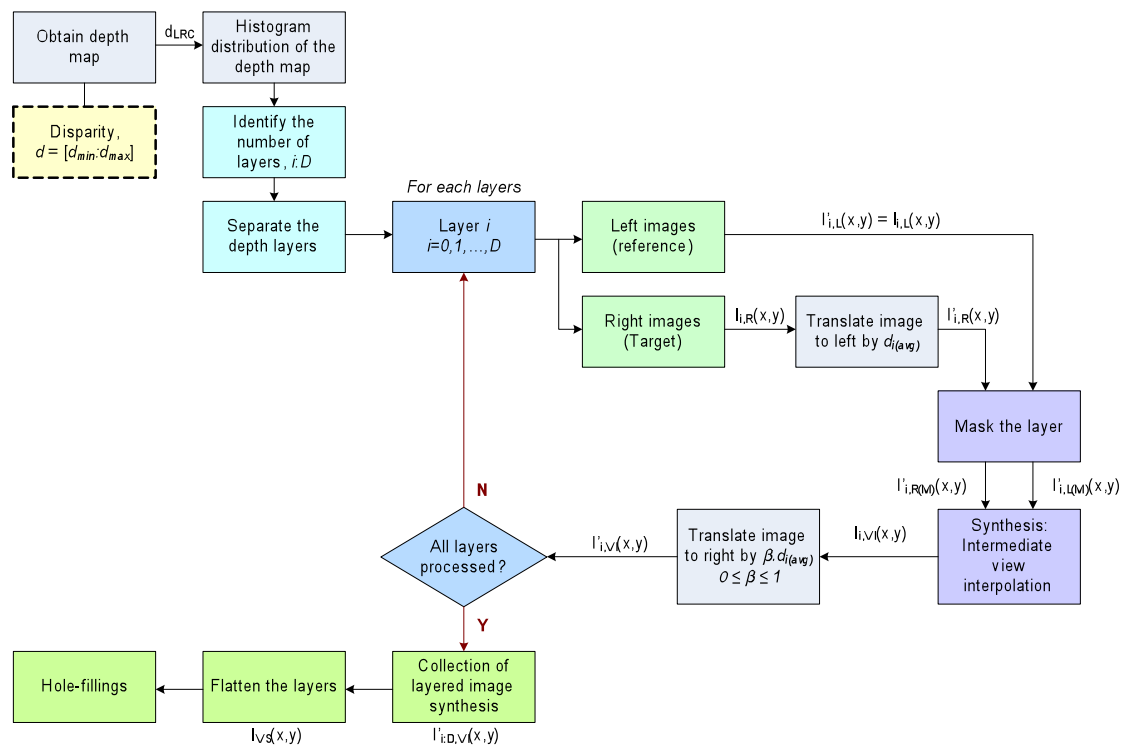


Figure 4.8: Depth Image Layers Separation (DILS) algorithm on the view synthesis module

The view synthesis module consists of two main phases: depth layer separation and intermediate view interpolation synthesis. In the first phase, the disparity depth image map is divided into several layers depending on the complexity of the image pairs. It is essential to have a good and smooth disparity depth map where the layers of disparity depth map can be segmented into clusters. The numbers of matched pixels, p in each disparity levels (horopter) of the disparity depth map image, d_{LRC} can be distinguished from the histogram distribution (refer to Figure 4.8). Each disparity levels, d quantized based on selected threshold to define and identify the layers from layer 0 to D .

In the second phase, each layer is masked based on whether they represent non-occluded or occluded regions. This phase is defined in the iteration process for every layer that is identified in the DILS algorithms as shown in Figure 4.8. The left and right images synthesized to produce the intermediate view interpolation for that particular layer, where the layer consists from 0 to D (D is the maximum layer for the stereo pair). The intermediate virtual view, $I_{i,VI}(x, y)$ image is located between the left and right image for layer i . Therefore, the $I_{i,VI}(x, y)$ is translated to the right by $\beta \cdot d_{i(avg)}$, where $d_{i(avg)}$ is obtained through the layer identification process based on the minimum and maximum disparity value for the layer i . The value of β is obtained from Equation (4.3), where $0 \leq \beta \leq 1$. Assuming the new intermediate view of layer i is exactly in the middle of the left and right images, then β is 0.5. To ensure consistency throughout the layers, the value of β remains the same for the translation after the intermediate view interpolation and synthesis operations. After all layers have been processed, the collection of layered image synthesis flattens into a single image as the inter-view image synthesis for the image pair, $I_{VS}(x, y)$. The hole and cracks in the final inter-view image will be corrected in the hole-fillings technique. The following section discusses the algorithm in detail with the mathematical framework for the DILS algorithms based on Figure 4.8.

4.3.2.1 Histogram Distribution

Histogram distribution is used in the disparity depth map to distinguish the number of matched pixels in each disparity level. For simpler depth maps, it consists of a few numbers of matched pixels depending on the disparity levels. However, for complex depth maps, the distributions of matched pixels will be spread out along the disparity pixel values from minimum to the maximum value of the disparity.

A histogram is a table that simply counts the number of times a value appears in the data set. In image processing, a histogram is a histogram of sample values [172]. For an 8-bit image, there will be 256 possible values in the image and the histogram will simply count the number of times each value actually occurs in the image.

Consider the disparity depth map as an N -bit for a $W \times H$ size greyscale image. There are d distinct sample values that could occur in the disparity depth map, depending on the disparity levels. The histogram of the disparity depth map image comprises a table of d

values (from d_{min} to d_{max}) in integer, where the k^{th} entry in the histogram table contains the number of times a sample of value k occurs in the image map. If the image was not entirely matched, for example, the 0^{th} entry in the table would contain a value of $W \times H$ and all other table entries would be zero. In general, for a $W \times H$ disparity depth image where the k^{th} sample is known to occur n_k times, the histogram h is formally defined by Equation (4.8), where d_{max} is the maximum disparity value:

$$h(k) = n_k \quad k \in 0, 1, \dots, d_{max} \quad (4.8)$$

For the disparity depth map, the horizontal axis of the histogram refers to the disparity range levels and the vertical axis corresponds to the number of matched pixels count. The shape of a histogram does not convey much useful information but there are several key insights that can be gained. The spread of the histogram is directly related to the distribution of the disparity depth map levels, where close histogram distribution is a representative of complex stereo matched images and a wide distribution represents plain and simple stereo matched images.

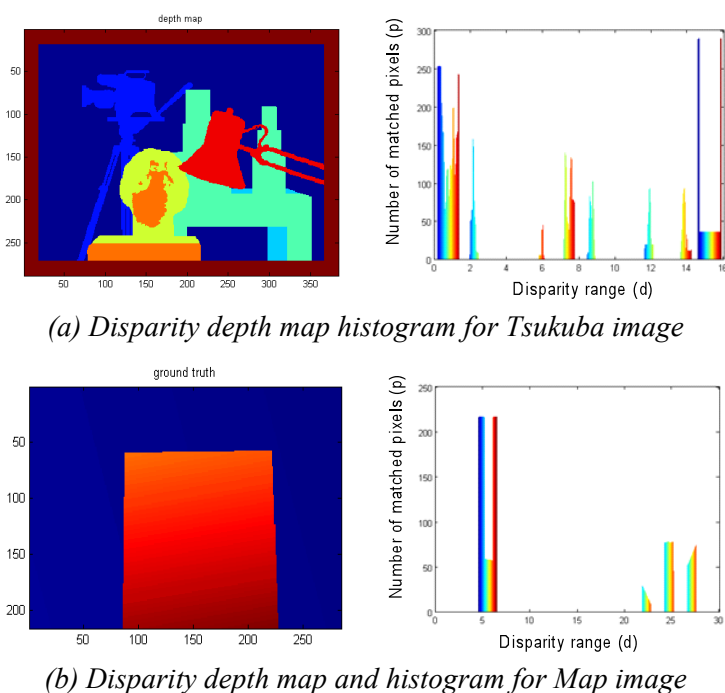


Figure 4.9: Disparity depth map and the histogram distribution based on the ground truth image: (a) Disparity depth map and its corresponding histogram for Tsukuba image. (b) Disparity depth map and its corresponding histogram for Map image.

Figure 4.9 shows the disparity depth maps and the histogram distributions for the two ground truth of the stereo pairs. The ground truth is a disparity map with accurate disparity values obtained, either by using the range sensor cameras, or which are

manually labelled and calculated based on piecewise planar surfaces, as given by Middlebury datasets [22]. Figure 4.9(a) presents a relatively complex disparity depth map image and its corresponding histogram. The disparity depth map consists of several objects through disparity planes based on the distance from the camera. The histogram is distributed unevenly based on the matching pixels and has a relatively close distribution since most of the samples are narrowly separated. Meanwhile Figure 4.9(b) gives a simpler disparity depth map image and its corresponding histogram with only few widely separated samples. Most of the matched pixels samples fall within the range of approximately 5 to 7, while relatively few of the matched pixels samples are between 21 and 30. The main interest of the histogram distribution is that the matched pixels are close to the d_{max} . These are the closest objects to the camera that represents the foreground object.

4.3.2.2 Layers Identification

The next step is to identify the layers on the disparity depth map as illustrated in Figure 4.8. Basically, the layers can be recognized immediately with the disparity level values $[d_{min}, d_{max}]$, also known as horopter on the stereo matching algorithms. The samples of the matched pixels can be grouped to clusters of layers according to the threshold histogram distribution. Figure 4.10 illustrates the example of histogram distribution for a disparity depth map that consists of the disparity range value levels, can take values between 1 and d_{max} , 30. The layer can be easily identified with the number of matched pixels, p , quantized according to the following equations:

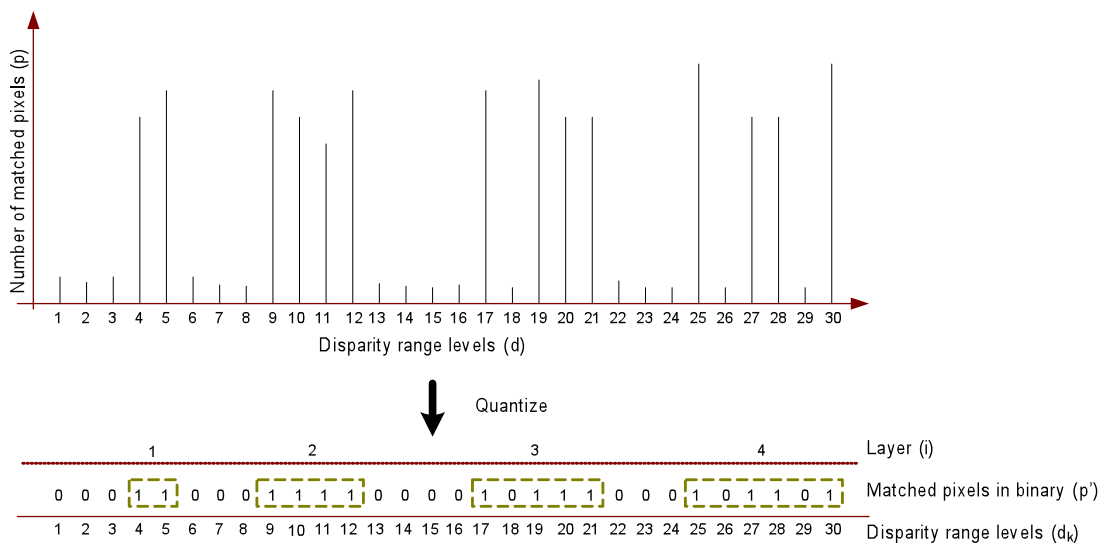
$$\begin{aligned} p'(d_k) &= 1, \quad \text{if } p(d_k) > T \quad k \in 0, 1, \dots, d_{max} \\ p'(d_k) &= 0, \quad \text{elsewhere} \end{aligned} \quad (4.9)$$

where T is a threshold to set the minimum number of pixels to be selected as the matched corresponding points for the stereo pair. Otherwise, it will be set as occlusion. In this research, we set the T parameter as 20. The output from the quantization is shown in Figure 4.10. The threshold values that set the pixels of p' to '1' can be grouped as one layer with non-linear segment of d into D layers. The clustering process to identify the layers uses the algorithm that is similar to zero run-length in the image compression algorithms but with some modification required to accommodate the searching process of the first and the last '1' in the layer or cluster.

In order to extract the layers the threshold matched pixels, p' can be clustered and grouped as follows,

- Group of continuous zeros are regarded as non-layers.
- Group of continuous ones (including a single '1') are regarded as layer. For example, layer 1 and 2 in Figure 4.10.
- Isolated individual zero surrounded by 1's in a group are regarded as layer in that group. For example, layer 3 and 4 in Figure 4.10.

Here, we assumed that the isolated individual zero surrounded by 1's in a group is not repetitively for the whole stream.



Notation:

i : refers to the number of layers index, $0:D$

d : disparity range levels, $d_{min}:d_{max}$

k : refers to disparity range levels index

p : number of matched pixels

p' : threshold matched pixels

Figure 4.10: Histogram distribution of the disparity depth map and the matched pixels in binary after the quantization

Basically, the layers can be grouped into two main regions:

- a) *Non-occluded region.* This layer is for the non-linear segment d_k into $(i:D)$ layers (clusters), where $i \in 1,2,\dots,D$ and $k \in 0,1,\dots,d_{max}$.
- b) *Occluded region.* For part complete and occluded disparity map, it will be labelled in layer 0.

With the layers identified, we can determine the minimum ($d_{i(min)}$) and the maximum ($d_{i(max)}$) disparity range levels for each layer. For example (taking the histogram distribution of depth map from Figure 4.10), the minimum and maximum disparity range for each layer can be defined as follows:

Table 4.1: Sample of disparity range levels for different layers

Layer (i)	$(d_{i(min)}, d_{i(max)})$
1	(4,5)
2	(9, 12)
3	(17, 21)
4	(25, 30)

The main objective of this algorithm is to find the group of ‘1’ in the threshold matched pixels data set. As described earlier, a string of ‘1’s separated by only a single ‘0’, considered as one layer or cluster. Otherwise, they will be identified as a different layer. Figure 4.11 shows an example of the indexes for the threshold matched pixels data, p' . The disparity range value shows the number of threshold data in array form, where k is the index of threshold matched pixels data, p' , and d_{max} is the maximum index or the maximum disparity in the case for the stereo pair. The i_F denotes the index of the first ‘1’ in the cluster or layer, while i_L denotes the last index that signifies the last ‘1’ of the i^{th} layer. The value of i_F determine the $d_{i(min)}$ and i_L represents the value of $d_{i(max)}$ for layer i . The data samples are the threshold matched pixels in binary, p' , with the index of disparity range levels ($k, k+1, \dots, d_{max}$).

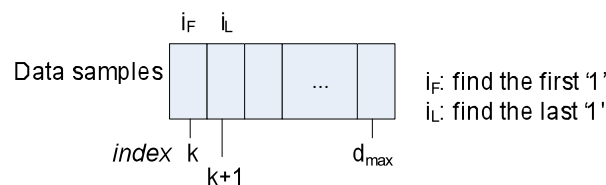


Figure 4.11: Identify the layer on the threshold data samples from index k to d_{max}

Figure 4.12 illustrates the algorithm functionality to obtain the $i_F = d_{i(min)}$ and $i_L = d_{i(max)}$ for layer i . The entire index here is in integer value data type. As described earlier, the main purpose of i_F is to find the first ‘1’ in the data sample. During the first stage, the i_F is initialized to index 1 and i_L to the next index, which is index 2. The i_F checks the data value of index 1. If the data is ‘1’, then i_F immediately halt on that index. The $d_{i(min)}$ is now set as the i_F , with the index 1. The objective of i_L is to search the last ‘1’ occurrence in the data sample of p' . The starting point for i_L is in index 2, just beside the index i_F .

The i_L checks the value in index 2 and also the next index (i_L+1) to search for the next '1'. If the data in the next index (i_L+1) is also '1', then the i_L moves to the next index that is now in index 3, as shown in Figure 4.12(a). A similar process is carried out in this state, where the i_L check the current data in $p'[i_L]$ and the data in the next index, $p'[i_L+1]$ until the data '0' in p' appears more than once. In this case, the i_L stops at index 4 when the remaining next two indexes are zero. For the example in Figure 4.12(a), the values obtained for layer $i=1$ are $d_{i(\min)}=i_F=1$ and $d_{i(\max)}=i_L=4$, where the data sample of p' is grouped as layer 1 from index 2 to 4.

In Figure 4.12(b), i_F found the first '1' in the index 2. The i_L moves from index 3 to 4 when the value in $p'[4]$ is '1'. Since the remaining next two indexes are zero, i_L stops at index 4. With this, the $d_{i(\min)}$ is set to 2 and $d_{i(\max)}$ is 4. The example in Figure 4.12(c) is almost similar to Figure 4.12(b). However, the i_L continuously moving to the next index and stops at index 6 when the data in the remaining index 7 and 8 are '0'. Therefore, the values obtained in this case are $d_{i(\min)}=2$ and $d_{i(\max)}=6$.

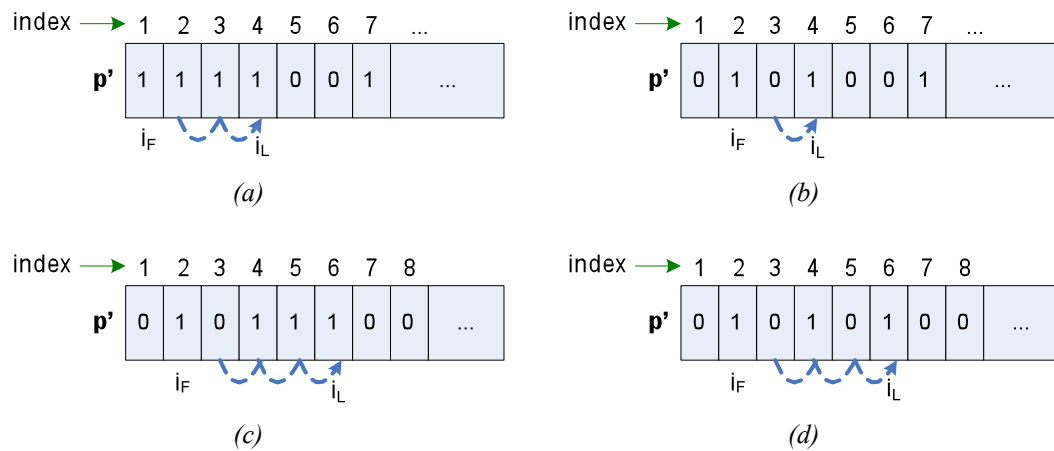


Figure 4.12: Finding the i_F and i_L to distinguish the layer i in the threshold data sample p'

For the example in Figure 4.12(d), the data of p' contained the '0' that appeared non-continuously after the i_F holds the first '1' at index 2. The i_L moves from index 3 to 4 when the value of $p'[4]=1$. It cannot stop at index 4 as the last value of '1' since the next two index contained '1', which is in this case at index 6. Therefore the i_L moves to index 6 as the last '1' in the cluster when the remaining next two indexes do not contain any '1' (at index 7 and 8). The $d_{i(\min)}$ and $d_{i(\max)}$ corresponds for the i^{th} layer sample are $i_F=2$ and $i_L=6$ respectively.

The appearance of a single '0' in the data sample is detected by the exclusive OR Boolean operations. This process will be done only when the i_F has already detected the first '1' in the data stream of p for layer i . The $flag$ is defined to check any changes of data in the sample, where $flag = p'[i_L] \oplus p'[i_L+1]$. If there is any change of data, the $flag$ is set to 1 and it will signal the i_L to increase by 1, which is to check the data is in the next index if it contains value of '1'. The i_L keeps on moving to the next index until two consecutive pieces of data in the index (i_L+1 and i_L+2) contained the value of '0'. Then the $flag$ is set to 0 and stops with $d_{i(max)} = i_L$. With this case, the iteration process of finding i_F and i_L for layer i is stopped and moves to search for the next layer. The iteration to identify the layer i , $d_{i,min}$ and $d_{i,max}$ stopped when i_L is more than the value of the maximum disparity levels for the stereo pair, d_{max} .

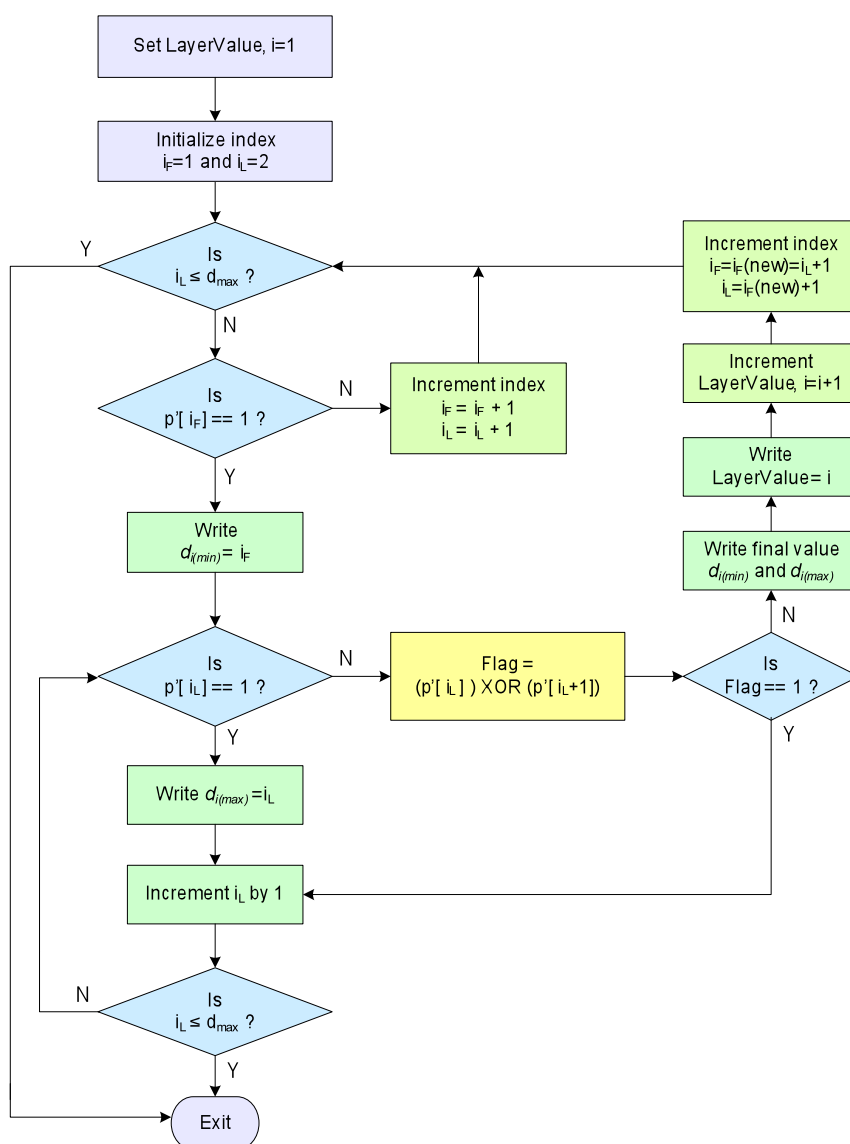


Figure 4.13: The zero run-length algorithm to determine the non-linear segment layer d into D layers

The summary of the algorithm to segment the layer based on the zero run-length is given in Figure 4.13. The initialization value is set as layer, $i=1$, $i_F=1$ and $i_L=2$. When the first '1' is found in the data, $d_{i(min)}$ is given as $d_{i(min)}=i_F$. Then, the algorithm looks for the last '1' in the cluster searching throughout the data until the flag is nonzero. With that, $d_{i(max)}=i_L$. The new search $d_{i(min)}$ and $d_{i(max)}$ for next layer starts again with the increment of layer, i by 1. The process ends when it reaches the end of the data stream with the index of d_{max} .

Another approach to group the '1' as a continuous stream is to search for a single occurrence of '0' in data p . When the single '0' is found in the data sample, it will be change to '1'. For example, when the data stream is '00110100', it will be transformed to '00111100'. The single occurrence of '0' after i_F can be identified by the following equation:

$$p'[i_L] = p'[i_L] \vee (p'[i_L + 1] \wedge p'[i_L - 1]) \quad (4.10)$$

where \vee and \wedge symbolize OR and AND logical operation respectively. The continuous data stream of '1' will be group as a single layer with the algorithm shown in Figure 4.14.

The algorithm in Figure 4.14 is the similar technique presented in Figure 4.13. The flag process has been eliminated since it is not required and the process has been done by Equation (4.10). However, this process is time consuming and not cost effective since it requires additional search runs to find a single '0' in the data stream before the zero run-length algorithms can be performed.

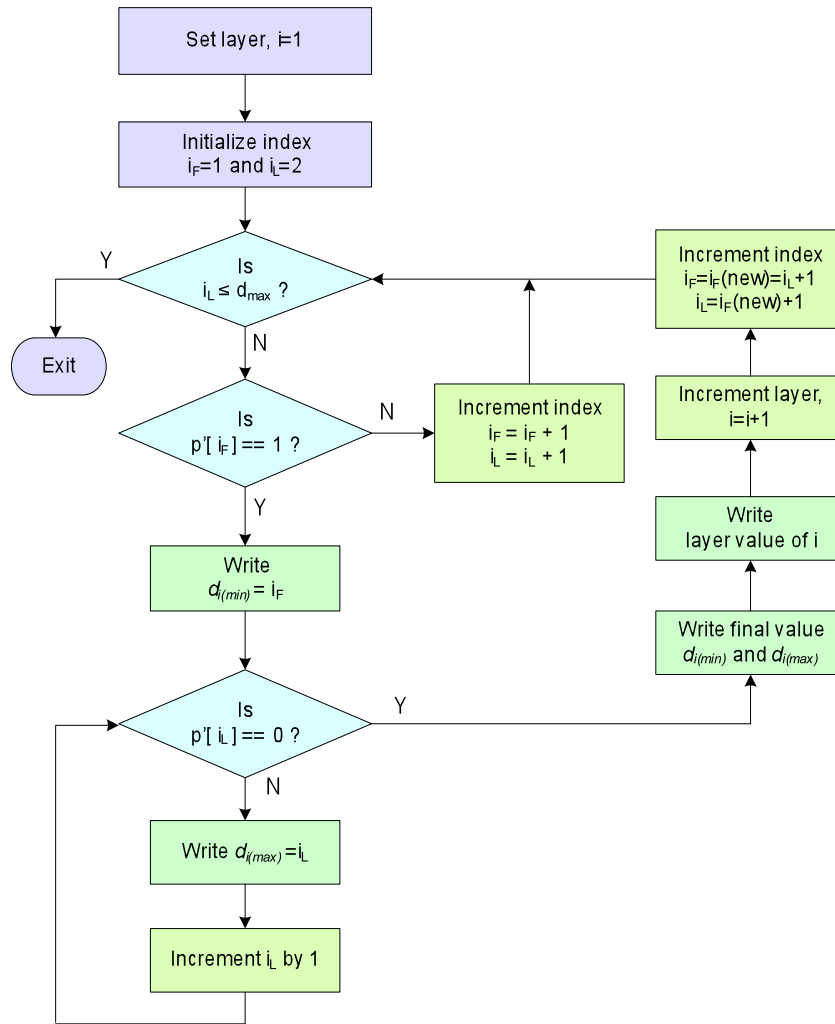


Figure 4.14: New zero run-length algorithm to determine the non-linear segment layer d into D layers for the data stream without single '0' occurrence

4.3.2.3 Layers Separation

During this stage, the number of layers has been identified with the maximum of D . As illustrated in Figure 4.8, this is a starting point for the iteration process to synthesize each layer. The disparity depth map separated to D layers based on the cluster of i , where $i=1,2,\dots, D$ with the known $d_{i(min)}$ and $d_{i(max)}$ for the i^{th} layer. As described earlier in previous section, there are two regions identified for the layers, which are the non-occluded and occluded regions. The new mask, $M_i(x, y)$ for the disparity depth map created for each layer with the disparity depth map, $d_{LRC}(x, y)$ based on the Equation (4.11) for non-occluded region. Basically, it is a binary mask layer created into a number of D layers.

a) Non-occluded region:

$$\begin{aligned} M_i(x,y) &= 1 && \text{if } (d_{i(\min)} < d_{LRC}(x,y) < d_{i(\max)}) && i \in 1,2,\dots,D \\ &= 0 && \text{elsewhere} \end{aligned} \quad (4.11)$$

Equation (4.12) defined a special mask for layer 0, $M_0(x, y)$, which is known as the occlusion region. Any unmatched correspondence pixel points for the stereo matching algorithm will fall within this category.

b) Occluded region:

$$\begin{aligned} M_0(x,y) &= 1 && \text{if } [d_{LRC}(x,y) = 0] \\ &= 0 && \text{elsewhere} \end{aligned} \quad (4.12)$$

Figure 4.15 shows an example of the disparity depth map with only two layers, and no occluded region. With the separation layer process, there are two new binary mask layers created. The first layer defines with the smallest disparity d , which is the background of the depth map. The selected region is set to '1', while the other region is set to '0'. Meanwhile, in layer 2, the foreground region is selected and the mask of the second layer is set to '1' as shown in Figure 4.15. The number of binary layers is created based on the number of D layers identified that are dependent on the complexity of the obtained disparity depth map. It is important to yield the correct and smooth depth map in order to define the accurate layer masks since the algorithm depends on accuracy of the disparity depth map.

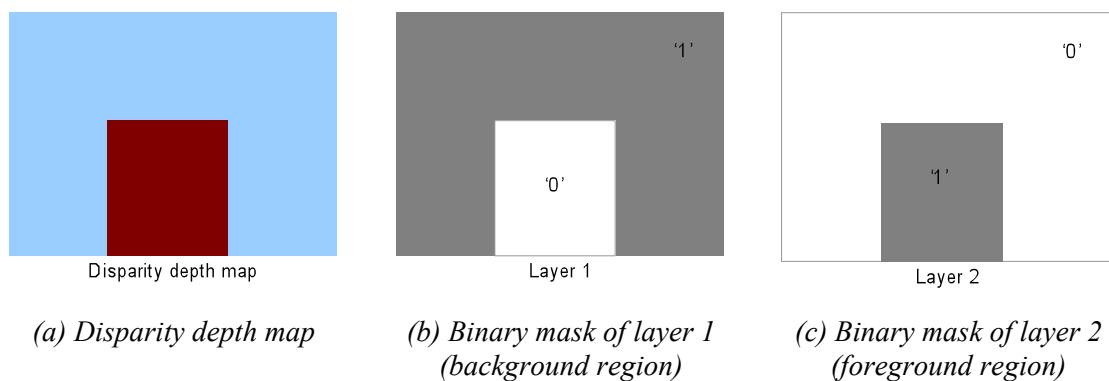


Figure 4.15: Example of layer masks based on layer separation process

4.3.2.4 Image Translation (Left)

The original disparity depth map, d_{LRC} is obtained with the left-right consistency check as illustrated in Figure 4.8. The L-R considers the left image as the reference and the right as the target image. Therefore, the right image requires to be translated to the left on horizontal plane only, which reflects disparity depth map, d_{LRC} . The object region of the right image on the layer is positioned on the same location in the left image. When the same corresponding pixel points are located in the same position, the intermediate view interpolation can be easily obtained in the next stage. Before the image can be combined with the mask layer, the right image is translated horizontally to the left using on the following equations:

$$\text{int}[d_{i(\text{avg})}] = (d_{i(\text{min})} + d_{i(\text{max})})/2 \quad (4.13)$$

$$I_{i,L}'(x,y) = I_{i,L}(x,y) \quad (4.14)$$

$$I_{i,R}'(x,y) = I_{i,R}(x - d_{i(\text{avg})}, y), \quad i \in 0, 1, \dots, D \quad (4.15)$$

Since the left image is the reference image, it does not undergo the translation process as the right image. For each layer i , the proposed algorithm consists $d_{i(\text{min})}$ and $d_{i(\text{max})}$ which show the range of the pixel translation value for this translation. In this case, the average value $d_{i(\text{avg})}$, which is calculated $d_{i(\text{min})}$ and $d_{i(\text{max})}$ will be used to translate the right image to the left for the i^{th} layer. The value of $d_{i(\text{avg})}$ in the form of integer data type is calculated using the range of $d_{i(\text{min})}$ and $d_{i(\text{max})}$ for the layer i . An example for this process is illustrated in Figure 4.16. The process is repeated for every layer from i to D , including the occlusion layer $i=0$.

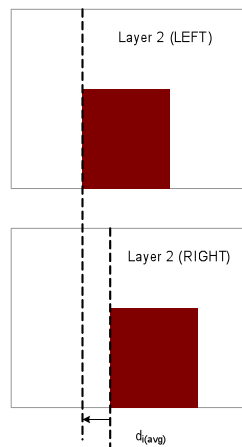


Figure 4.16: Translation process for right image to the left by d value

4.3.2.5 Mask Layers

After the left and right images for the disparity depth i have been located to the same pixel location as the mask region, we can then proceed to the next stage which is to mask the layers as indicated in Figure 4.8. The left and right images merge with the respective mask of the i^{th} layer using Equation (4.14):

$$\begin{aligned} I_{i,L}(M)(x,y) &= I_{i,L}'(x,y) \cdot M_i(x,y), \\ I_{i,R}(M)(x,y) &= I_{i,R}'(x,y) \cdot M_i(x,y), \quad i \in 0,1,\dots,D \end{aligned} \quad (4.16)$$

This stage will be iterated for every layer. The new image based on mask layers, $I_{i(M)}(x, y)$ obtained for the left and right image. The mask region of '1' will be replaced with the pixel values of image left and right corresponding to the pixel locations and the disparity depth.

4.3.2.6 Synthesis: Intermediate View Interpolation

Now, two identical images have been obtained for a particular region of interest based on the disparity depth as illustrated in Figure 4.8. The new intermediate view, $I_{i,VI}(x, y)$ is generated through the interpolation process using the left and right image masked layers ($I_{i,L(M)}(x, y)$, $I_{i,R(M)}(x, y)$). The interpolation combines the pixel values for the left and right image mask layers. However, the summation of the pixel values could exceed the normal value, for example in greyscale if the left and right pixels both have the value 128, it could produce the new pixel value of 256, which is not the anticipated value for the pixel. The weighted element given in the intermediate view interpolation process known as alpha blending, α , can threshold the pixel value for the left and right images. With alpha blending the expected pixel values negotiated between the left and right image mask layers. Considering that the pixels for the left and right are almost similar, the value of the α can be set to 0.5, taking the nearest average pixel values between the stereo pair. The intermediate view $I_{i,VI}(x, y)$ can be defined as follows:

$$I_{i,VI}(x,y) = \alpha I_{i,L(M)}(x,y) + (1 - \alpha) I_{i,R(M)}(x,y), \quad i \in 0,1,\dots,D \quad (4.17)$$

where ($0 \leq \alpha \leq 1$).

The value of $I_{i,VI}(x, y)$ is obtained for each layer of i through the iteration process. The sample of the intermediate view interpolation is shown in Figure 4.17, where the region is only interpolated based on the selected mask regions.

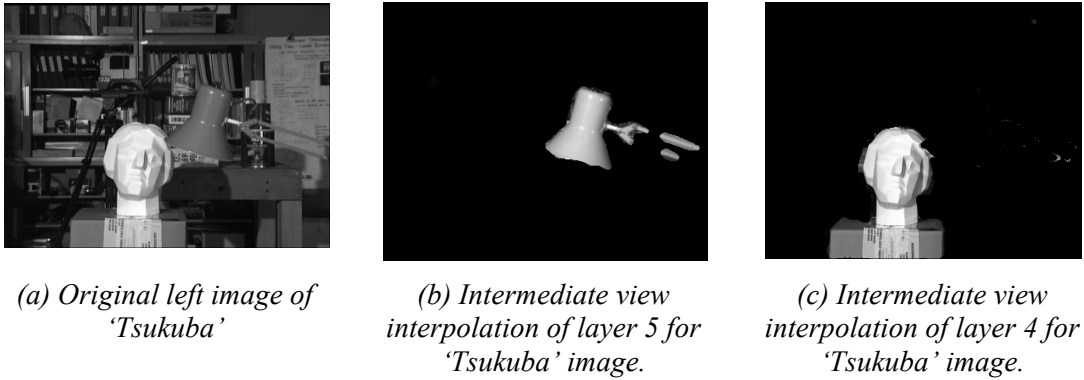


Figure 4.17: Sample of inter-view interpolation at different layers.

4.3.2.7 Image Translation (Right)

As discussed earlier, the disparity depth map uses the left image as the reference. During the first translation stage, the right image has been translated to the left, which is reflected on the left image as the reference. Therefore, after the intermediate view interpolation process, the new image layer i , $I_{i,VI}(x, y)$ undergo as the same translation process. In spite of translation of the image to the left, it will be translated to the right by $d_{i(avg)}$ and limited by the camera baseline ratio of β as defined in Equation (4.3). If the new view synthesis is located exactly in the middle of left and right, then the β value is set to 0.5. The translation to right on horizontal is defined in the following:

$$I_{i,VI}'(x, y) = I_{i,VI}(x + \beta d_{i(avg)}, y), \quad i \in 0, 1, \dots, D \quad (4.18)$$

where ($0 \leq \beta \leq 1$).

To ensure consistency throughout the layers, the value of β must be constant in every translation process from layer i to D after the intermediate view interpolation and synthesis operations.

Figure 4.18 shows an example of the translation process for the view synthesis based on layer 2. The image view interpolation, $I_{2,VI}(x, y)$ is translated to the right by $\beta \cdot d_{2(avg)}$. Assuming the value of $d_{2(avg)}$ for this layer is 7. In this example, the new inter-view image is translated to the middle (between the left and right image). Therefore, β is set to 0.5. The intermediate view interpolation of layer 2 is given by:

$$\begin{aligned} I_{2,VI}'(x, y) &= I_{2,VI}(x + \beta d_{i(avg)}, y) \\ &= I_{2,VI}(x + (0.5)(7), y) \end{aligned} \quad (4.19)$$

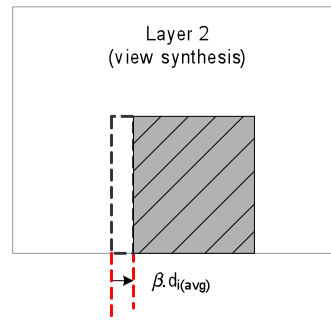


Figure 4.18: The translation process to the right based on the disparity range value

4.3.2.8 View Synthesis

At this stage, all of the layers have undergone the process of translation, masking and synthesis as shown in Figure 4.8. A number of image view interpolations, $I_{i,VI}(x, y)$ is collected from $i=0$ to D . The final inter-view synthesis of the image pair can be generated when all the layers are flattened into a single layer based as follows:

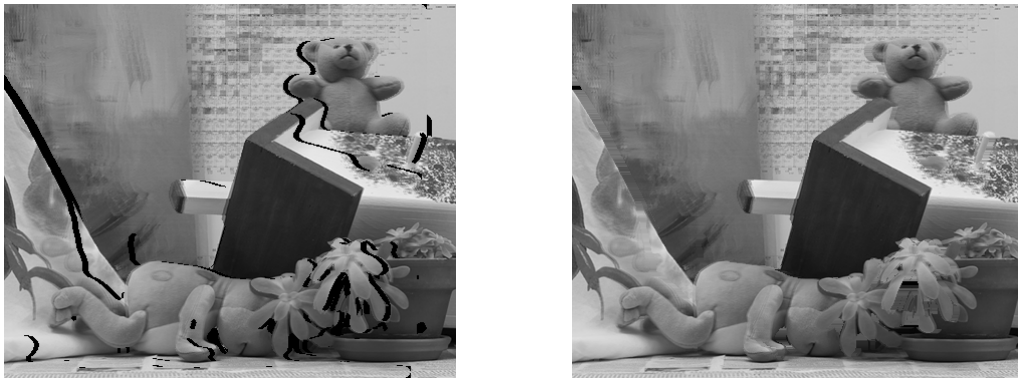
$$I_{VS}(x, y) = \sum_{i=0}^{i=d_{\max}} I_{i,VI}'(x, y), \quad i \in 0, 1, \dots, D \quad (4.20)$$

To ensure the accuracy of inter-view images to be composed into a single layer, the value of β in the translation process should be consistent for all layers. One of the advantages of this approach is that, several number of view synthesis can be synthesized with different β values to corresponds the location of virtual views. Besides, the newly synthesized virtual view for each layer can be decomposed into a different image layers with the same image size.

4.3.2.9 Hole-Filling

Holes and cracks between layers in the interpolated image may arise from layer image translation when mapping the layers into a single image as indicated in Figure 4.8. This case is shown in Figure 4.19, where the layers are flatten into a single virtual view image. The holes are identified by zero pixels (dark regions) in the image as shown in Figure 4.19(a). An efficient method to overcome this error is to fill the holes by the adjacent pixels or offset vectors through the horizontal plane. Although this approach is less accurate, the holes can be easily corrected without reducing the quality of the image. Figure 4.19(b) shows the results for applying the hole-fillings process to complete the whole image. Notice that with the hole-fillings technique, the new image

has been transformed significantly with the cracks and holes cleanly recovered.



(a) The synthesis view image with holes

(b) The synthesis view image with hole-fillings

Figure 4.19: Holes in the final virtual view images.

4.4 Results and Discussion

This section gives a detailed evaluation of the proposed algorithm in terms of results and quality. In order to validate the performance of the DILS framework, we performed the experiments with the Middlebury test scenes [22]. We use the following test data sets: ‘Teddy’ (450 x 375 pixels, search range: 60), ‘Venus’ (434 x 383 pixels, search range: 20), ‘Tsukuba’ (384 x 288 pixels, search range: 16) and ‘Cones’ (450 x 375 pixels, search range: 60). The ‘Tsukuba’ image set contains five images (views 1-5), and the ‘Venus’, ‘Teddy’ and ‘Cones’ image sets contain nine images (views 0-8). The Middlebury image sets are performed with views 1 and 5 for the ‘Tsukuba’ image sets and views 2 and 6 for the ‘Venus’, ‘Teddy’ and ‘Cones’. The synthesized view images created for the camera baseline ratio $\beta=0.25$, 0.5 and 0.75 at the views (2, 3, 4) for ‘Tsukuba’ and (3, 4, 5) for the remaining data sets. The samples of left images for the Middlebury data sets are shown in Figure 4.20. The left view images and their corresponding ground truth disparity maps are shown in Appendix C.



(a) Teddy

(b) Venus

(c) Tsukuba

(d) Cones

Figure 4.20: Middlebury data sets for the left image

The proposed method was tested using the same parameters for all the test images. The weighted factor for alpha blending was $\alpha=0.5$ in the intermediate view interpolation process. The sizes of the sets of neighbouring pixels in the window searching cost were 21×21 ('Cones' and 'Teddy'), 11×11 ('Tsukuba') and 25×25 ('Venus'). In the first evaluation, the reconstructed virtual images were performed and synthesized by conventional inter-view interpolation and DILS algorithms. The quality of the synthesized novel views that were obtained by the virtual camera with these approaches were measured and analyzed using PSNR and SSIM index performance measurements. As described earlier, the DILS algorithm can be implemented and adapted to different stereo matching algorithms. In order to justify this statement, the second evaluation presented and analyzed with two disparity depth maps, which were obtained using a basic fixed-window similarity metric (SAD) and cross-based cost aggregation method. The selected stereo matching system in the cross-based cost aggregation method is known as AD-Census [130], which is the top performer in the Middlebury benchmark. The DILS algorithm reconstructs the novel views between the left and right images based on the disparity depth map yielded using SAD and AD-Census. The performance of the algorithms was compared in term of PSNR and SSIM index based on synthesized views.

4.4.1 Performance Evaluation of Conventional Linear Interpolation and DILS

The first evaluation is performed based on the conventional Linear Interpolation (LI) (from Equation (4.4)) and DILS algorithms. In this assessment, the reconstructed novel view images for the virtual camera views are compared with the original images captured by the camera of the same views. For example, the real camera views for 'Teddy', 'Venus' and 'Cones' located at camera 2 to 6 and for 'Tsukuba' at 1 to 5. The results of PSNR and SSIM are calculated based on the conventional inter-view interpolation and DILS algorithms for each camera views.

The ground truth of the Middlebury data sets is shown in Figure 4.21(a) for the 'Teddy', 'Venus', 'Tsukuba' and 'Cones'. The proposed method of the DILS algorithm matches the image stereo pairs based on SAD cost matching with Left-to-Right Consistency Check (LRCC). During the stereo matching process, the disparity depth maps are obtained using LRCC are shown in Figure 4.21(b). It contains holes and noises that can

be problematic in the image synthesis process. Therefore, the median filter (with the size of 11x11) is used to eliminate the noise and improve the disparity depth map obtained from the LRCC. The improved and smoother disparity depth maps after the filtering process is shown in Figure 4.21(c). The dark regions in the disparity depth maps indicated the unmatched pixels based on the LRCC operations. It is defined as the occlusion and edge-boundary regions. The ability of DILS to distinguish each disparity depth layers not only can be improved the reconstructed novel view images but also are able to remove the occlusion regions through hole-fillings and morphological process.

The histogram distributions indicate the range of the disparity depth layers for the stereo pairs as revealed in Figure 4.22. The number of layers for the disparity depth maps can be identified based on a histogram distribution. The complexity of the images is known with the data distribution in the plot. Closely grouped data shows the image contained high texture information, for example in Figure 4.22(d) in the histogram distribution of ‘Cones’. The layers for the disparity depth are estimated and separated based on the data allocation information in the histogram. Through a layer identification process, the layers were separated into six layers (for ‘Cones’, ‘Teddy’) and five layers (for ‘Tsukuba’, ‘Venus’). The threshold T parameter for the layer identification step (in Section 4.3.2.2) has been set to 20.

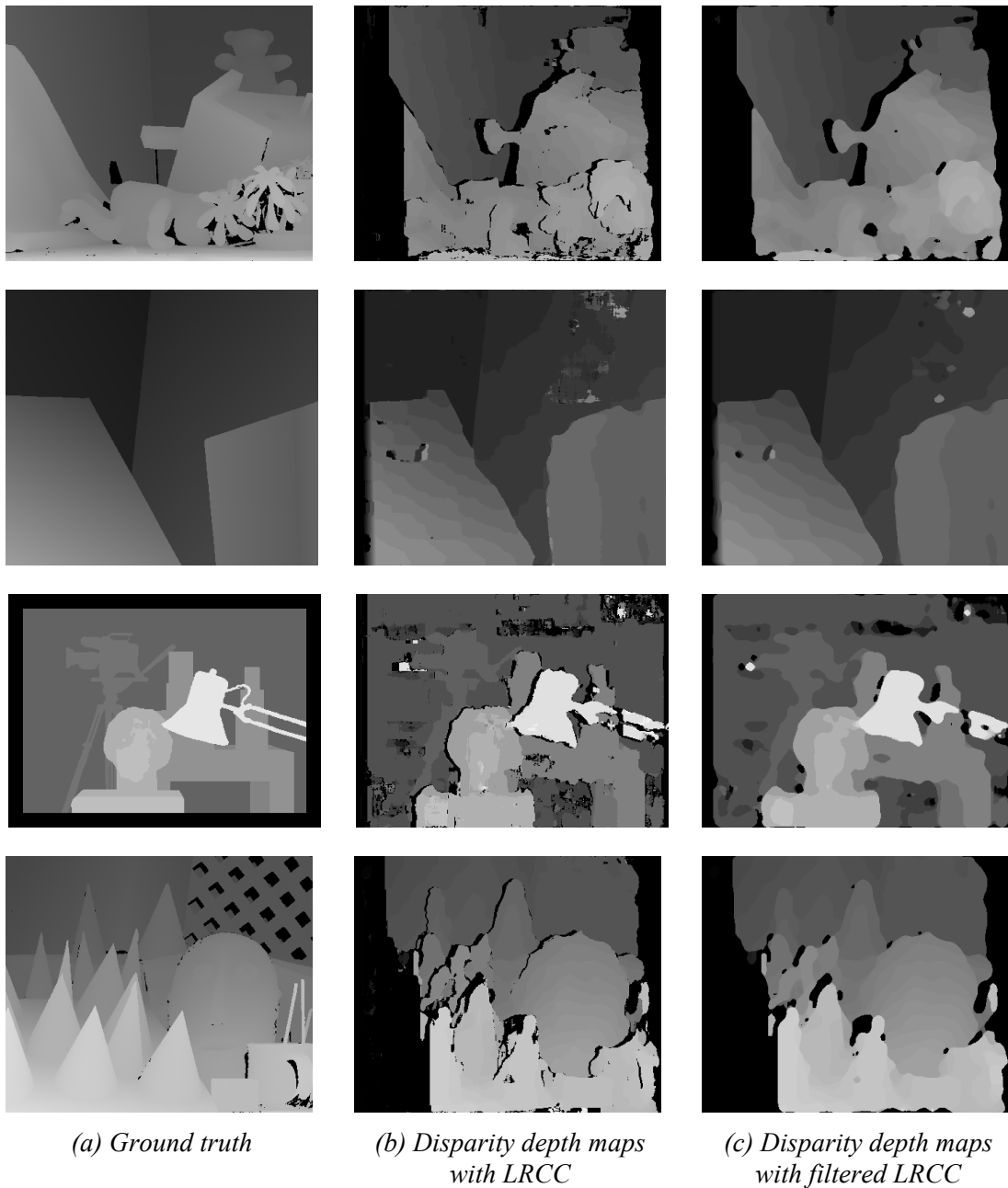


Figure 4.21: Disparity depth maps for (from top to bottom) ‘Teddy’, ‘Venus’, ‘Tsukuba’ and ‘Cones’ image pairs based on the (a) Ground truth, (b) Left-to-right consistency check (LRCC) disparity maps and (c) Filtered LRCC disparity maps using 11x11 median filter

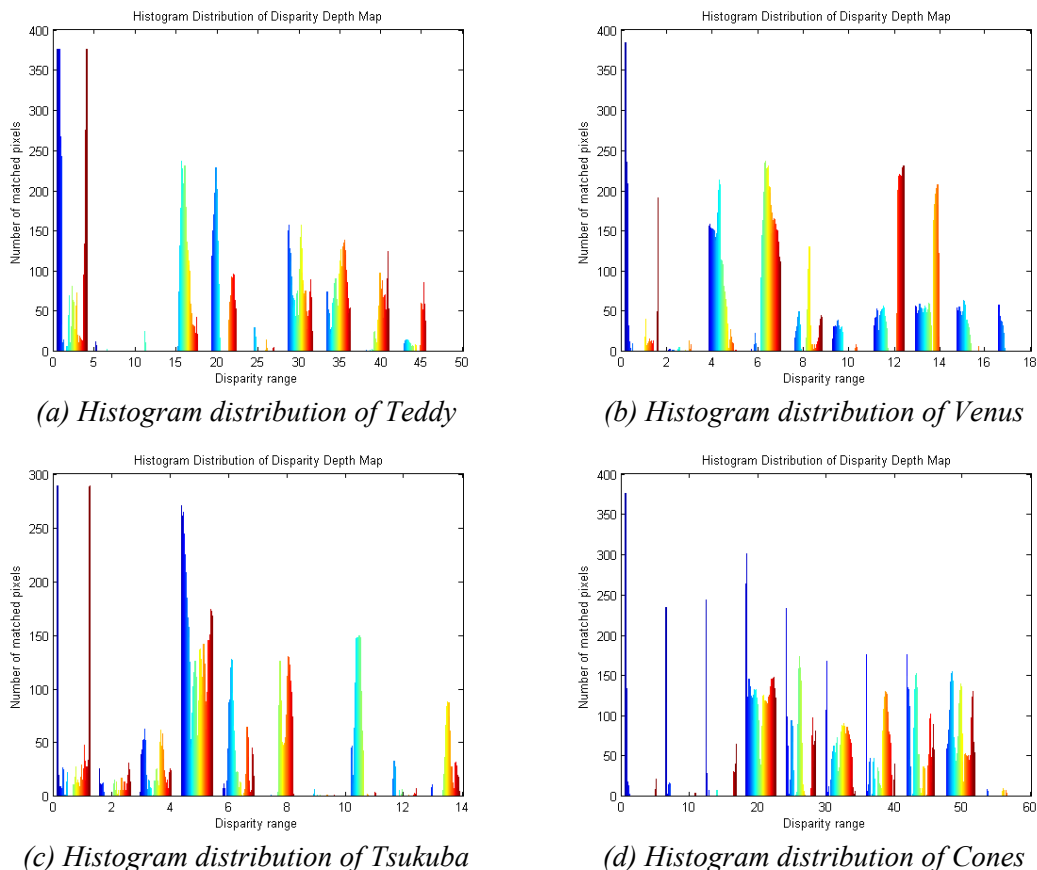


Figure 4.22: Histogram distribution of the disparity depth map using the LRCC.

(a) *Teddy*; (b) *Venus*; (c) *Tsukuba*; (d) *Cones*.

Based on the stereo pair images in the Middlebury datasets, we synthesized the intermediate 3 views (with camera baseline ratio $\beta=0.25, 0.5$ and 0.75) and compared them against the original images provided in the datasets. Figure 4.23 shows the sample original image view of the data sets at camera baseline ratio 0.5. In these data sets, the real cameras are located at views 4 ('Teddy', 'Venus', 'Cones') and 3 ('Tsukuba'). Figure 4.24 shows the results for the synthesized image view of the datasets at camera baseline ratio 0.5 through the conventional Linear Interpolation (LI) and DILS algorithms. It also illustrates the SSIM image map for each respective image view synthesis approaches. In general, both techniques produce good results on the synthesis images although the results based on LI consist with small holes. The quality of the synthesized images was satisfactory enough to provide users with natural free-view images and videos for 3DTV and free-viewpoints applications. However, when we compared them with the original images through SSIM image map, the DILS found to produce fewer errors when compared with LI as indicated in Figure 4.24(b) and (d). The SSIM map of LI consists with more dark regions compared with the SSIM map of DILS.

(a) *Teddy*(b) *Venus*(c) *Tsukuba*(d) *Cones*Figure 4.23: Original image view of the data sets at camera baseline ratio, $\beta=0.5$ (a) *Image synthesis view using LI*(b) *SSIM map for the LI*(c) *Image synthesis view using DILS*(d) *SSIM map for the DILS*Figure 4.24: Image view synthesis of Middlebury datasets at camera baseline ratio, $\beta=0.5$ obtained through (a) Conventional Linear Interpolation (LI) and (c) DILS algorithm; and SSIM map images respectively.

For objective evaluation, we compared the PSNR and SSIM indices results for the reconstructed image LI and DILS as shown in Table 4.2 and Table 4.3 respectively. The standard SSIM parameters used are based on Wang [166] and have been described in Section 3.7.2. The DILS algorithm generally performs well compared to the LI, with an average PSNR of 33.52 dB and SSIM of 0.72. The LI and DILS have the best outcome

for ‘Venus’ dataset. However, the ‘Tsukuba’ gives the worst results in term of PSNR, MSE and SSIM. Most of the errors are due to the high texture background region of the ‘Tsukuba’ image, which cannot be interpolated accurately between the left and right images. Subjectively, the quality of the synthesized image views for all the datasets are satisfactory for free-viewpoint applications. The PSNR and SSIM indices results of the LI and DILS can be plotted in the graph as shown in Figure 4.25. The performance of LI and DILS algorithm is equal for the ‘Venus’ and ‘Tsukuba’ datasets. Nevertheless, the DILS outperforms LI for the ‘Teddy’ and ‘Cones’ datasets.

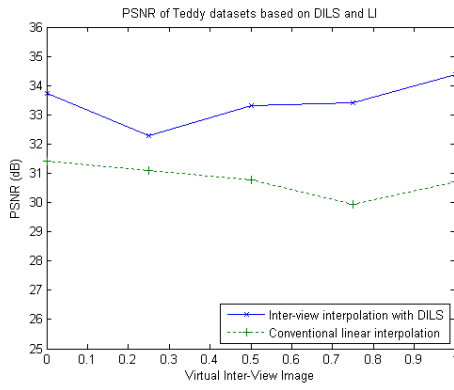
The overall performance of DILS algorithms for the Middlebury datasets is shown in Figure 4.26, indicates that the ‘Venus’ is the best results followed by ‘Cones’, ‘Teddy’ and ‘Tsukuba’. The next evaluation is based on the disparity depth map of AD-Census and fixed window SAD matching aggregation.

Table 4.2: PSNR results of inter-view synthesis images based on conventional Linear Interpolation (LI) and DILS algorithms

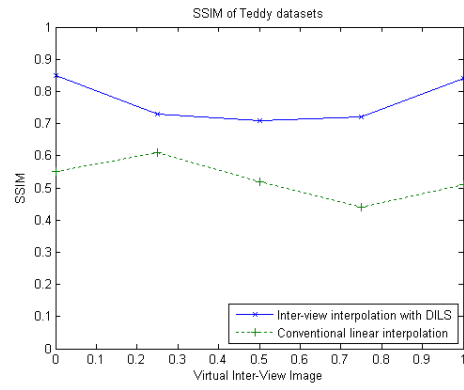
Baseline ratio	Teddy (dB)		Venus (dB)		Tsukuba (dB)		Cones (dB)	
	LI	DILS	LI	DILS	LI	DILS	LI	DILS
0.25	31.1	32.3	36.83	37.41	29.64	30.36	30.44	32.93
0.5	30.76	33.33	35.56	36.39	30.16	31.86	29.38	31.66
0.75	29.94	33.42	34.87	35.8	31.34	33.12	29.84	33.68

Table 4.3: SSIM index results of inter-view synthesis images based on conventional Linear Interpolation (LI) and DILS algorithms

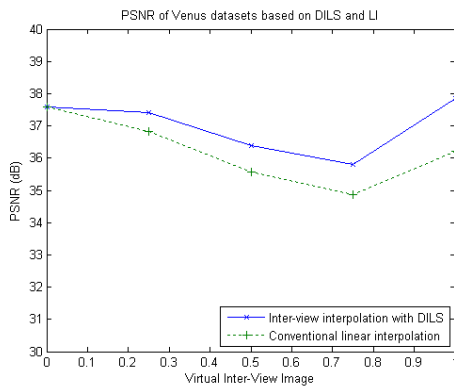
Baseline ratio	Teddy		Venus		Tsukuba		Cones	
	LI	DILS	LI	DILS	LI	DILS	LI	DILS
0.25	0.61	0.73	0.88	0.91	0.41	0.45	0.43	0.72
0.5	0.52	0.71	0.85	0.89	0.45	0.56	0.36	0.68
0.75	0.44	0.72	0.83	0.87	0.57	0.67	0.35	0.71



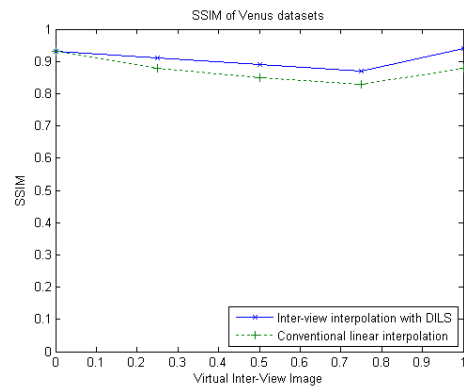
(a) PSNR Teddy



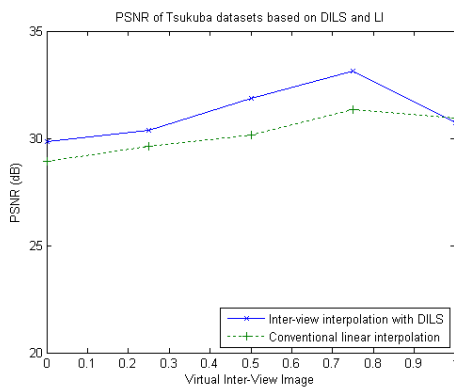
(b) SSIM index of Teddy



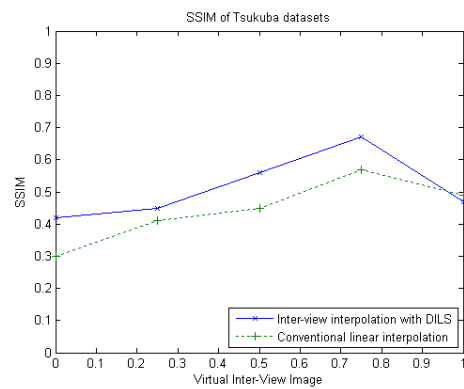
(c) PSNR Venus



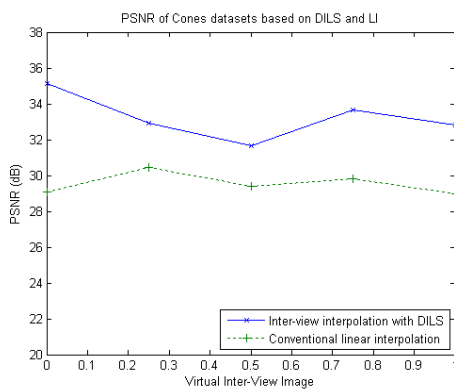
(d) SSIM index of Venus



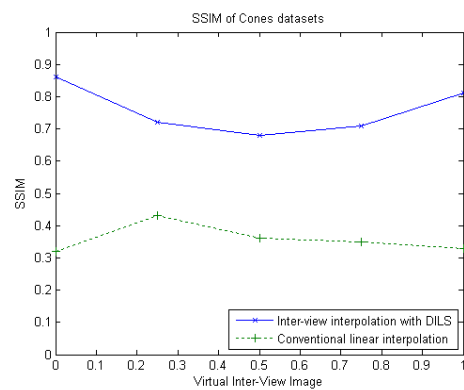
(e) PSNR Tsukuba



(f) SSIM index of Tsukuba



(g) PSNR Cones



(h) SSIM index of Cones

Figure 4.25: PSNR and SSIM index of Middlebury datasets based on LI and DILS algorithms

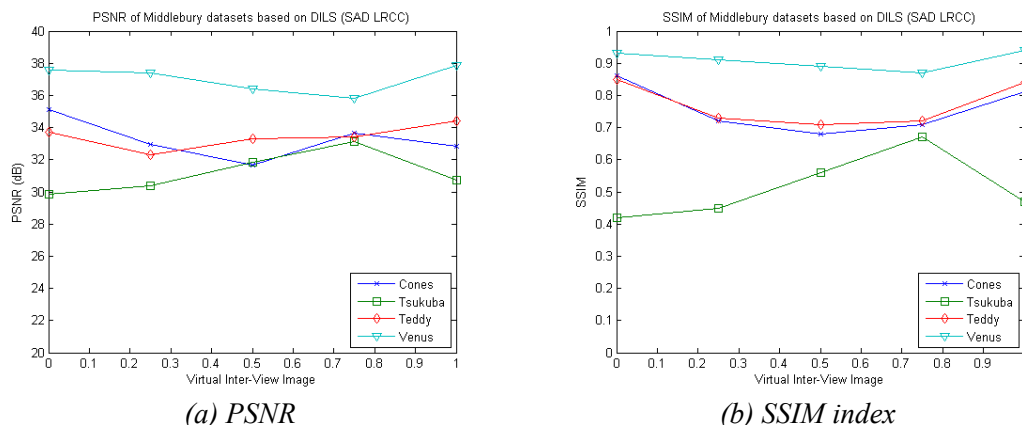


Figure 4.26: PSNR and SSIM index of Middlebury datasets based on DILS algorithm

4.4.2 Performance Evaluation Based on AD-Census and SAD Disparity Depth Map

In order to isolate the performance of the synthesis DILS algorithm, we use the disparity depth map obtained using AD-Census [130] and fixed-window SAD approaches. Each of the disparity depth maps were undergone the DILS process to separate the layers, view interpolation and flatten into a single synthesized view. Figure 4.27 shows the results for the reconstructed image view synthesis at the camera baseline ratio 0.5. The quality of the synthesized images was satisfactory between the AD-Census and SAD. The SSIM map indicates the mismatched pixels between the original images. These errors are not noticeable and not degrading the performance of the image quality for each approach. The error is typical for the data tested and much of the error is around object boundaries where the original disparity map values are not stable or are missing entirely. Even in images where there are many holes, the algorithm produces visually reasonable results due to the disparity and consistency constraints.

We compared the objective evaluation of PSNR and SSIM indices between the AD-Census and SAD disparity depth map results in Table 4.4 and Table 4.5 correspondingly. The performance of DILS is improved by using the AD-Census disparity depth, with an average PSNR of 34.81 dB and SSIM index of 0.78.

As predicted, the image view synthesis reconstruction with the AD-Census outperforms the disparity depth map obtained with the SAD cost matching. It signifies that the DILS performance will improve with better stereo matching algorithms that will give better results as shown in Figure 4.28.

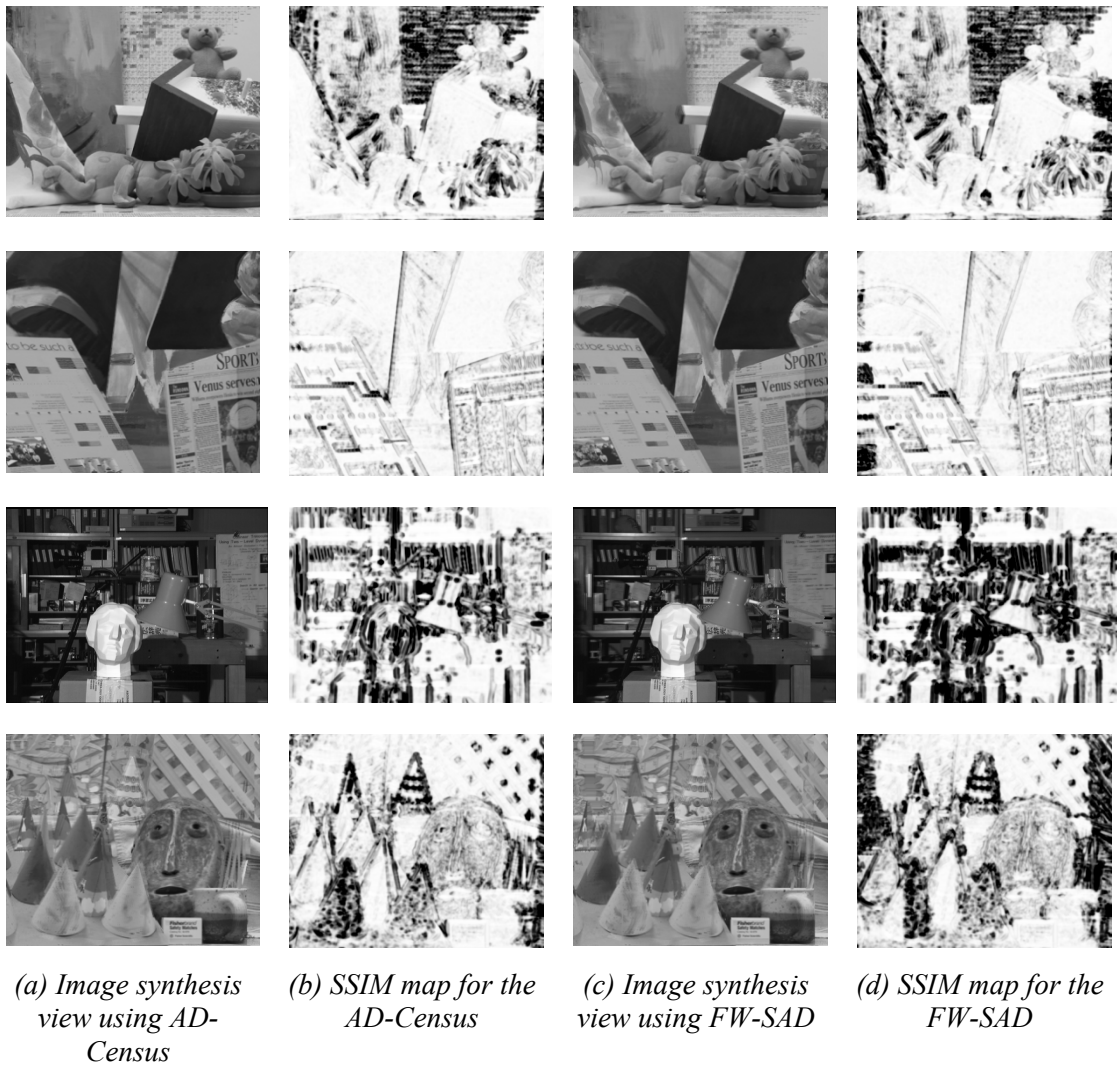


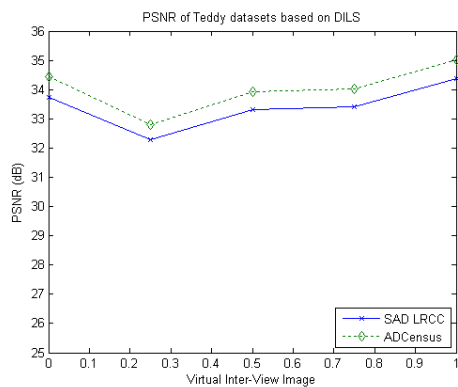
Figure 4.27: Image view synthesis of Middlebury datasets at camera baseline ratio, $\beta=0.5$ through DILS algorithm based on disparity depth maps of (a) AD-Census and (c) Fixed Window (FW) SAD; and SSIM map images respectively.

Table 4.4: PSNR results of inter-view synthesis images based on AD-Census and FW-SAD disparity depth map

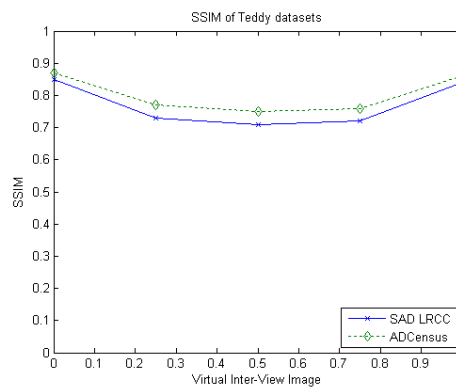
Baseline ratio	Teddy (dB)		Venus (dB)		Tsukuba (dB)		Cones (dB)	
	ADCensus	SAD	ADCensus	SAD	ADCensus	SAD	ADCensus	SAD
0.25	32.8	32.3	38.1	37.41	32.49	30.36	34.3	32.93
0.5	33.94	33.33	36.93	36.39	34.85	31.86	32.47	31.66
0.75	34.02	33.42	36.24	35.8	36.93	33.12	34.67	33.68

Table 4.5: SSIM index results of inter-view synthesis images based on AD-Census and FW-SAD disparity depth map

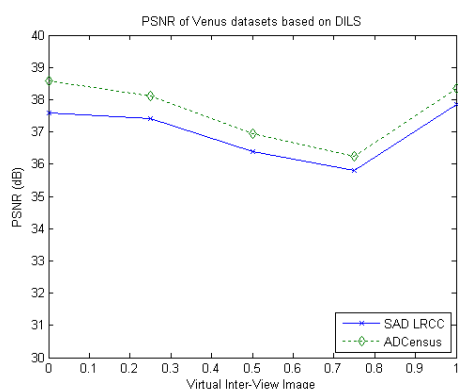
Baseline ratio	Teddy		Venus		Tsukuba		Cones	
	ADCensus	SAD	ADCensus	SAD	ADCensus	SAD	ADCensus	SAD
0.25	0.77	0.73	0.93	0.91	0.55	0.45	0.81	0.72
0.5	0.75	0.71	0.91	0.89	0.68	0.56	0.75	0.68
0.75	0.76	0.72	0.89	0.87	0.8	0.67	0.78	0.71



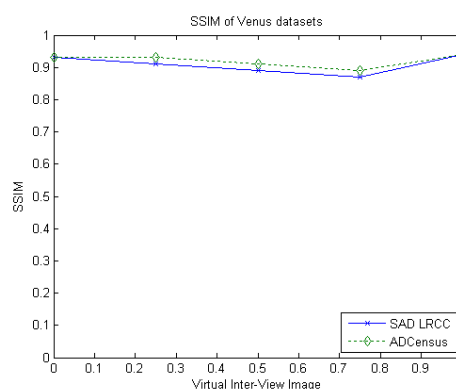
(a) PSNR Teddy



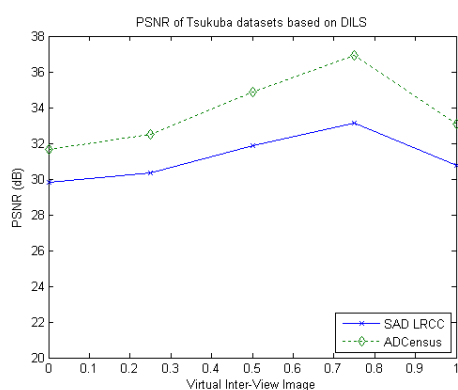
(b) SSIM index of Teddy



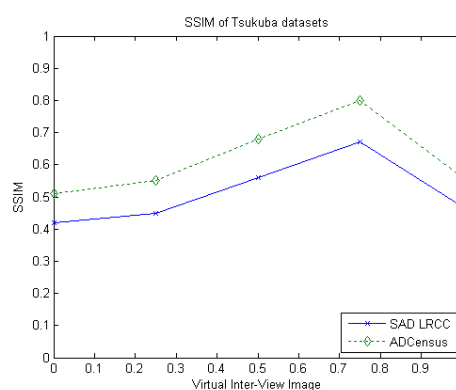
(c) PSNR Venus



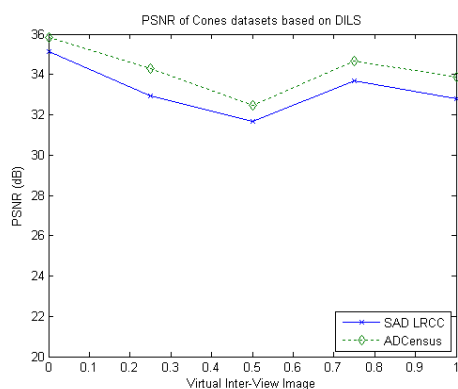
(d) SSIM index of Venus



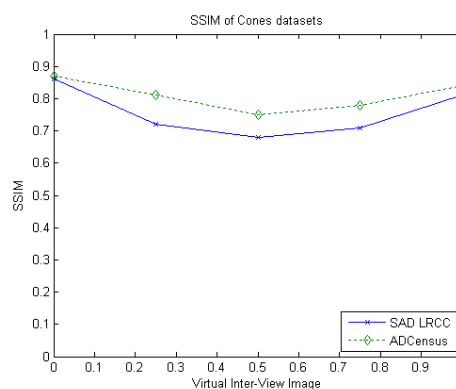
(e) PSNR Tsukuba



(f) SSIM index of Tsukuba



(g) PSNR Cones



(h) SSIM index of Cones

Figure 4.28: PSNR and SSIM index of Middlebury datasets through DILS based on based on AD-Census and FW-SAD disparity depth map

4.5 Conclusion

Image view synthesis is a practical solution for generating content for autostereoscopic multi-view displays and free-viewpoint video applications. In addition, synthesizing views is essential for certain applications and future displays that will incorporate more views created. Image view from a stereo pair also has benefits when considering data transmission of multi-view imagery. In this chapter we have presented a novel algorithm for Depth Image Layers Separation (DILS) to synthesize novel inter-view images based on disparity map layers representation. The work presented exploits inter-view correlation to generate intermediate view synthesis image that locates in the virtual viewpoint between source image viewpoints. DILS features a new paradigm that is not just a method to select interesting locations in the image based on the depth analysis. It is also a new image representation that allows the descriptions of the objects or parts of the image without the need of segmentation and identification. The image view synthesis can reduce the complexity of multi-camera array configuration for 3D imagery and free-viewpoint applications.

The performance of the algorithm was tested on the Middlebury Database yielding high PSNR and SSIM index values. The quality of synthesized multi-view images is very impressive and satisfactory for free-viewpoint applications. The proposed method gives comparable performance to the conventional inter-view interpolation. In the experiments, it was demonstrated that it is possible to efficiently synthesize realistic new views even from inaccurate depth information through the DILS algorithm. DILS can be used with simple or sophisticated stereo matching techniques to synthesize better quality inter-view images.

There are many possible extensions to this work. The layers of the disparity depth map from the DILS can be used to refine the disparity depth map from the stereo matching algorithm. In the next Chapter 5, a new algorithm to refine the disparity map based on the disparity layer refinement using the DILS will be developed. The DILS is expanded to generate dense multi-view images of multi-camera arrays configuration for free viewpoint video and light-field imaging applications in Chapter 6.

Chapter 5

Depth Layer Refinement (DLR) Algorithm for Disparity Depth Map

5.1 Introduction

In disparity refinement step, raw disparity maps were computed by correspondence matching algorithms containing outliers that must be identified and corrected. Several approaches aimed at improving the raw disparity maps computed by stereo correspondence algorithms such as sub-pixel interpolation [23], image filtering techniques, Bidirectional Matching [24] and Single Matching Phase [25]. Even though the proposed algorithms can provide accurate disparity depth map, they tend to suffer from significant complexity requirements, making them less suitable for real-time applications.

In this chapter, a new algorithm is presented that improves the raw disparity maps in the disparity refinement stage with low complexity. The proposed algorithm uses a simple stereo matching correspondence algorithm with a basic similarity metric of SAD. The similarity metric finds the pixel points between the left and right images under the Fixed Window (FW) searching process. With this approach the raw disparity depth map obtained is not smooth and contains errors, particularly with the depth discontinuities, and it is unable to detect the uniform areas and repetitive patterns. The proposed algorithm uses the disparity depth map from the stereo matching algorithm as initial disparity depth output. The initial disparity depth will be used to identify the layers of disparity depth map since the depth consists of a range of disparities. This approach is adapted from the Depth Image Layers Separation (DILS) algorithm described in Section 4.5 that separates the layers of depth based on disparity range. In general, each particular disparity depth map is distributed along the disparity range and can be divided into several segments, which are known as layers. Instead of using each layer to

synthesize inter-view images in the DILS, the layer will be mapped to segment reference image and to refine the disparity depth map. This method is defined as the Depth Layer Refinement (DLR).

This chapter is organized in six sections. Section 5.2 provides an overview of the system design and also outlines the main features of the model that consist of two main modules: stereo matching algorithm and disparity refinement modules. Section 5.3 covers the proposed algorithm for the disparity refinement by adapting the Depth Image Layers Separation (DILS) algorithm. In section 5.4, the performance evaluation for the disparity depth map is presented. The results and performance are discussed in Section 5.5, which compares the proposed algorithm with the state-of-the-art stereo matching algorithm in the Middlebury Ranking Stereo Page. Section 5.6 presents some concluding remarks.

5.2 Overview of System Design

The proposed DLR system design, shown in Figure 5.1, consists of two stages: a stereo matching engine and a disparity refinement module. The first stage of the DLR system comprises three main components: matching cost computation, cost aggregation and disparity computation/optimization. The matching cost computation step can be divided into two main categories; the pixel-based matching costs and the area-based matching costs. Some similarity metrics used in the matching are the Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC). The classification and evaluation of cost aggregation strategies for stereo correspondences [108] depends on the position, shape and weights.

In this system design, the raw disparity depth map is obtained from the stereo matching based on left-to-right matching using a block-based fixed window similarity metric. In this case we are using the SAD metric that has been proven to be a trade-off between reliability and computational cost [97]. However, other similarity metrics can be used as well. Window-based methods implicitly make the assumption of continuity by assuming constant disparity for all pixels inside the matching window. This assumption is broken at depth boundaries where occluded regions lead to erroneous matches, resulting in a familiar foreground flattening effect.

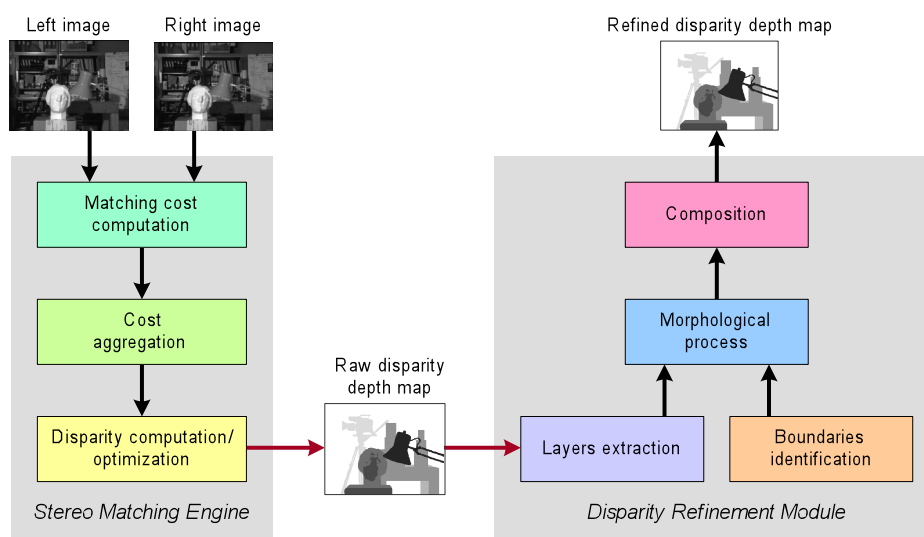


Figure 5.1: Overview DLR system

Generally, the choice of an appropriate window size is a crucial decision. Small windows do not capture enough intensity variation to give correct results in less-textured regions. On the other hand, large windows tend to blur the depth boundaries and do not capture well small details and thin objects. This motivates the use of adaptive windows [87], shiftable windows [32], multiple windows [116], variable windows [109], bilateral filtering [123] and adaptive weights [99]. The new algorithms adopt some of these approaches to improve the disparity depth map. In spite of its limitation, SAD with a Fixed Window (FW) is the most frequently used algorithm for real time applications due to having easy implementation, being fast and having limited memory requirements. Therefore, the fixed window similarity stereo matching technique is adequate when obtaining the estimated depth map. This configuration can be adapted for computation optimization in real-time hardware implementation [25].

The disparity computation or optimization step aims at finding the best disparity assignment that minimizes a cost function over the whole stereo pair. The relevant approaches are with the Graph Cuts [79], Belief Propagation [80, 81, 113-115] and Dynamic Programming [82-84]. The most common and effective method is a simple Winner-Takes-All (WTA) minimum or maximum search over all possible disparity levels. The matching can be done from right to left or vice versa (Bidirectional Matching), so occlusions and uncertain matches can be filtered out with a Left-to-Right Consistency Check (LRCC). This means only disparities with the same value (within a certain range) for both directions are accepted. In this case, only a single matching is needed for the DLR algorithm. The main reason of this is to use the depth layer and

edge maps to remove the uncertain matches.

In the second stage, the disparity depth map is separated into a number of layers based on the disparity range of the stereo pair. The disparity depth map can be improved with some techniques such as sub-pixel interpolation [23], image filtering techniques, Bidirectional Matching [24] and Single Matching Phase [25]. Even though these algorithms provide exceptional accurate disparity depth map, extra iterations are also required to compute the mismatch between the uncertain pixels in a Bidirectional Matching and computational complexity within some of the proposed techniques for real-time and practical implementation. The proposed disparity refinement is developed through the layer extraction and separation process was implemented using the DILS algorithm. A new approach to refine the disparity image map is presented at this stage with boundaries identification, morphological and composition process, which are the DLR components. The layers are mapped and adaptively fused with a reference image to identify the edges, borders, depth discontinuities, uniform areas and repetitive patterns. The description of the disparity refinement module will be given in the next section.

5.3 Disparity Layer Refinement

This section describes the proposed Disparity Layer Refinement (DLR) algorithm. The overall algorithm for DLR is based on the DILS algorithm and can be divided into four major steps that are summarized in Figure 5.2. These steps include the stereo matching and layers extraction (Part 1), boundaries and edges identification (Part 2), morphological process (Part 3) and, lastly, the layer composition stage (Part 4). The input to the matching engine comprises two stereo images in epipolar geometry.

The first processing step is the stereo matching and layer extraction that will be described in Section 5.3.1. We calculate an initial disparity map using a fixed window-based correlation technique. The DILS algorithm separates the disparity depth map into several numbers of layers depending on the complexity of the image pairs. The disparity levels and layers can be determined using a histogram distribution, which were described in the DILS algorithm. The number of layers is indexed by i , where i can be between 1 and the maximum D . The stereo matching and layer extraction were discussed in Section 4.3.1 and Section 4.3.2, respectively. The main difference in this

stage is that the stereo matching is done only for left-to-right matching and the left-right consistency will not be checked.

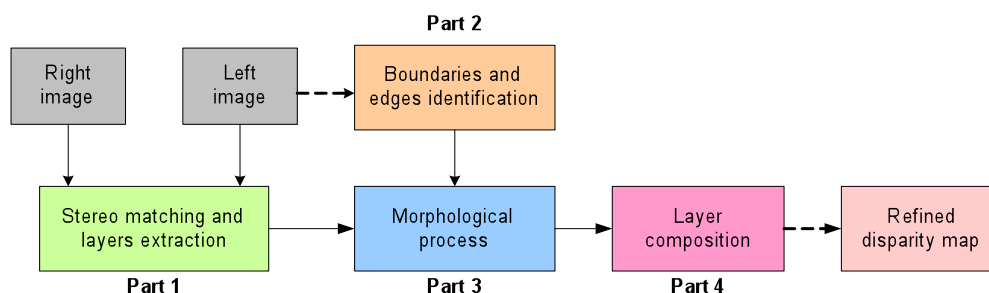


Figure 5.2: Block diagram of the proposed algorithm on disparity refinement based on DILS

Since the discontinuities in the disparity map are usually reflected by discontinuities in the edge and colour information, the borders of the segmented regions can be considered as a set of candidates for the boundaries of the disparity layers that we aim to compute and refine. For layers identification, the left image is selected as the reference image. This image undergoes the edge detection and colour segmentation process to obtain the edge and borders in the reference image. From this stage, the new edge map is obtained as the reference image is masked to create the edge boundaries of the layers. This process explained in more detail in Section 5.3.2.

In Section 5.3.3, we create an initial representation for each extracted layer by separating the disparity depth that is obtained from the DILS in Part 1. The computed layer in Part 3 is obtained by fusing the disparity layer and edge map from Part 1 and 2 respectively. The mapping process of the layer and edge map are used to create a new binary image mask layer that will be processed with the morphological operation. Each disparity depth map is refined individually through layer separation and mapping process. As described in the previous section, the disparity depth map is generated using the left-to-right matching algorithm. The raw disparity depth map consists of false matches and does not address any occlusions. Through individual layer refinement process, the noise and false matches can be removed without degrading the discontinuities in the edge map.

The last block in the DLR module is the layer composition, which is explained in Section 5.3.4. During this stage, the final disparity map is composed by the extracted refined layers from Parts 2 and 3. The top layer is the closest object to the camera view. The layers under this layer are the layers identified by the disparity range in the DILS algorithm. All the layers are then combined in a single disparity depth map. The cracks

and holes are corrected with the hole filling techniques, described in Section 4.3.2.9.

5.3.1 Stereo Matching and Layers Extraction

The stereo matching and layers extraction are based on the left-to-right image matching. For this implementation, the matching cost computation uses the basic similarity metric, SAD, that is a conventional approach for many stereo matching algorithms. As described earlier, any similarity metric and approaches can be selected to enhance the accuracy and reliability of the disparity map obtained from this process. The main idea for this implementation is to show that even by using a basic similarity metric, the disparity map can be improved significantly through the DLR algorithm. The matching process uses Equation (4.5), which was defined in the previous Chapter.

The cost aggregation is done by summing matching costs over fixed square windows searching with constant disparity. The accuracy of the depth map can be increased using a larger window size. However, there is a trade-off between the accuracy and the depth discontinuities of the objects. Many methods have been proposed in the literature to improve the disparity map with efficient and robust approaches for the cost aggregation. As observed by Kanade and Okutomi [87], the correlation window which covers a region with non-constant disparity does not perform well and the error in the depth discontinuities grows with the window size. Reducing the window size makes the computed disparity more noise-sensitive. To overcome this problem, Kanade proposed an adaptive window, which can statistically select each pixel that minimizes the uncertainty of the disparity estimation. This approach has been improved by Fusiello [173] with the symmetric multi-window to provide efficient and robust disparity estimation in the presence of occlusions. Although the presented cost aggregations in [87, 125, 173] perform very well by improving the disparity map, the fixed square window is sufficient for basic area-based stereo matching. This provides faster implementation and low complexity. Furthermore, the configuration of the fixed square window can be adapted for computation optimization in the hardware parallel implementation that has been proposed by Stefano [25].

The raw disparity map can be visualized by selecting the minimal aggregated value at each pixel. For applications such as robotic navigation or people tracking, the disparity map obtained from this stage may be perfectly adequate. However for image-based rendering, the raw disparity maps lead to errors and unappealing view synthesis results.

To enhance the performance for the DILS algorithm, the raw disparity map is filtered with a median filter, which cleans up mismatches, holes and noises. In our implementation, we are not performing bidirectional matching (LRCC) to calculate the occlusion since we want to measure the performance of the DILS and DLR algorithm components. Within this stage, we obtained two main results that are the raw disparity depth map and the layers of the disparity depth (from the DILS algorithm).

5.3.2 Boundaries and Edges Identification

After the stereo matching and layer extraction processes, the boundaries and edges of the reference (left) image are identified. By assuming that for regions of homogeneous colour, the disparity varies smoothly and the depth discontinuities coincide with the boundaries of those regions, which is true for most natural scenes as described by Bleyer [118]. This assumption is incorporated, by applying colour segmentation, to the reference image and, by using a disparity layer, to represent the disparity inside the new layer segments. In addition to the colour segmentation, the reference image are derived the edge boundaries using the edge detection algorithms. In theory, any algorithm able to identify sharp edges and discontinuities in the edge detection can be used for the proposed boundaries and edge identification stage. Also, any algorithm that divides the reference image into regions of homogeneous colour can be used for this stage. In our implementation, we used the mean-shift colour segmentation algorithm proposed by Comaniciu [110] and incorporate edge information by using the Canny edge detection algorithm [174].

The mean shift analysis approach is essentially defined as a gradient ascent search for maxima in a density function defined over a high dimensional feature space. The feature space includes a combination of the spatial coordinates and all its associated attributes that are considered during the analysis. The main advantage of the mean-shift approach is based on the fact that edge information is incorporated as well [81]. The Edge Detection and Image Segmentation (EDISON) system [175], developed by Rutgers University, provides a complete toolbox for discontinuity preserving filtering, colour segmentation and edge detection. This EDISON has also been used in our system.

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity, which characterize boundaries of objects in a scene. Classical methods of edge detection involve

convolving the image with an operator of a 2-D filter, which is constructed, to be sensitive to large gradients in the image while returning values of zero in uniform regions. There are an extremely large number of edge detection operators available. Each of them is designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include orientation, noise environment and structure. In edge orientation, the geometry of the operator determines a directional characteristic in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges.

Edge detection is difficult in noisy images, since both the noise and the edges contain high-frequency components. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges. In the edge structure, not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator needs to be chosen so as to be responsive to such gradual changes in these cases.

The Canny edge detection algorithm is known as the optimal edge detector. It is important that edges occurring in images should not be missed and that there are no responses to non-edges. The second criterion is that the edge points should be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge has to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the Canny edge detector firstly smooths the image to minimise the noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (a non-edge). If the magnitude is above the high threshold, it is identified as an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the threshold [174]. Therefore, the Canny edge detection is used along with the colour

mean-shift segmentation.

The algorithm outlines Parts 1 and 2 are summarized in Figure 5.3, where the results of stereo matching and layer extraction in Part 1 and the edge map image obtained in Part 2 are used in Part 3, which is the morphological process. The binary images from edge detection and colour image segmentation steps combined to create the edge map image using AND operator. The new segmented and edge map image is defined as I_S . The segmented image I_S will be mapped and fused together with the layer i . The fusion process of the disparity depth layer and edge map image is described in Section 5.3.3.

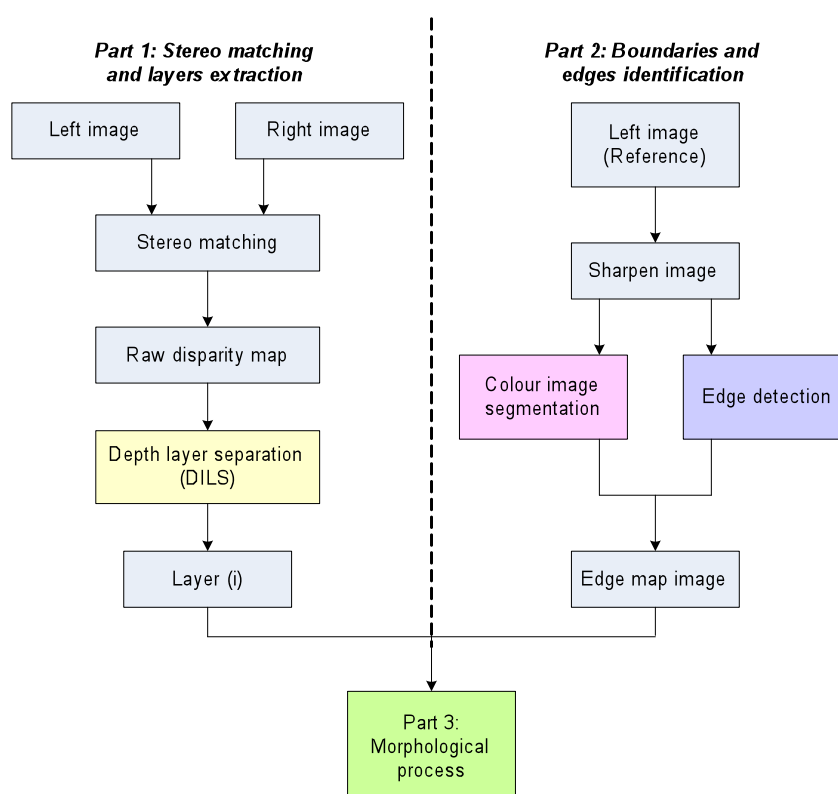


Figure 5.3: Part 1 and 2 block diagram for the DLR that consist: a) stereo matching and layers extraction, and b) boundaries and edges identification

Figure 5.4(c) shows the edge map constructed by the Canny edge detection. In this example, the image of ‘Teddy’ has been used, which contained high textured region. The original ‘Teddy’ image has been shown in Figure 5.4(a) and 5.4(b). The lines and edges in the result provide the boundary borders to separate the inner and outer region in the morphological process in Section 5.3.3. The reference left image has been processed by the colour image segmentation as illustrated in Figure 5.4(d). The additional information in the colour image segmentation enhances the boundaries and edges identification process. The result of edge detection map and colour segmentation map are combined with AND operator to create the new edge map image (Figure

5.4(e)), that can be translated in the binary image for the morphological process in Part 3. The sample of raw disparity depth map as shown in Figure 5.4(f) extracted into several layers based on disparity range explained in the DILS algorithm.

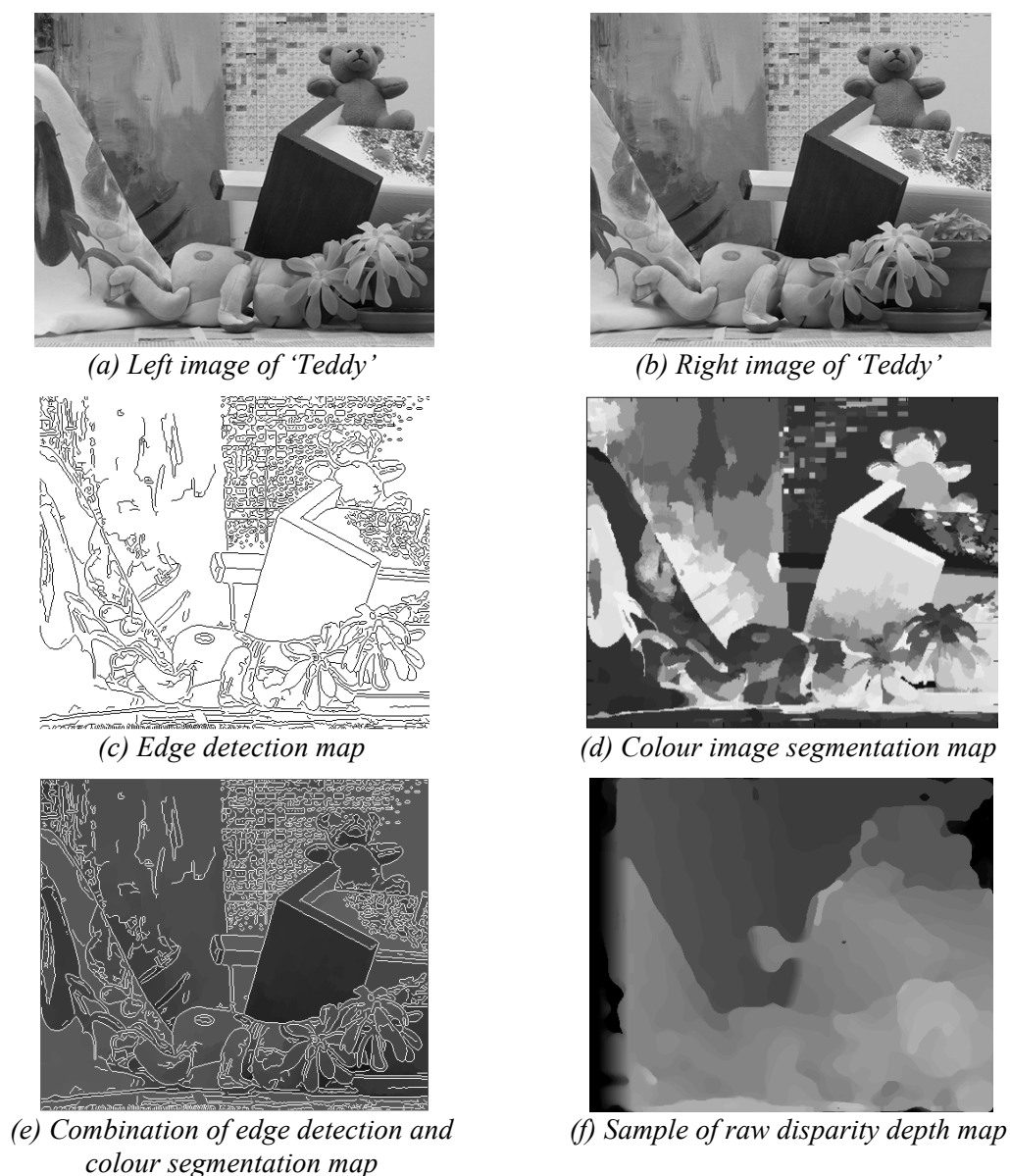


Figure 5.4: Boundaries and edge detection example

5.3.3 Morphological Process

The disparity depth map can be refined by a median filter approach, where the outliers and noise can be removed. However, some of the noises cannot be removed automatically without affecting the whole portion of the disparity depth map, which is obtained from the stereo matching algorithms. With disparity layer separation, particular noise can easily be removed while maintaining the quality in some of the

disparity layers. The accuracy of the disparity depth map can be enhanced with each layer is being processed with a morphological process. The unwanted pixels in the object mask layer can be removed using erosion and dilation processes.

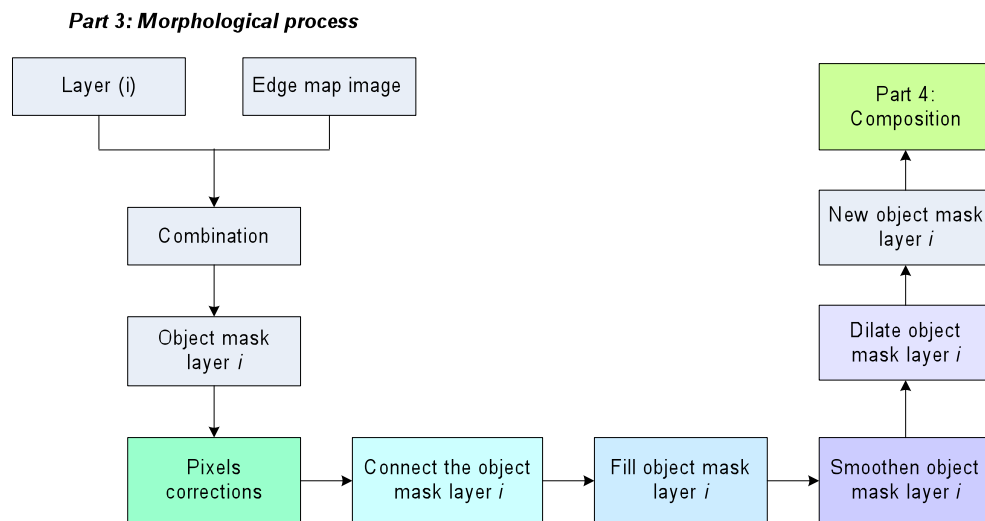
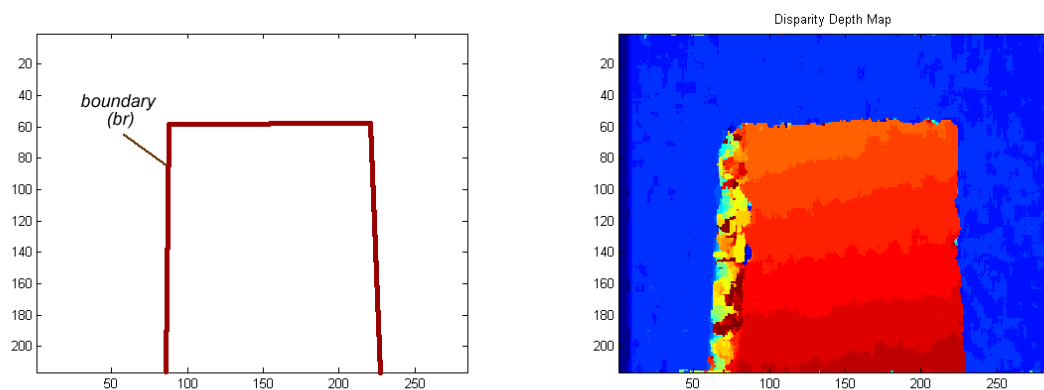


Figure 5.5: Part 3 block of the DLR algorithm, morphological process

Figure 5.5 shows the block diagram of the Part 3 of the DLR algorithm, which takes as input the edge map image and the layer i (separated by the DILS algorithm). The combination of the input created a binary object map with AND operator that holds the boundary of the edge layers. Each layer will be mapped on the same segmented edge map image I_S . Any edges and borders of the objects mapped and crossed with the same region on the layer i remain in the image, while the remaining is removed. The new-segmented image is now fused with the same region of layer i . The edge on the segmented image will create a cross path along the layer i . The cross path is defined by the new boundary notated as br and illustrated in Figure 5.6(a), with the disparity depth map in Figure 5.6(b).



(a) Cross path defined by the boundary

(b) Sample of disparity depth map

Figure 5.6: Sample of boundary path for layer i and the disparity depth map

The pixels in the object map are corrected by removing unwanted pixels. The technique used in this block is based on erosion process combined with the algorithm proposed by Fergusson [176]. After that, the object map pixel of the layer is connected with a convex hull, which creates the closed-loop boundary region. The boundary region will be filled to produce the binary object map image.

During this stage, two regions of the disparity layer i can be distinguished based on the boundary created, which are the inner region and the outer region. The inner region is the disparity depth map that is contained inside the boundary. Any zero pixels on this region will be filled with the same value of layer i . The inner region is dilated until the boundary that is set as the threshold is reached. Meanwhile, the outer region is for the disparity depth map that is beyond the boundary edge of the segmented image. Any outer region of the disparity map will be eliminated. As a result, the new disparity layer i is created adaptively based on the boundary of the object from the segmented reference image. This approach addresses the disparity depth discontinuities problems and is able to detect the uniform areas and repetitive patterns on the stereo pairs. The process is illustrated in Figure 5.7. The sample of output from the layer i and edge map image combination is shown in Figure 5.8(a). The object mask layer i is processed in the morphological stage that finally produces the new binary object mask layer i as illustrated in Figure 5.8(b). This process is iterated for all the layers of the disparity depth map before the layers can be composed as a single refined disparity depth map.

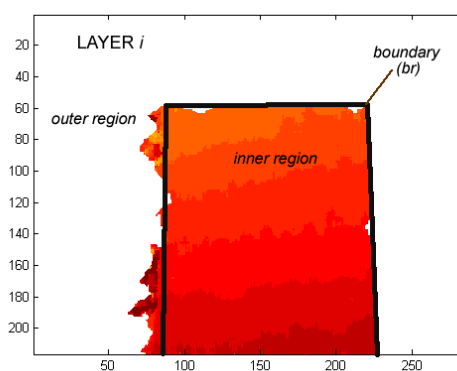


Figure 5.7: Mapping and diffusing for layer i with the border set by the segmented reference image

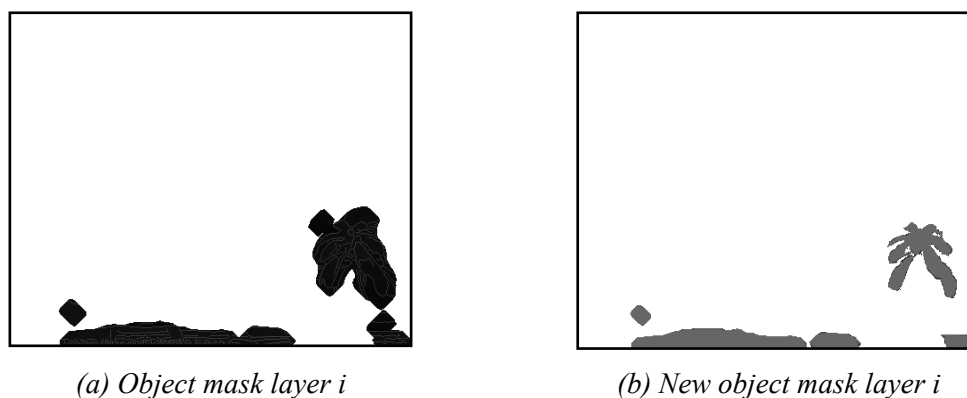


Figure 5.8: Layer extraction with edge map image. (a) Raw object mask layer i ;
 (b) Refined layer i through morphological process

5.3.4 Layers Composition

Each layer of the disparity depth map undergoes the same process of mapping and fusing (diffusion) with the same reference segmented image. After all of the layers have been processed, the collection of layered disparity depth images merges into a new single refined disparity depth image. The final disparity map is combined all the layers according to the rank layer. The arrangements of the layers are as follows: the top layer is closest to the camera view (which is the highest number of layer i), then followed by the next layers according to layer i level values. Basically, the layers composition is similar to image layers view synthesis in DILS algorithm, where all the layers are flattened into a single layer. The process of the layer composition of the DLR algorithm can be summarized in Figure 5.9.

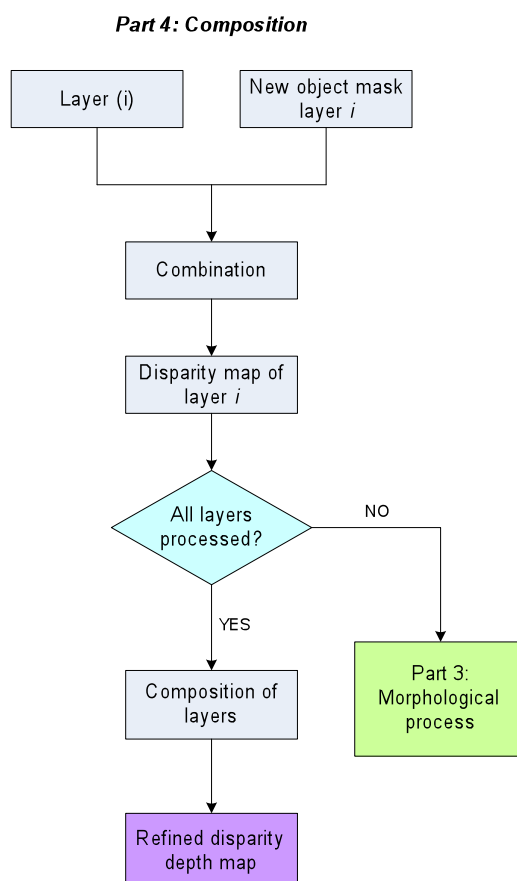


Figure 5.9: Part 4 block of DLR algorithm, layers composition

5.4 Performance Evaluation

In this section, the quality metrics are described for evaluating the performance of the stereo correspondence algorithms based on the image data sets and ground truths according to the evaluation platform by Scharstein and Szelinski [22].

5.4.1 Quality Metric

In order to evaluate the performance of a stereo algorithm, a quantitative way is needed to estimate the quality of the computed correspondences. Two general approaches to this are to compute error statistics with respect to some ground truth data and to evaluate the synthetic images obtained by the disparity depth map. Two quality measures based on known ground truth data provided by the Middlebury Vision Page are RMS (Root-Mean-Squared) error and percentage of bad matching pixels. RMS error is measured in disparity units between the computed disparity map $d_c(x, y)$ and the ground truth map $d_T(x, y)$ [22, 85]:

$$RMS = \sqrt{\frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_T(x,y)|^2} \quad (5.1)$$

where N is the total number of pixels. The percentage of Bad Matching Pixels, BMP is given by [22]:

$$BMP = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d) \quad (5.2)$$

where δ_d is a disparity error tolerance. For the experiments and evaluations, the disparity error tolerance, δ_d is set to 1.0. In addition, to compute these statistics over the whole image, two different kinds of regions are evaluated which are the non-occluded and depth discontinuities regions.

5.5 Results and Discussion

The results for the depth refinement algorithms are evaluated based on the performance evaluation with these two approaches. The first performance is tested based on different similarity metric for the cost aggregation. Although the selected similarity metric is SAD, the comparison with different approaches will also be shown. This includes the selection of window size for the correspondence matching. The second performance is based on the Middlebury Stereo Evaluation [177]. The evaluation platform provides stereo image datasets consisting of the stereo image pair and the ground truth image. The proposed algorithm evaluated by using the Middlebury datasets and is compared with results with many others through online. The online page is constantly updated and provides some common benchmark datasets and evaluation systems, where we can examine and analysis the proposed algorithm objectively and universally by using standard parameters.

5.5.1 Performance Evaluation Based on Different Similarity Metric

This section gives a detailed evaluation of the proposed algorithm in terms of results, quality and processing time. The DLR algorithm can be used with any stereo matching algorithm since it was developed to refine the raw disparity map images (in the post-processing stage). For this case, the evaluation has been made with Map (284x216 pixels) and Tsukuba (384x288 pixels) image with different similarity metric including

SAD, SSD, SHD and NCC. The parameter of the stereo pair images set to 9x9 window size with maximum disparity 30 (Map) and 16 (Tsukuba).

The results of stereo matching for Tsukuba image based on different similarity metrics are shown in Figure 5.10. The raw disparities based on the block-based window searching contained errors with unmatched pixels especially with the similarity metric SHD. For this sample, the disparity depth map has been mapped with colour to show the hotter the colour, the closer the object is to the camera. In this case, the red colour (the lamp) is the closest object. The output of the disparity depth maps can be improved with the post-processing stage by using a median filter to smooth the result. The bidirectional matching can be used to eliminate the unmatched pixels, which can produce accurate disparity depth maps. Based on the results as shown in Figure 5.10, the disparity depth maps produced using SAD and SSD similarity metrics are better and SHD is the worst. The errors and unmatched pixels in the disparity depth map generated by SHD are obvious in comparison with the SAD, SSD and NCC similarity metrics.

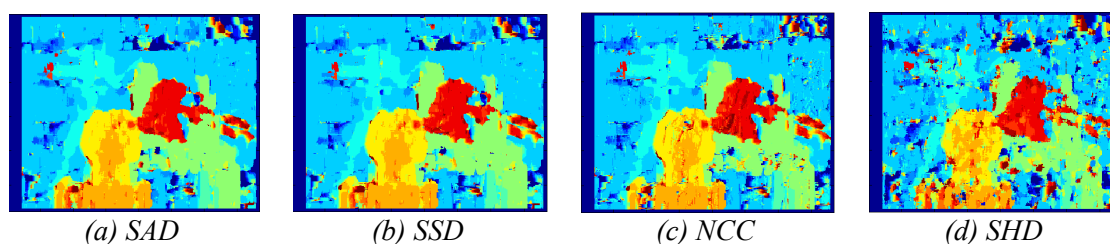


Figure 5.10: Results of stereo matching based on different similarity metric

The size of window for the block-based matching affects the performance of the matching algorithm as indicated in Figure 5.11(a), where the RMS errors are reduced accordingly when the window size is increased for the all-pixels evaluation. The non-occluded pixel errors are not affected with different window size as shown in Figure 5.11(b). The errors are significantly reduced when the disparity depth map is filtered (in this case by using 11x11 median filter).

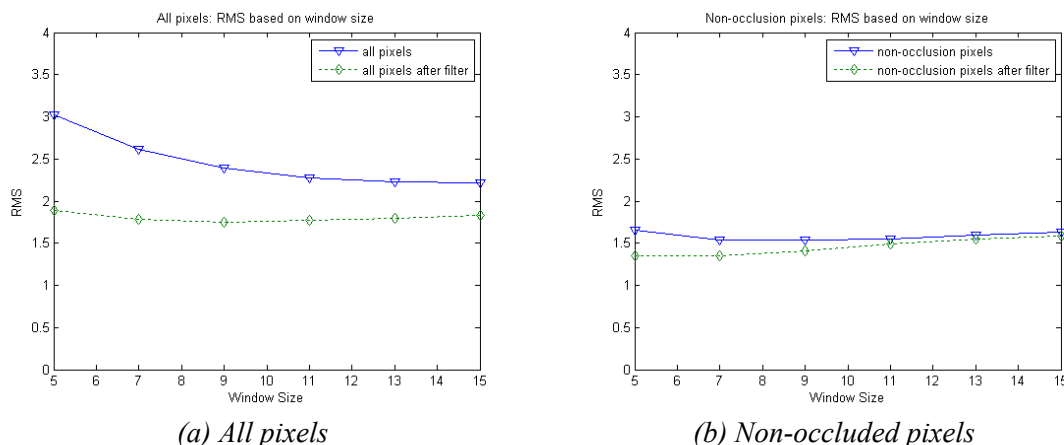


Figure 5.11: RMS error based on window size for all pixels and non-occluded pixels

The performance of different similarity metrics is presented in Table 5.1 for Tsukuba and Map images. The table shows the statistics aimed at assessing the capability of the similarity metrics in term of processing time and RMS. The time is calculated in seconds for the processing time and the RMS in term of pixels. The stereo matching evaluated with Intel Quad CPU of 3.0 GHz, 3.25 GB of RAM. The comparison of similarity metrics with Bidirectional Matching (BM) is also included. It is worth noticing that with the Map stereo pair, the similarity metrics of SAD, SSD and NCC perform similarly and pretty well, with slightly better RMS yielded by BM. The BM shows the capability to deal with occlusions and uncorrected disparities. The similarity metric performs better with Map image pairs compared to the Tsukuba due to the complex objects at different depths generating several occlusions, as well as poorly textured regions in the background. Moreover, this stereo pair contains some specular regions (such as the face of statue and some regions of the lamp) that are quite difficult to deal with in the stereo matching process.

Based on this evaluation, it shows that the similarity metric using SAD is satisfactory. Besides the simplicity, reliability and low computational cost, SAD has been adapted for real-time implementation. Faster execution can be implemented by using SAD through computational optimisation techniques, which has been proposed by Stefano [25, 85]. In the next section, the performance on DLR by using SAD similarity metric is presented with the Middlebury Stereo Evaluation.

Table 5.1: Processing time and RMS of Tsukuba and Map images

Algorithms	Map image		Tsukuba image	
	Time	RMS	Time	RMS
SAD	6.84	41.29	7.09	57.15
SSD	6.07	42.48	6.54	56.86
NCC	9.94	43.15	10.58	56.99
SHD	38.37	43.85	41.58	58.58
BM SAD	6.76	26.86	7.23	53.22
BM SSD	6.05	28.95	6.59	52.75
BM NCC	9.93	29.36	11.07	51.14
BM SHD	41.98	37.17	41.16	50.55

5.5.2 Performance Based on Middlebury Stereo Evaluation

Scharstein and Szelinski [22] have developed an online evaluation platform for the Middlebury Stereo Evaluation [177], which provides a large number of stereo image datasets consisting of the stereo image pair and the ground truth images. We evaluated our algorithm by using the Middlebury datasets and compared the results with many others online. The samples of these datasets are shown in the first row of Figure 5.12, which consist of ‘Tsukuba’, ‘Venus’, ‘Teddy’ and ‘Cones’ stereo pairs. Since this evaluation is very well known and state-of-the-art, the proposed algorithm in this work is also evaluated in this manner. In order to evaluate an algorithm on this website, the disparity maps of all four datasets have to be generated and uploaded online. The disparity maps have to correspond to the left stereo image and the disparities have to be scaled by a certain factor. The evaluation engine calculates the percentage of bad matched pixels within a certain error threshold by pixel-wise comparison with the ground truth image. This is done three times for each dataset. Firstly the disparity map image is evaluated for all pixels where a ground truth value is available. Secondly, it is evaluated for all non-occluded pixels. Lastly, the disparity map images are compared for all pixels at disparity discontinuities. Many researchers use this platform for evaluation and this gives a significant overview of how the developed algorithm performs in comparison to other algorithms. The platform is up-to-date and constantly updated.

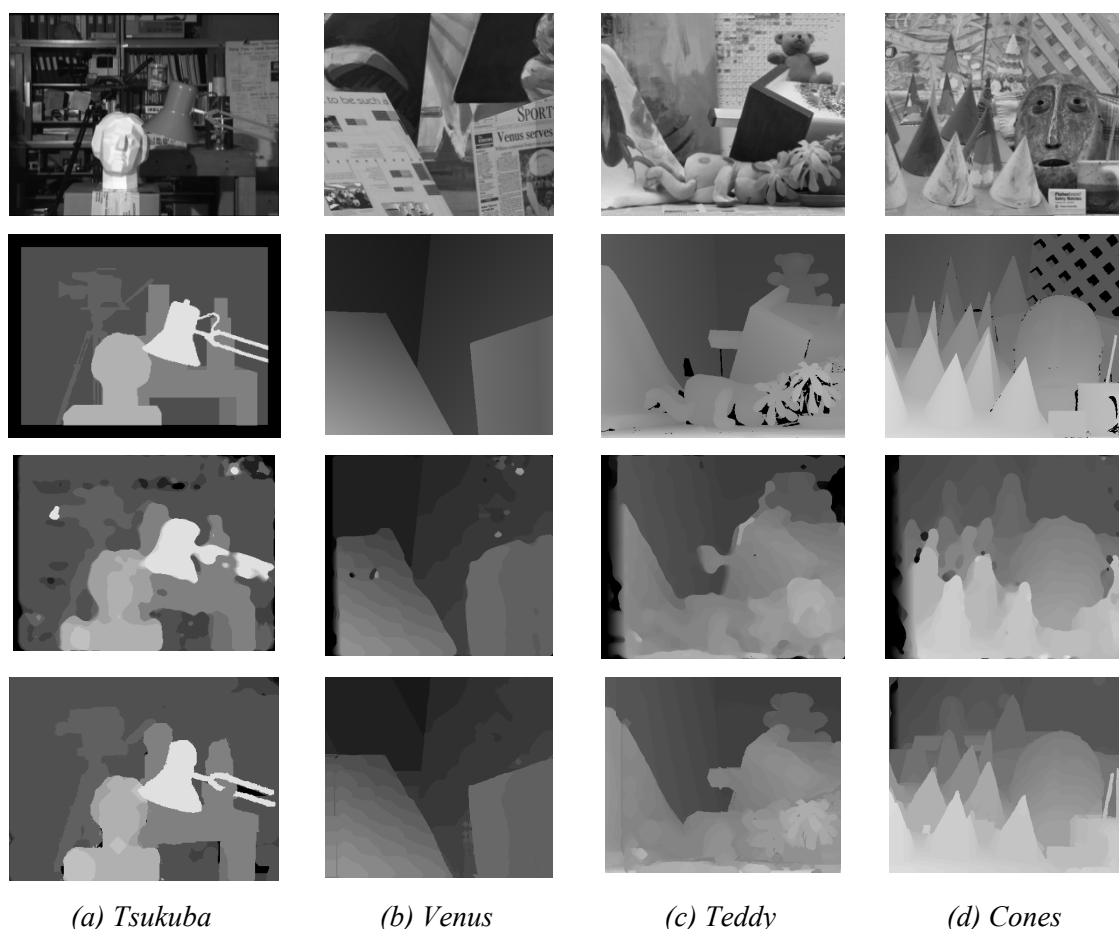


Figure 5.12: Results of the proposed method by the Middlebury benchmark datasets. The first row images are the reference images of each set. The second row images are the ground truths. The third row images are the disparity maps by left-to-right matching using SAD metric. The fourth row images are the resulting disparity maps by DLR-SAD method

Figure 5.12 shows the Middlebury evaluation datasets, the ground truths of four datasets and the resulting disparity maps estimated by SAD and DLR-SAD methods. The results based on the third row in Figure 5.12 used the fixed window SAD of 21×21 ('Cones' and 'Teddy'), 11×11 ('Tsukuba') and 25×25 ('Venus'). Additionally, an 11×11 median filter is applied as a post-processing step for the SAD. The selected parameters are chosen to achieve the best possible result for the disparity maps. The results based on SAD have been further enhanced and refined by using the proposed algorithm, DLR where every disparity depth layer has been separated. Through the DLR, the new disparity maps have been formed. The fourth row of Figure 5.12 indicates that disparity maps improved and removed the noise and errors in the basic SAD stereo matching. It is worth mentioning that several major occlusions and boundary discontinuities have been discarded. It shows the ability of DLR to deal with these problems. The morphological operation processed in separated layers enables the unwanted regions, errors and noise to be removed efficiently. Due to the erosion and dilation process in the

morphological operation, the final disparity map probably contained holes and cracks between the depth layers. Therefore, the missing values in the disparity maps have to be extrapolated with adjacent pixel values by using hole-filing techniques.

Table 5.2 shows the performance of our method by the Middlebury ranking list [177] with the error threshold of 1 pixel. All values are given in percentages: for non-occluded regions ('N-o'), all regions ('All') and discontinuities regions ('Disc'). The last column in the table is the average percentage of bad pixels ('Avg') over all twelve columns. The main ranking published on the Middlebury Stereo Page [177] is ordered by the overall performance of the algorithms (notated as 'Avg Rank' in the Table 5.2) and described in Appendix C.2. During the algorithm submission to the Middlebury Stereo Page, the rank of *Bipartite* algorithm for Tsukuba datasets are given as 78 ('N-o'), 89 ('All') and 73 ('Disc'); the *RegionalSup* placed at 96 ('N-o'), 102 ('All') and 88 ('Disc'); the *STICA* algorithm at 118 ('N-o'), 119 ('All') and 117 ('Disc'); and our method obtained at 100 ('N-o'), 92 ('All') and 108 ('Disc') for the similar datasets. The average ranks ('Avg Rank') are obtained by calculating the total ranking for each datasets. Due to the frequent and up-to-date algorithms submission, the ranking list changes dynamically. At the time of writing this thesis, the main ranking consists of a total 120 algorithms.

The basic Fixed-Window (FW) with SAD as cost aggregation methods is placed in the last ranking. It shows that the stereo matching by using the basic approach is not accurate and contains errors for all the regions, non-occluded and near depth discontinuities. However, after the FW-SAD is refined by using the proposed method of DLR, the results significantly improved and the new results moved up 13 places.

As can be seen, the results in Table 5.2 indicate our algorithm is competitive with other existing algorithms. In contrast to others, the presented algorithms of DLR obtained by using a basic similarity metric. Therefore, the complexity of the algorithm is low and can be easily adapted with any stereo matching system. Our result is the best among all nominated algorithms for the non-occluded region in the Venus dataset, and the second for the Teddy dataset. The scenes of the Venus dataset consist of many textured surfaces, such as the background and printed document. With respect to the evaluations in 'all' sections, our results are moderate since the 'all' region includes occluded regions and the occluded regions mainly consist of planes of background.

Table 5.2: Middlebury dataset ranking with the 1 pixel threshold. These values indicate the percentage of bad pixels whose errors are more than 1 pixel, where ‘N-o’ (non-occluded regions), ‘All’ (for the all regions), ‘Disc’ (near depth discontinuities regions) and ‘Avg.’ (average percentage of bad pixels over all datasets).

Algorithms	Avg Rank	Tsukuba			Venus			Teddy			Cones			Avg (%)
		N-o	All	Disc	N-o	All	Disc	N-o	All	Disc	N-o	All	Disc	
2DPOC	101.4	2.88	4.8	10.5	6.55	7.8	17.4	14.4	22	27.9	15.2	23	24.5	14.7
Bipartite	102.8	2.54	4.4	13.6	6.62	7.5	18.6	16.9	24	30.2	15.1	22	23	15.4
SAD-DLR	106.3	4.22	5.1	19.5	2.5	3.2	18.3	18.2	19	37.2	18	21	32.9	16.5
Phase-based	106.4	4.26	6.5	15.4	6.71	8.2	26.4	14.5	23	25.5	10.8	21	21.2	15.3
RegionalSup	107.2	3.99	6.1	14.2	8.14	9.7	36.8	18.3	27	32.1	9.16	19	19.9	17
BioDEM	107.4	6.57	8.4	28.1	3.61	4.8	33.7	13.2	21	34.5	6.84	16	19.8	16.4
IMCT	107.5	4.54	5.9	19.8	3.16	3.8	23.2	18	23	35.3	12.7	19	27.9	16.3
SSD+MF [79]	107.8	5.23	7.1	24.1	3.74	5.2	11.9	16.5	25	32.9	10.6	20	26.3	15.7
SO [22]	109.5	5.08	7.2	12.2	9.44	11	21.9	19.9	28	26.3	13	23	22.3	16.6
MI-nonpara	111.8	5.59	7.5	18.8	7.5	9	35	17.4	26	36.9	10.2	20	22.6	18
PhaseDiff	112.7	4.89	7.1	16.3	8.34	9.8	26	20	28	29	19.8	29	27.5	18.8
STICA	113.0	7.7	9.6	27.8	8.19	9.6	40.3	15.8	23	37.7	9.8	18	28.7	19.7
Rank+ASW	113.0	6.51	8.4	19.7	10.5	12	32.7	15.7	24	32.8	14.1	23	21.7	18.4
LCDM+AdaptWgt	113.3	5.98	7.8	22.2	14.5	15	35.9	20.8	27	38.3	8.9	17	20	19.5
Infection	114.6	7.95	9.5	28.9	4.41	5.5	31.7	17.7	25	44.4	14.3	21	38	20.7
FW-SAD	118.2	7.51	9.5	30	9.15	11	48.7	22	30	47.3	15.7	25	36.3	24.3

* Note: The results based on submission on 27th February 2012

Figure 5.13 shows the analysis and error evaluation for the non-occluded regions based on bad pixel with (absolute disparity error > 1). The first row of Figure 5.13 show the samples images for evaluation provided by Middlebury Stereo Page. The non-occluded regions visualized by the white areas while the occluded and border regions shown in black. The second row shows the errors for non-occluded regions based on FW-SAD. By comparing the non-occluded regions for the disparity depth map of the proposed algorithm, Figure 5.13 (in the third row) visually points where incorrect measurements are produced by the SAD-DLR. We can notice that the number of errors are low for the Tsukuba and Venus datasets. The incorrect disparities are higher for the Teddy and Cones datasets due to the complexity and texture regions. In general, the SAD-DLR has improved the disparity maps obtained from the FW-SAD where most of the sparse small black regions (in the second row of Figure 5.13) have been removed. One of the disadvantages of using the SAD metric is the incompetency of the similarity metric to calculate the discontinuity regions. This can be improved by selecting a different cost aggregation method.

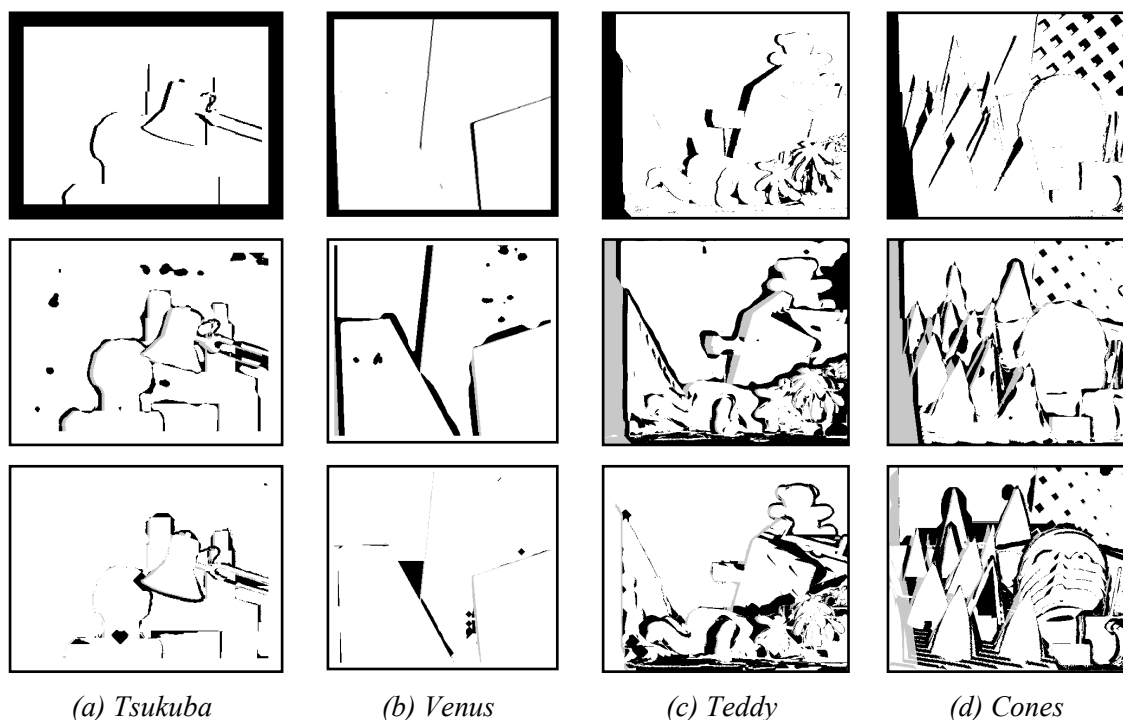


Figure 5.13: Analysis for non-occluded region based on bad pixel (absolute disparity error > 1). Non-occluded regions (white) with occluded and border regions (black)

The analysis for the signed disparity error based on discontinuity and half-occluded regions are shown in Figure 5.14. The regions near depth discontinuities are indicated in white, occluded and unknown regions are shown in black and other regions in gray. The discontinuity and half-occluded errors by using the FW-SAD and SAD-DLR are shown in the second and third row respectively. From the results, the SAD-DLR method indicated to improve the discontinuity and half-occluded regions obtained from the FW-SAD.

The results obtained have been shown to be adequate for the DLR to improve the disparity depth map. Though the DLR does not deal with the cracks and holes due to the layer separations, the merging of disparity and edge boundaries regions change the new disparity maps significantly. The performance of the DLR can be improved by using advanced matching techniques such as graph cut, segmented-matching and dynamic programming, which can produce more accurate disparity depth maps. Furthermore, a more sophisticated cost aggregation strategy could lead to better results. However, based on the performance evaluation of DLR with SAD, the results are satisfactory in term of accuracy and quality of the disparity depth maps.

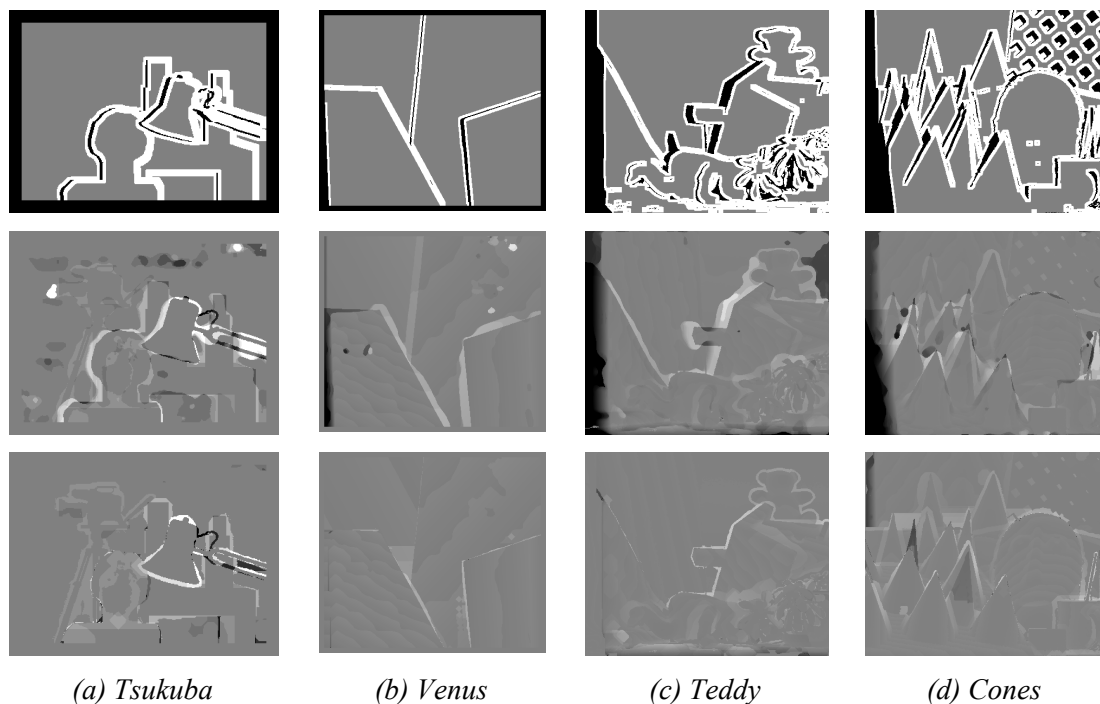


Figure 5.14: Analysis for signed disparity error based on discontinuity. Regions near depth discontinuities (white), occluded and unknown regions (black) and other regions (gray)

5.6 Conclusion

The Depth Layer Refinement (DLR) module has been presented with the aim at improving the raw disparity maps in the post-processing stage. The proposed system takes advantage of the Depth Image Layers Separation (DILS) algorithm that separate the layers of depth based on disparity range. Each particular disparity depth map is distributed along with the disparity range and can be divided into several segments known as layers. Instead of using each layer to synthesize inter-view images in the DILS, the layer will be mapped to segment reference image to refine the disparity depth map. The algorithm has used a simple stereo matching correspondence algorithm with a basic similarity metric of SAD. The similarity metric will search the pixel points between the left and right images under the Fixed Window (FW) searching process. With this approach, the raw disparity depth map obtained is not smooth and contains errors particularly in the depth discontinuities, and was unable to detect the uniform areas and repetitive patterns. In spite of its limitation, SAD with FW is the most frequently used algorithm for real time applications due to it is easy at implementation, fast and has limited memory requirements. Therefore, the fixed window similarity stereo matching technique is adequate to obtain the estimated depth map.

We have analyzed the differences between the window size and similarity metric used for the stereo matching engine. The SAD selected as the main similarity metric in the cost aggregation due to its simplicity, reliability and low computational cost. The resulting disparity maps are evaluated on the Middlebury Stereo Vision website and perform well in comparison to other algorithms although it only uses the basic similarity metric of SAD. Qualitative and quantitative evaluation proved the satisfactory quality of the achieved matching results. The proposed method has improved by 13 places from the last place after the basic FW-SAD was refined by using DLR in the online evaluation in the Middlebury Stereo Vision website. We found that the proposed technique removes the noise and unmatched pixels on the fixed window searching SAD. It also improved the depth discontinuities of the disparity depth maps.

The limitation of the presented approach lies on the assumption that the scene can be well approximated by a set of rectified images. In the future development, the system can be incorporated in real-time implementation, which can be used for the novel inter-view synthesis algorithm for 3D video and free-viewpoint applications. The proposed algorithm is quite practical for applications such as robot navigation and autonomous operations.

Chapter 6

Multi-Level View Synthesis (MLVS) based on DILS Algorithm for Multi-Camera Array

6.1 Introduction

In 3D vision, most image processing and stereo vision based approaches use image pairs captured by left-to-right in the epipolar of horizontal line. The image pairs are rectified before being processed with stereo matching algorithms to obtain the disparity depth map and matching correspondence pixels. This is similar to the human visual system. However, when the application requires multiple camera arrays configuration, the cameras will not be fitted in stereo. For example, in the dense camera and free-viewpoint television system, the cameras can be arranged in many locations. The main interest in this research is to create multi-perspective panoramas from the multiple cameras.

This chapter describes a novel view synthesis in the multi-view for 3D vision and free viewpoint technique for video application such as in light field imaging. This method exploits the advantage of the new inter-view interpolation algorithms described in Section 4.5, by extending stereo to multiple camera configurations. In this technique, novel multi-view synthesis created based on a limited number of cameras for sparse camera arrays. This will reduce the camera usage required to create dense images. This method is known as the Multi-Level View Synthesis (MLVS), which finds the pixel correspondences and synthesis through three levels of matching and synthesis process. The first stage identifies the pixel correspondences and synthesis based on left-right image pairs, while the second stage is based on upper-lower image pairs. The third stage uses the output obtained in the first or second stage for the new inter-view synthesis to create full virtual multi-camera array image views. The new structures and design are

shown to offer improved performance and provide additional views with fewer cameras arrangements compared to the conventional high volume camera configurations for free-viewpoint video acquisition.

The chapter is organized into eight sections. Section 6.2 describes the multi-camera array configuration and its applications. Section 6.3 provides an overview of the proposed system design architecture and the MLVS algorithm is described. The experimental results and the parameters used in the algorithm are described in Section 6.4. The next three sections present some performance results and related discussion for the multi-camera datasets. Section 6.5 provides some essentials analysis of the first selected multi-camera dataset. Section 6.6 discusses the results of MLVS by incorporating it to different multi-camera datasets. The algorithm implementation issues are discussed in Section 6.7. Finally, Section 6.8 provides some concluding remarks.

6.2 Multi-Camera Array Configuration and Applications

Multi-camera systems can function in many ways, depending on the arrangement of the cameras. The basic horizontal parallel camera arrangement has been used in the Free-Viewpoint Video (FVV) capturing system for Free-viewpoint Television (FTV) has been developed by Nagoya University [35, 36]. FTV enables the viewer to view a 3D scene by selecting any viewpoints of the scenes. The standardization of FTV was due to Multi-view Video Coding (MVC), which enables the efficient coding of multiple camera views. The video acquisition of the system comprises 16 cameras, 16 clients and 1 server that is connected to a Gigabit Ethernet. The system configuration has been developed with 100-camera system to capture larger space and can generate free-viewpoint images. Most of the current techniques for FVV are designed around a multi-camera studio environment with controlled lighting and well-calibrated static cameras to perform at an acceptable quality [49].

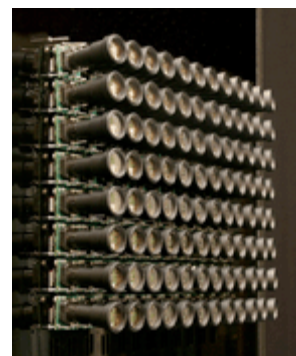
A similar system with the multi-camera array configurations has also been developed for light field imaging, which rely solely on epipolar geometry. Light fields were introduced into computer graphics in 1996 by Marc Levoy and Pat Hanrahan [6]. The main proposed application of light fields was Image Based Rendering (IBR), which computes new views of a scene from pre-existing views without the need for scene geometry. It uses a geometric interpretation of the relationship between the source

images to obtain correspondence between pixels and does not contain any information about the specifics of the scene geometry. It depends on dense sampling and the synthetic views that can be produced are constrained to lie close to the original camera locations.

The light field imaging system has been implemented with multi-camera array configuration by the Stanford Computer Graphic Laboratory in their project of the Stanford Multi-Camera Array, which is shown in Figure 6.1. In particular, if the cameras are packed close together, then the system effectively functions as a single-centre-of-projection synthetic camera. It can be configured to provide unprecedented performance along one or more imaging dimensions, such as resolution, signal-to-noise ratio, dynamic range, depth of field, frame rate or spectral sensitivity. If the cameras are placed further apart, then the system functions as a multiple-centre-of-projection camera, and the data it captures is called a light field [6, 159]. The particular interests in their project are novel methods for estimating 3D scene geometry from the dense imagery captured by the array, and novel ways to construct multi-perspective panoramas from light fields. These techniques have potential applications in scientific imaging, remote sensing, underwater photography, surveillance and cinematic special effects.



(a) The 128 video cameras are arranged 2 inches apart to simulate a single camera with an aperture 3 feet wide.



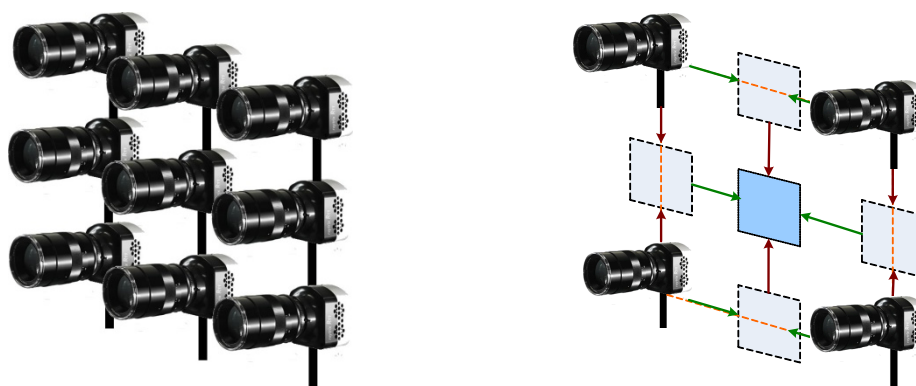
(b) The camera attached with telephoto lenses to create a very high-resolution video camera up to 6,000 pixels wide.

Figure 6.1: Multi-camera array configuration by the Computer Graphic Laboratory, Stanford University

The main interest in the multi-camera array configuration is increased with the reducing costs of the digital cameras. However, more cameras require complex camera calibration and configuration to ensure the cameras can be integrated and work seamlessly. The large number of cameras for multi-view imaging needs high

processing, bandwidth and storage requirements. This will increase the additional cost to the system. How to make the multi-camera affordable in typical environments? By using a small number of cameras, can we create a ‘virtual’ camera that can be viewed as if there is an extra camera attached to it? The cameras in the multi-camera system are arranged along the horizontal and vertical baselines as it has been specifically developed for dense imagery. Based on the example in FVV and light-field imaging, it clearly shows a high number of cameras required for the image and video acquisition for both of the system. Therefore, our research work is to provide a multi-camera system of sufficient images by reducing the number of cameras required for image acquisition, while maintaining the quality of the virtual view synthesis images. This is one of the main contributions in the MLVS algorithm by creating virtual multi-view images for a minimal number of multi-cameras array configurations.

The basic 4D (x, y, V_x, V_y) multi-camera array configuration, arranged in horizontal and vertical baselines shown in Figure 6.2(a), consists of nine cameras that produce 3x3 views. For this case, we will identify it as group of camera array (GCA). The MLVS algorithm will create the inter-view ‘virtual’ cameras for the similar camera array configuration as illustrated in Figure 6.2(b) to simulate the light-field imaging and FVV application. With this approach, only four cameras are required to capture the full nine views for the light field and FVV while maintaining the quality of the synthesis view images. The baseline camera ratio, β between the left-right and upper-lower camera pairs is used to create the virtual camera views location. Multiple views can be synthesized between the cameras to produce a 4D dense camera array configuration with less cost and required cameras.



(a) Multi-camera array configuration with 9 camera views.

(b) Multi-camera array configuration of 4 camera views and additional 5 virtual views.

Figure 6.2: Sample of multi-camera configuration

6.3 System Design Architecture

The system architecture for MLVS can be divided into three main categories: Level 1 to synthesize an image based on the horizontal stereo matching, Level 2 is responsible for vertical stereo matching for upper and lower images to synthesize the virtual inter-view image and Level 3 to create the image synthesis by using the results obtained from either Level 1 or 2. The system design architecture for MLVS algorithm is shown in Figure 6.3, distinguishing the three levels of matching and synthesis. The image view synthesis module (in Figure 6.4) comprises the disparity refinement, DILS algorithms and hole-filling techniques as presented in the Chapter 4.

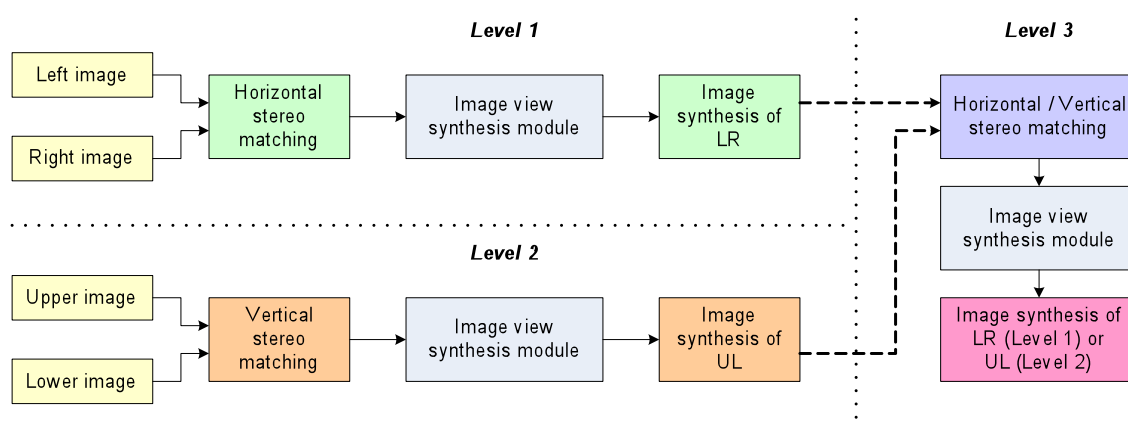


Figure 6.3: System design architecture for MLVS algorithm

The matching and synthesis of Level 1 and 2 in the MLVS algorithm determines the inter-view image synthesis between the left-right and upper-lower views, which is along the real cameras configuration as illustrated in Figure 6.2(b). Meanwhile, the middle image view synthesis of the multi-camera array configuration can be obtained through MLVS Level 3, where the image matching and synthesis is based on the result from Level 1 or 2. If the inputs are taken from Level 1, the vertical stereo matching will be processed in the entire Level 3. Otherwise, the horizontal stereo matching is used to synthesize the images obtained from Level 2. The comparative results are based on subjective and quantitative evaluation, between the image matching and synthesis on these two outcomes shown that there are no significant differences.

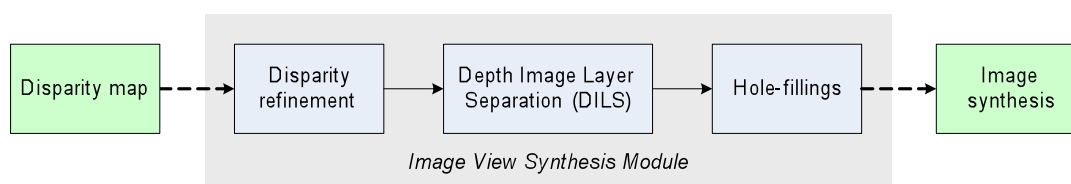


Figure 6.4: Image view synthesis module of MLVS algorithm

The outcomes of MLVS for every level in the proposed system design are illustrated in Figure 6.5, where it shows the three levels of matching and synthesis. The inter-view synthesis image in MLVS Level 1 is obtained from the Left-to-Right (UL) matching and DILS synthesis. From the example, the upper images for the synthesis are upper-left ($I_{U,L}$) and upper-right ($I_{U,R}$) which will produce $I_{U(LR)}$. The lower image pair ($I_{L,L}$ and $I_{L,R}$) is used to create the new synthesis image of $I_{L(LR)}$ in the Level 1 of MLVS. The synthesized images from the Level 1 will be used as the input for MLVS in Level 3 based on upper-lower (vertical) stereo matching and DILS synthesis to create the middle novel view synthesis image, $I_{b(UL)}$.

In Level 2 of the MLVS system, the vertical stereo matching is applied on image pairs of upper and lower for the left and right section as shown in Figure 6.5, where the matching is performed between $I_{U,L}$ to $I_{L,L}$ and $I_{U,R}$ to $I_{L,R}$. The resulting synthesis images through this level are called $I_{L(UL)}$ and $I_{R(UL)}$, where the UL notation indicates the vertical matching based on upper-lower. The results based on Level 2 are used for MLVS Level 3, wherein the matching and synthesis is done based on left-right (horizontal) stereo matching and DILS synthesis for the final image, $I_{b(UL)}$.

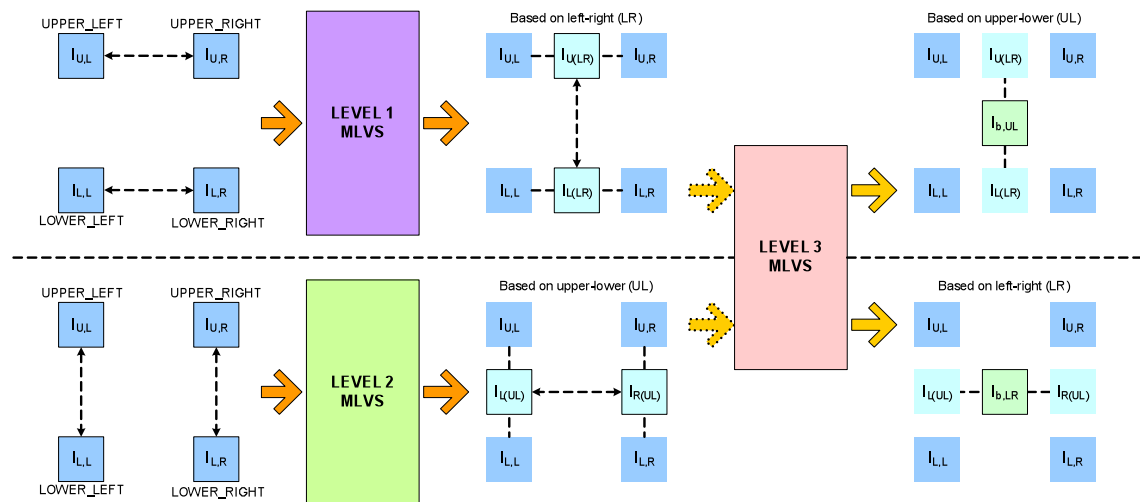


Figure 6.5: The image matching and synthesis based on the three different levels of MLVS

All the synthesis images created from Level 1 to 3 comprise the full multi-view images in the group of camera arrays consisting of four original real views with five ‘virtual’ views. As described earlier, the number of ‘virtual’ views can be increased to a number of images by synthesizing the images based on camera baseline ratio, $0 \leq \beta \leq 1$. The next section will discuss the MLVS algorithm for every level in detail.

6.3.1 Multi-Level View Synthesis Algorithm

The main aim of this algorithm is to create multiple virtual camera views based on four basic camera arrangements as a single Group Camera Array (GCA) as illustrated in Figure 6.6. The upper row is divided into left and right image pairs, which are $I_{U,L}$ for the upper-left image and $I_{U,R}$ for the upper-right image respectively. Meanwhile the similar configuration of left and right image is for the lower row of the camera array, $I_{L,L}$ (lower-left image) and $I_{L,R}$ (lower-right image). This basic configuration can be expanded (by horizontal or vertical) into several groups of camera array for larger multi-image arrays. The MLVS algorithm consists of three different phases or levels, which represent the newly inter-view images obtained, either for horizontal, vertical or middle views. The images in the GCA are assumed to be calibrated and rectified accordingly to speed up the matching process, where the searching is along the epipolar line only.

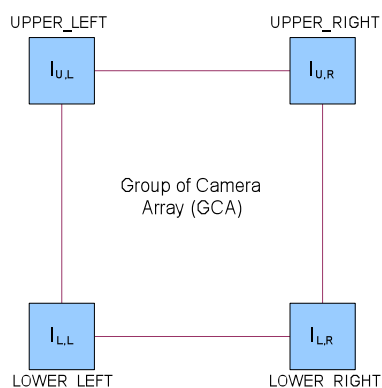


Figure 6.6: Basic arrangement for the group of camera array, which consists of four images, obtained from four cameras views

The main differences between the levels are the matching and synthesis directions. The image view synthesis module is the same for each level that includes the disparity refinement, depth image layer separation and synthesis as depicted in Figure 6.4. The detail description of image view synthesis module will not be discussed in detail as it was explained in Section 4.3.2 of Chapter 4.

In the first phase, the new inter-view images are synthesized based on the Left-to-Right (LR) matching as illustrated in Figure 6.7. Basically, the algorithm for the first level is similar to normal stereo matching and image view synthesis in the DILS algorithm. For the proposed implementation, the SAD function is used as the similarity metric between the left and right stereo pair (for upper and lower images) as defined in Equation (4.5) in

Section 4.3.1.1. This is the horizontal stereo matching, which has been implemented in almost every conventional approach for the stereo matching algorithms. The disparity depth map is calculated based on Bidirectional Matching (BM) that consist of the Left-to-Right Consistency Check (LRCC) technique proposed by [24]. This approach eliminates the occlusion and discontinuity errors from the stereo matching algorithm.

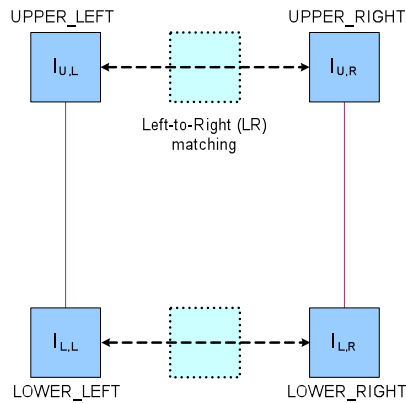


Figure 6.7: The matching and synthesis for the Level 1 of MLVS

The resulting raw disparity depth map refined to make it smoother and remove the noises. For this case, the median filter with size of 11x11 will be used. In median filtering, the neighbouring pixels are ranked according to their brightness (intensity) and the median value becomes the new value for the central pixel. Median filters can do an excellent job to reject certain types of noises, in particular, ‘shot’ or impulse noise in which some individual pixels have extreme values.

The resulting disparity depth map is then used to determine the virtual inter-view image. The DILS algorithm is used to divide the disparity depth map into several layers according to their depth and distances of the object. Each layer’s visual information is finally interpolated to create the new virtual inter-view images between the left and right cameras using Equation (4.17) in Section 4.3.2.6. The intermediate view image of layer i is then translated according to the camera baseline ratio β as defined by Equation (4.18) in Section 4.3.2.7.

A number of inter-view images can be created and synthesized by defining the camera baseline ratio. For example, if the new image is synthesized in the middle of left and right, then the β will be 0.5. More inter-view images can be synthesized as long as it follows the limitation of the camera baseline ratio. Additional synthesized view images will create more ‘virtual’ views along the matching. This will enable multiple views

along the inter-view image pairs for upper and lower virtual camera views. The intermediate view for each layer will be compiled together as a single image view synthesis $I_{VS}(x, y)$ according to Equation (4.20) in Section 4.3.2.8.

Based on the camera array illustrated in Figure 6.7, two image view synthesis are created for the upper and lower rows called, $I_{VS(upper)}$ and $I_{VS(lower)}$ in the first level of MLVS. Since the image synthesized is through Left-to-Right (LR) matching, we denote this as $I_{U(LR)}$ and $I_{L(LR)}$ for image synthesis Left-to-Right (LR) upper and lower respectively to differentiate the image synthesis obtained from the second level of MLVS. The generated inter-view image synthesis will be corrected with hole-filling techniques to remove the hole and cracks caused by the occlusions and translation processes as discussed in Section 4.3.2.9.

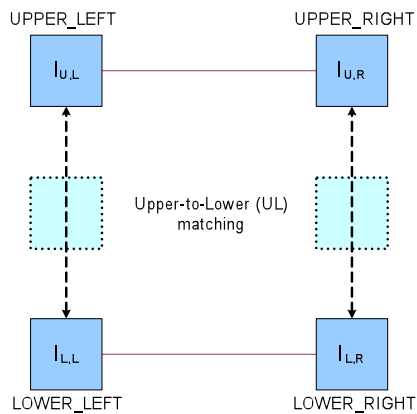


Figure 6.8: The matching and inter-view synthesis for the Level 2 of MLVS

The second phase, known as the MLVS Level 2 is basically the same matching and synthesis as described in Level 1. However, instead of obtaining the inter-view image between the left and right image, Level 2 focuses on the image based on the upper and lower image, which is known as the vertical stereo matching. For this case, the matching correspondence pixels found within upper and lower images for left and right respectively are shown in Figure 6.8. Even though the process is similar to the Left-to-Right (LR) matching, it still requires a different approach to finding the disparity range and depth. In left-to-right matching, the disparity depth map image is based on disparity range in the image columns. However, in the Upper-to-Lower (UL) matching, the disparity range for the matching pixel points is based on the image row through vertical matching as described as follows:

$$SAD(x, y, d) = \sum_{i, j=-n}^n |I_U(x + i, y + j) - I_L(x + i, y + d + j)| \quad (6.1)$$

where $I_U(x, y)$ and $I_L(x, y)$ are the gray-level intensities of the upper and lower image respectively, the window size is $n \times n$, and d is the disparity. After the disparity depth map is obtained in the MLVS Level 2, it will be refined with ULCC (Upper-Lower Consistency Check) that works similarly with LRCC in order to remove the occlusion regions. The disparity depth maps are also smoothed out using a median filter. Through the image view synthesis module, two pairs of inter-view image synthesis are generated using the camera ratio baseline, β , as shown in the Figure 6.8. The inter-view image synthesis based on MLVS Level 2 can be notated as $I_{L(UL)}$ and $I_{R(UL)}$ for the left and right image synthesis within the same camera ratio baseline. In this example, the new virtual image is synthesized for only one camera ratio baseline. The virtual inter-view images can be increased between the upper and lower images by including a series of camera baseline ratio between 0 and 1 to provide dense camera array along the vertical lines.

By setting the camera baseline ratio to a single value as $\beta=0.5$ for the GCA of the basic four camera arrangements (in Figure 6.6). After the MLVS Level 1 and Level 2 process, four pairs of new ‘virtual’ inter-view images are created on $\beta=0.5$. From the MLVS Level 1, the new virtual inter-view images are $I_{U(LR)}$ and $I_{L(LR)}$, which are the upper and lower images based on the Left-to-Right (LR) matching. By using the same approach, the new virtual inter-view images in the MLVS Level 2 are $I_{L(UL)}$ and $I_{R(UL)}$ that are the left and right images based on the Upper-to-Lower (UL) matching.

The Level 3 of the MLVS algorithm is used to generate the middle image in the group camera array. In order to synthesize the image in Level 3, either the ‘virtual’ inter-view image obtained in Level 1 or 2 can be used. If the outputs based on the left-to-right matching in Level 1 ($I_{U(LR)}$ and $I_{L(LR)}$) are used, then the Upper-to-Lower (UL) matching is selected in the MLVS Level 3 as shown in Figure 6.9(a). Otherwise, if the outputs based on the Upper-to-Lower (UL) matching in MLVS Level 2 ($I_{L(UL)}$ and $I_{R(UL)}$) are used, the Left-to-Right (LR) matching is selected as illustrated in Figure 6.9(b). Based on the quantitative performance obtained through the experimentation, the results based on the LR and UL matching in MLVS Level 3 do not provide distinct differences between the two approaches.

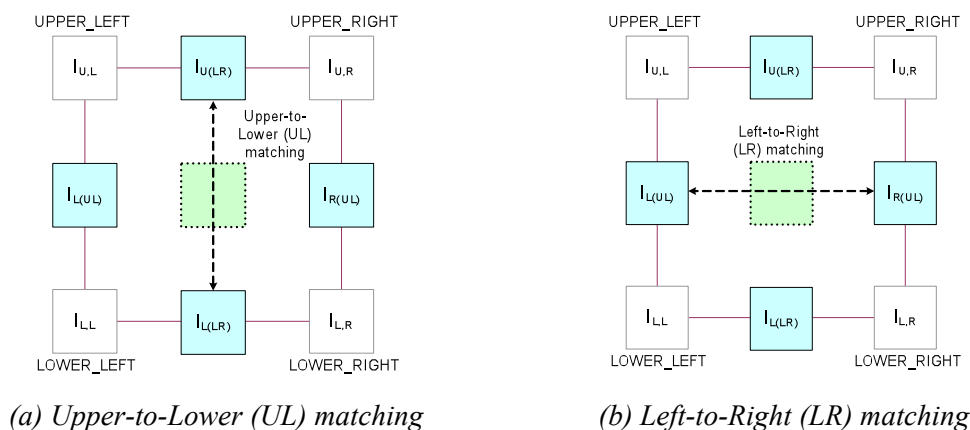


Figure 6.9: Level 3 MLVS algorithm

Based on the MLVS algorithms, new ‘virtual’ inter-view images are created based on four basic camera array configurations. In this example, another five virtual views have been synthesized in between the original camera views. By defining the camera baseline ratio into several points, more virtual inter-view images can be created. This will compose the multi-image into a light-field image rendering, which is useful for free-viewpoint image and video applications. With this approach, a large number of cameras for multi-view camera application can be reduced, such as in the light field imaging system. This approach not only reduces the cost, it also eliminates the complexity of camera calibration and configuration. In addition, the compression can be done mainly for the real camera views by using Multi-view Video Coding (MVC).

The basic group of camera array configurations shown in Figure 6.6 can be expanded into a bigger system with additional camera arrangements to the horizontal or vertical sections as shown in Figure 6.10. The new additions must be in the same epipolar lines to ensure the matching pixels between the images can be obtained through the matching algorithm. This can be done with the camera calibration configuration and rectification along the cameras through the horizontal and vertical lines. In GCA 1, the system consists of four cameras that are labelled as m_n , where m is for the row and n for the column in the camera arrays configuration. The similar size of GCA expands to the horizontal line as illustrated in Figure 6.11, where the new extension creates the GCA into several groups. The similar extension can be done along the vertical line. The framework of MLVS holds the basic structure for the multi-camera array configurations to create full dense multi-camera views. The analysis for this type of camera array configurations will be discussed in the results and performance analysis section.

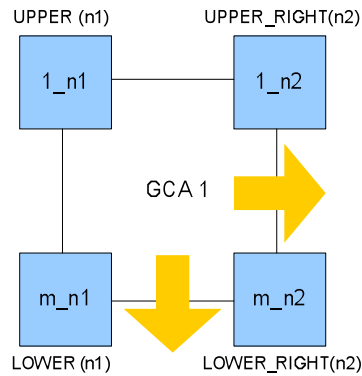


Figure 6.10: The group of camera array can be expanded horizontally or vertically depending on the required number of virtual inter-view cameras

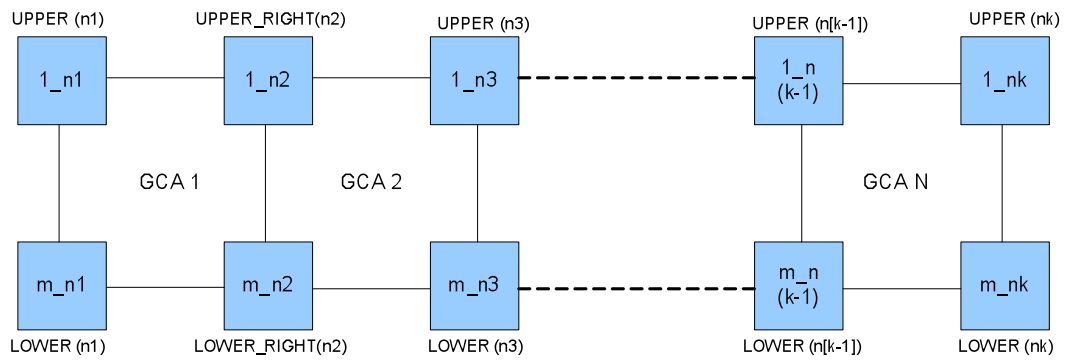


Figure 6.11: Expansion of group of camera array (GCA) with additional camera configuration along the horizontal lines

6.3.2 Multi-Camera Array Datasets

In this work, we are using the sample of multi-camera datasets provided by the Stanford Light field research group on their Light Field Multi-Camera Array [33, 34]. The proposed method in our research with MLVS is to create a complete view of the multi-camera images for a sparsely cameras array to emulate the complete cameras array that are placed closely together. For the experimental test, the real cameras are selected from the Stanford multi-camera datasets that are placed at an intermediate spacing, such as from camera 1 to 5, 5 to 10, 10 to 15 and so on. As an example, the matching of the image pairs will be done along the camera 1 as the reference (left) and camera 5 as the target (right). The disparity depth map obtained with this pixel correspondence stereo matching is used in the DILS algorithm to create virtual inter-view images along camera 1 to 5, which is the ‘new view’ for camera 2, 3 and 4. Similar procedure executed for camera pair 5-10, 10-15 and others along the horizontal and vertical arrays. The synthesized inter-view images created by the DILS algorithms are compared with the

original images in the multi-camera arrays data set using signal-to-noise ratio and Structural SIMilarity (SSIM) index [165].

Two data sets from the Stanford Multi-Camera Array are used to test the MLVS algorithm, which are Cookies (21x5 views) and Lego (17x17 views). The sample of image of their first camera is shown in Figure 6.12.

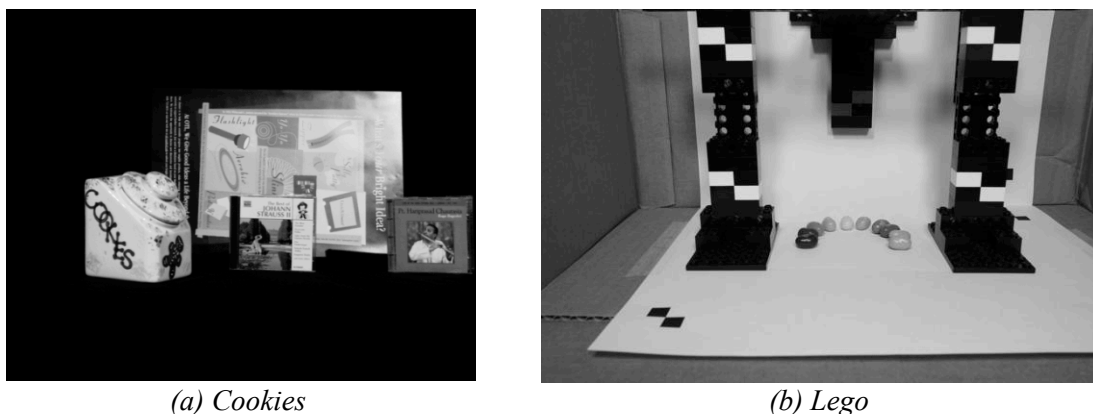


Figure 6.12: Sample of the first image camera from Stanford Multi-Camera Array

6.4 Experimental Results

We evaluate our algorithm using the Stanford datasets to compare with conventional inter-view synthesis methods. For this evaluation, we used several β values in the DILS synthesis to create multiple virtual inter-view images along the horizontal and vertical lines of the multi-camera array. The parameters of the evaluation are tabulated in Table 6.1 for Cookies and Lego datasets. The first level determines the disparity depth map between the left and right image based on the selected camera index, such as 1-6 for Cookies and 1-5 for Lego data. With this information, four newly virtual inter-view images are created along the horizontal line of Cookies data between camera 1 and 6. Meanwhile, three virtual inter-view images were synthesized for Lego in the first level.

In the second level, the matching and synthesis were applied between the upper and lower image from the camera index 1 and 5. Three virtual inter-view cameras were synthesized using camera baseline ratio, β , of 0.25, 0.5 and 0.75 to allocate the inter-view images at camera location 2, 3 and 4 respectively.

Table 6.1: Parameter of Cookies and Lego datasets for stereo matching and synthesis in DILS algorithm

Parameter	Cookies				Lego		
Size	21x5				17x17		
Level 1: Horizontal	Left		Right		Left		Right
	1		6		1		5
Disparity range	43				35		
β	0.2	0.4	0.6	0.8	0.25	0.5	0.75
Inter-view camera	2	3	4	5	2	3	4
Level 2: Vertical	Upper		Lower		Upper		Lower
	1		5		1		5
Disparity range	30				34		
β	0.25	0.5	0.75		0.25	0.5	0.75
Inter-view camera	2	3	4		2	3	4

The full inter-view images were synthesized based on Level 1, 2 and 3 in the MLVS for Cookies datasets, as illustrated in Figure 6.13, for the Group of Camera Array (GCA) 1. During Level 1 matching and synthesis, the matching and synthesis is done for left-to-right for the upper and lower rows, which are 1_1 (upper left, I_{UL}) with 1_6 (upper right, I_{UR}) and 5_1 (lower left, I_{LL}) with 5_6 (lower right, I_{LR}) respectively. The DILS algorithm along with this matching is composed of four inter-view images for each row based on camera baseline β at 0.2, 0.4, 0.6 and 0.8 respectively. The second level in the MLVS is to synthesize the inter-view image along the vertical line, where the disparity depth map and synthesis images are created with upper and lower images.

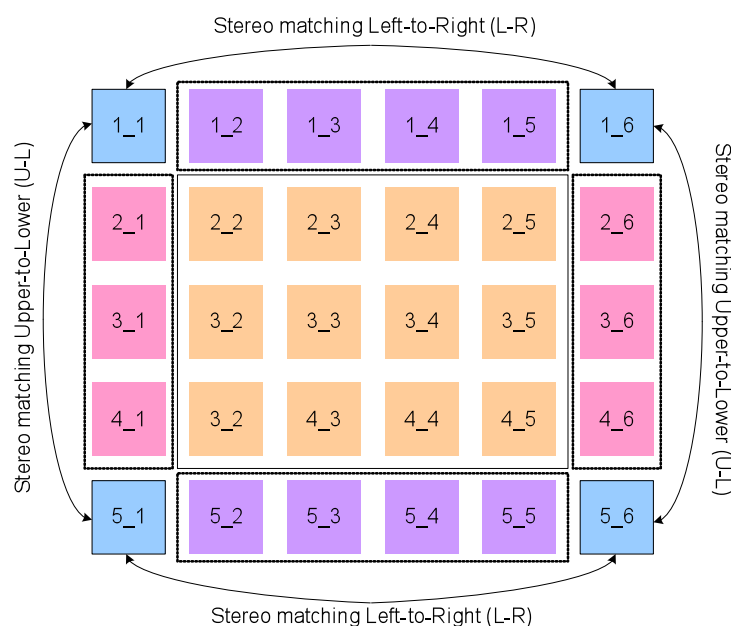


Figure 6.13: Multi-level view synthesis (MLVS) for Group of Camera Array (GCA) 1 in Cookies datasets

As shown in Figure 6.13, Level 2 matching and synthesis are based on the images between 1_1 (upper left, I_{UL}) with 5_1 (lower left, I_{LL}) and 1_6 (upper right, I_{UR}) with 5_6 (lower right, I_{LR}). With the value of β set to 0.25, 0.5 and 0.75, three inter-view images are synthesized along the row lines between the upper and lower region. This inter-view images are known as $I_{L(UL), \beta}$ and $I_{R(UL), \beta}$, where UL refers to the Upper-to-Lower (UL) matching process for left and right images. The inter-view images based on the Left-to-Right (LR) matching are notated as $I_{U(LR), \beta}$ and $I_{L(LR), \beta}$ for the upper and lower inter-view images. The inter-view images obtained in Level 1 and 2 will be used to synthesize the images through Level 3 composition in the middle region for the group of the cameras array.

To showcase the results of the MLVS algorithm, the section is divided into two main parts: Cookies and Lego datasets. The first data compilation is based on Cookies multi-camera datasets, which consists of 21x5 views. For simplicity, the inter-view synthesized images shown in the results are taken from the first group of the camera array that is within 6x5 views starting from the first column of the camera arrays. The whole camera array will be synthesized for the next group of camera array by extending the MLVS into the next column of the camera array. The inter-view synthesized images in the MLVS are compared with the conventional interpolation view method.

The second data is based on the Lego multi-camera datasets with 17x17 views. The evaluation for these dataset is based on the group of camera arrays of 5x5 and 9x9. The comparison of the different group displayed and analyzed using PSNR and SSIM metrics.

6.5 Cookies Datasets Analysis

The first part of the analysis is based on the Cookies datasets originally with 21x5 views. For the MLVS experimental, only 5x2 real camera views of this datasets are used to reproduce 21x5 views. The selected views along with the column are 1, 6, 11, 16 and 21 for the first and last row in the camera arrays.

6.5.1 MLVS: Level 1

The MLVS of Level 1 is based on left-to-right matching between the reference and

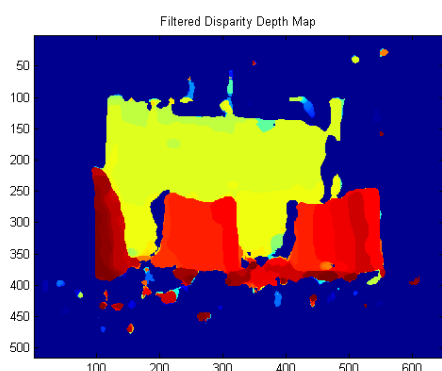
target image pairs. Figure 6.14(a) and Figure 6.14(b) shows the original left image at camera 1 as the reference and right image at camera 6 as the target image in the first row of camera arrays. Based on the stereo matching with the image pairs, the disparity depth map obtained is shown in Figure 6.14(c). The histogram distribution of the disparity depth map is displayed in Figure 6.14(d) a large amount of low disparity values because of the dark background in the image, which do not have much texture information.



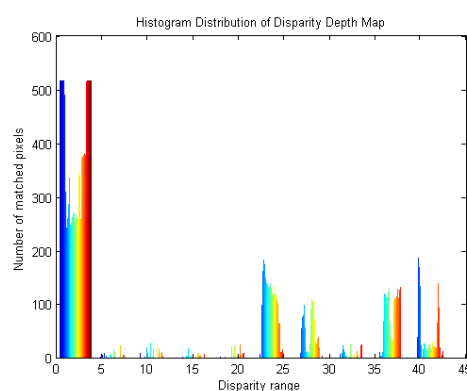
(a) Original left image of camera 1_1



(b) Original right image of camera 1_6



(c) Disparity depth map



(d) Histogram distribution of the disparity depth map

Figure 6.14: Cookies stereo image pair in the first row of the group camera array datasets.

Figure 6.15(a) shows the sample of inter-view images obtained with the conventional interpolation method at $\beta=0.4$, which is located at camera 3 in the first row in the multi-camera Cookies datasets. The SSIM image map for this approach is shown in Figure 6.15(b). The inter-view image at the same location is synthesized using MLVS at Level 1 is shown in Figure 6.15(c) with the SSIM map in Figure 6.15(d). The higher index value of SSIM (as discussed in Section 3.7.2) indicates the higher interpolated image is similar to the original. In this result, the SSIM for MLVS at Level 1 is 0.93 compared to the conventional method, which is 0.85. Based on the subjective evaluation in the SSIM image map, the dark regions, which define the errors are appeared fewer in the MLVS

output (Figure 6.15(d)) compared with the conventional method (Figure 6.15(b)).



(a) Synthesized image using the conventional interpolation method of inter-view image at $\beta=0.4$



(b) SSIM map for conventional method (SSIM=0.85)



(c) Synthesized image using MLVS with DILS of inter-view image at $\beta=0.4$



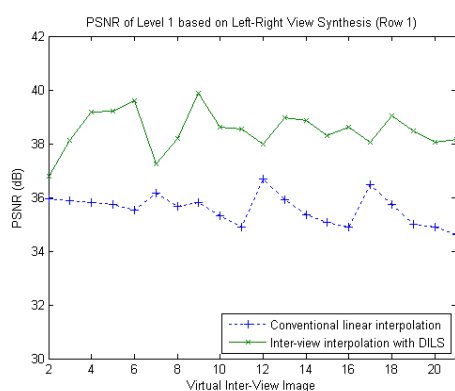
(d) SSIM map for MLVS method (SSIM=0.93)

Figure 6.15: The comparison of conventional interpolation and MLVS (Level 1) method. The sample of inter-view image synthesized at $\beta=0.4$ (camera 3) for the first row of Cookies datasets and the structural similarity index map.

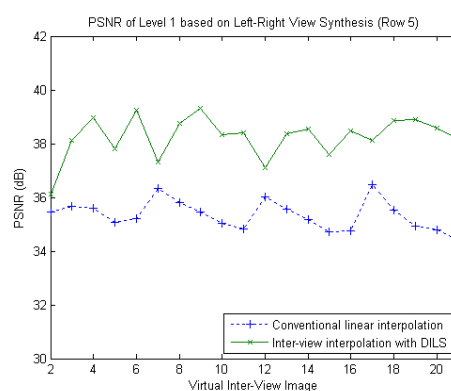
The full quantitative performance in PSNR and SSIM for the first level of MLVS is shown in Table 6.2. The value of β also indicates the camera location for the left-to-right image pairs, which is between camera 1 and 6. The table shown is for the MLVS in Level 1, which is along the row at 1 and 5 for the Cookies multi-camera datasets. The PSNR and SSIM obtained for the MLVS method are higher compared to the conventional interpolation method. The mean squared error (MSE) is lower for every inter-view images location by using the MLVS. Full matching and synthesis for the whole group of camera arrays in Cookies dataset in Level 1 is shown in Figure 6.16, which shows the graph of PSNR along the first and fifth row. Figure 6.17 is the SSIM for the first row to compare the results on conventional interpolation and the MLVS method. The SSIM results generated by using MLVS constantly higher, which is more than 0.9.

Table 6.2: Comparison results for the conventional Linear Interpolation (LI) and the MLVS method (for Level 1) along the row 1 and 5 in a group of camera array.

Original image	Camera ratio (β)	2(0.2)		3(0.4)		4(0.6)		5(0.8)	
Left-to-Right	Method	LI	MLVS	LI	MLVS	LI	MLVS	LI	MLVS
1_1 to 1_6	PSNR	35.94	36.78	35.88	38.11	35.79	39.19	35.73	39.21
	MSE	16.55	13.64	16.77	10.06	17.13	7.83	17.37	7.8
	SSIM	0.89	0.93	0.85	0.93	0.83	0.92	0.8	0.91
5_1 to 5_6	PSNR	35.45	36.14	35.65	38.11	35.59	38.98	35.06	37.82
	MSE	18.53	15.83	17.71	10.05	17.93	8.22	20.28	10.74
	SSIM	0.88	0.93	0.84	0.94	0.82	0.92	0.79	0.91



(a) PSNR of Row 1



(b) PSNR of Row 5

Figure 6.16: PSNR results for the Level 1 of the MLVS using Left-to-Right (LR) matching DILS algorithm compared to the conventional inter-view interpolation for the first and fifth row of Cookies datasets.

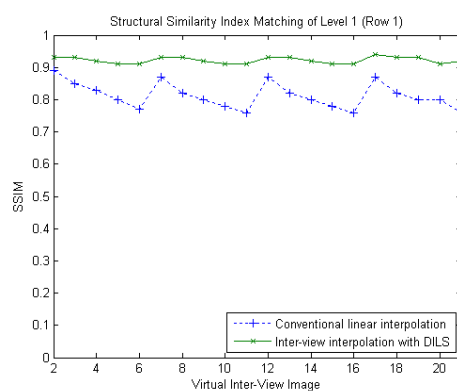


Figure 6.17: Comparison SSIM results for the Level 1 of the MLVS and the conventional inter-view interpolation for the first row of Cookies datasets.

6.5.2 MLVS: Level 2

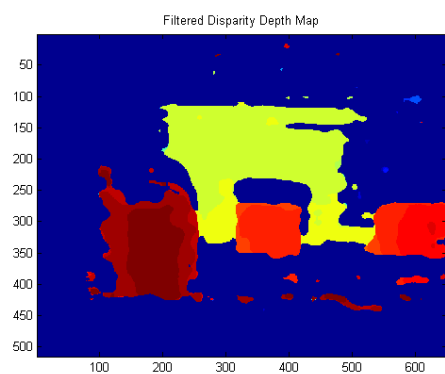
The next results will be on the second level of the MLVS which is to find the pixel correspondence matching and the inter-view images along the column. MLVS Level 2 is based on Upper-to-Lower (UL) matching between the reference and the target image pairs. Figure 6.18(a) and Figure 6.18(b) shows the original upper left image at camera 1 as the reference and lower left image at camera 5 as the target image in the first column of camera arrays. Based on the Upper-Lower (UL) matching with the image pairs, the resulting disparity depth map is shown in Figure 6.18(c). The histogram distribution of the disparity depth map is displayed in Figure 6.18(d).



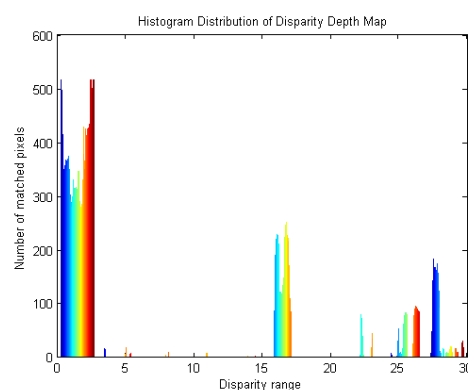
(a) Original upper left image of camera 1_1



(b) Original lower left image of camera 5_1



(c) Disparity depth map

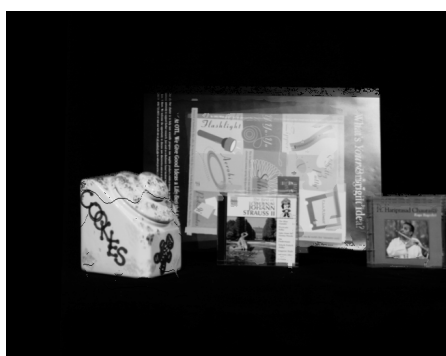


(d) Histogram distribution of the disparity depth map

Figure 6.18: Cookies stereo image pair in the first column of the group camera array datasets.

As described earlier, the matching is between the upper and lower images on camera location 1 (upper) and 5 (lower). Figure 6.19(a) shows the sample of inter-view images, which is obtained using conventional interpolation method at $\beta=0.5$, which is located at camera location 3 in the first column in the multi-camera Cookies datasets. The SSIM image map for this approach is shown in Figure 6.19(b). The inter-view image at the same location synthesized by using MLVS at Level 2 is shown in Figure 6.19(c) with

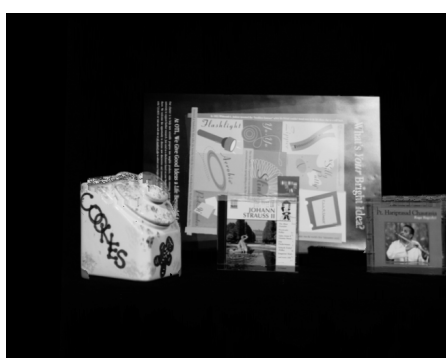
the SSIM map in Figure 6.19(d). In the SSIM map obtained with a conventional interpolation method, the darker region appeared in the boundaries of the objects that are not similar to the original image. Therefore, the SSIM is quite lower at 0.87. Meanwhile, the SSIM map by using MLVS Level 2 through the DILS algorithm has shown that the inter-view image gives a white region in almost all areas. However, some border are slightly darker indicating some errors eventhough the SSIM=0.92, which is higher than the result obtained in the conventional inter-view interpolation method. These errors occurred due to the occlusion and mismatching on the interpolation. The errors can be corrected by replacing the basic upper-to-lower stereo correspondence pixel matching with a better approach of matching algorithms. As discussed in the previous chapter, the DILS algorithm, which was adapted in the MLVS, can be integrated to any stereo matching algorithms.



(a) Synthesized image using the conventional interpolation method of inter-view image at $\beta=0.5$



(b) SSIM map for conventional method (SSIM=0.87)



(a) Synthesized image using MLVS with DILS of inter-view image at $\beta=0.5$



(b) SSIM map for MLVS method (SSIM=0.92)

Figure 6.19: The comparison of conventional interpolation and MLVS (Level 2) method. The sample of inter-view image synthesized at $\beta=0.5$ (camera 3 in column 1) for the third row of Cookies datasets and the structural similarity index map.

The full quantitative performance in PSNR and SSIM for the Level 2 of MLVS is shown in Table 6.3. The value of β also indicates the camera location between the upper-to-lower images, which is between camera 1 and 5. The table shown is for the MLVS in Level 2, which is along the column 1 for the Cookies multi-camera datasets. The PSNR and SSIM obtained for the MLVS method are higher compared to the conventional interpolation method. The mean squared error (MSE) is lower for every inter-view image location by using the MLVS. The graphs of PSNR for the full group of camera arrays in Cookies dataset of Level 2 are shown in Figure 6.20. Based on the results, it is shown that the MLVS outperforms the conventional inter-view interpolation method, which could provide similar and good inter-view images when are compared with the original images. This is illustrated by the SSIM graphs in Figure 6.21.

Table 6.3: Results comparison between conventional Linear Interpolation (LI) and MLVS method (for level 2) along the column 1 in a group of camera array.

Original image	Camera (β)	2(0.25)		3(0.5)		4(0.75)	
Upper-to-lower	Method	LI	MLVS	LI	MLVS	LI	MLVS
1_1 to 5_1	PSNR	36.34	38.65	36.29	38.76	35.93	39.43
	MSE	15.11	8.87	15.28	8.66	16.6	7.42
	SSIM	0.89	0.94	0.87	0.92	0.85	0.93

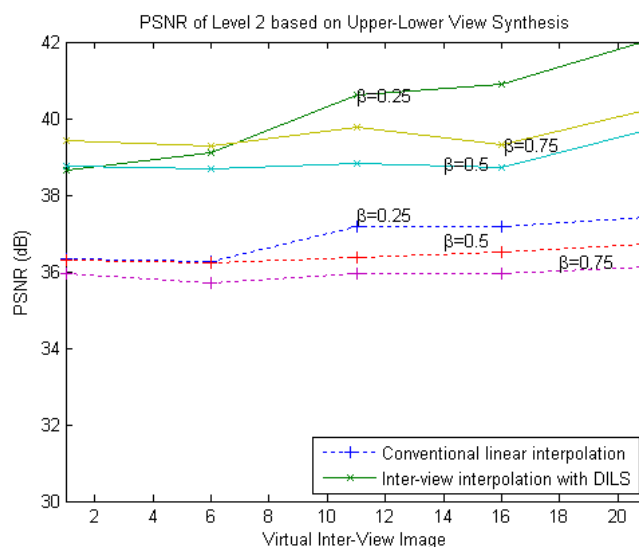


Figure 6.20: PSNR results for the Level 2 MLVS using Upper-to-Lower (UL) Matching DILS compared to the conventional linear interpolation PSNR of Cookies datasets.

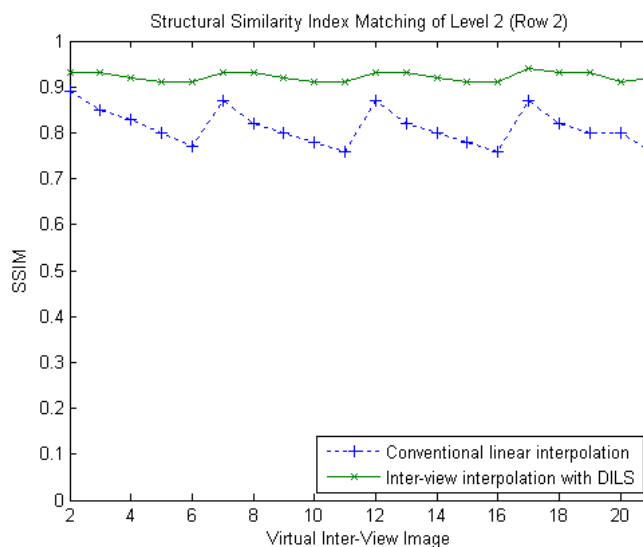


Figure 6.21: SSIM results of Level 2 MLVS compared with the SSIM results of the conventional linear interpolation for the first row of Cookies datasets.

6.5.3 MLVS: Level 3

The final stage of the MLVS is the level 3 matching and synthesis, which comprises the inter-view images in the middle of the group camera arrays. Level 3 of the MLVS algorithm, uses either of the image view synthesis obtained from Levels 1 or 2 in the previous stage. Table 6.4 shows results between the matching based on Left-to-Right (LR) and Upper-to-Lower (UL) of the Level 3 of the MLVS algorithm for the third row as shown in Figure 6.22. From the sample of results in the third row, it can be seen that the image view synthesis based on the left-to-right and upper-to-lower matching and synthesis do not show significant differences. However, the resulting PSNR and SSIM for the third level of MLVS are slightly lower compared to the results of Levels 1 or 2. This is expected since the inter-view image syntheses are generated using the synthesis image obtained from the Level 1 and 2. Given this fact, the inter-view image syntheses in Level 3 are showing acceptable quality, which do not contain too many pixel errors. The sample of image synthesis for the third row are obtained in Level 3 by using left-to-right and upper-to-lower MLVS are shown in Figure 6.23, with the SSIM map images.

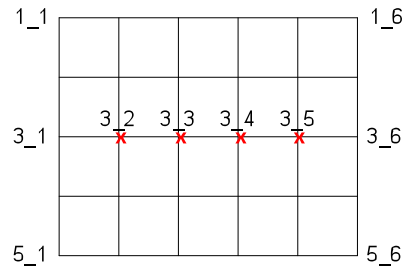


Figure 6.22: The inter-view image synthesis along the third row in the MLVS Level 3 for the Cookies multi-camera datasets (group camera array 1).

Table 6.4: Comparison results between the use of Left-to-Right (LR) matching and the use of the Upper-to-Lower (UL) matching in the MLVS method (Level 3) for the row 3 in a group of camera array.

Original image	Camera (β)	2(0.2)		3(0.4)		4(0.6)		5(0.8)	
		UL	LR	UL	LR	UL	LR	UL	LR
Row 3	Method								
GCA 1 (3_1 to 3_6) Col: 2-5	PSNR	35.53	35.54	36.42	36.48	37.38	37.3	37.1	37.08
	MSE	18.22	18.17	14.83	14.61	11.9	12.11	12.68	12.75
	SSIM	0.86	0.85	0.86	0.85	0.84	0.84	0.83	0.83



(a) MLVS Level 3 image view synthesis based on left-to-right (LR) matching



(b) SSIM map for the LR MLVS Level 3 (SSIM=0.85)



(c) MLVS Level 3 image view synthesis based on upper-to-lower (UL) matching



(d) SSIM map for the UL MLVS Level 3 (SSIM=0.86)

Figure 6.23: The comparison of the MLVS (Level 3) algorithm by using Left-to-Right (LR) and Upper-to-Lower (UL) matching approaches. The sample images are for the third row and third column (3, 3) of the Cookies multi-camera datasets.

The PSNR results obtained in the Level 3 of MLVS based on the left-to-right and upper-to-lower view synthesis are plotted in Figure 6.24. The three different rows (2, 3 and 4) are matched and synthesized by Left-to-Right (LR) or Upper-to-Lower (UL) images for the group of camera array 1 (based on the column 1 to 6). The PSNR and SSIM plots of MLVS for the whole range of row 3 in the Cookies multi-camera datasets are shown in Figure 6.25, which are compared to the conventional inter-view interpolation methods.

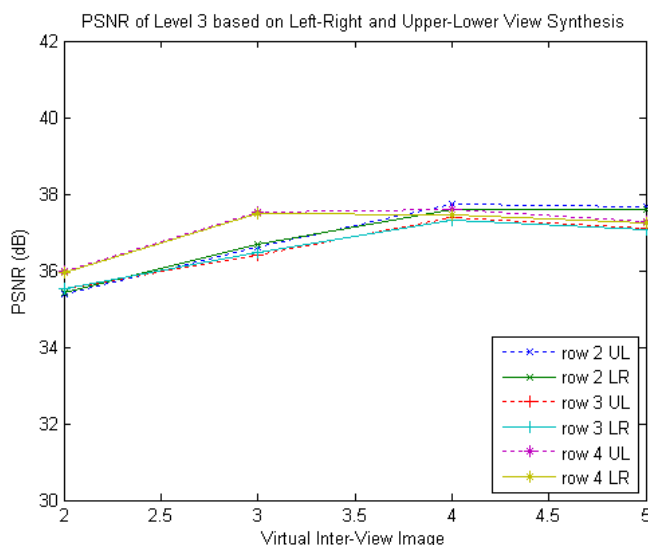
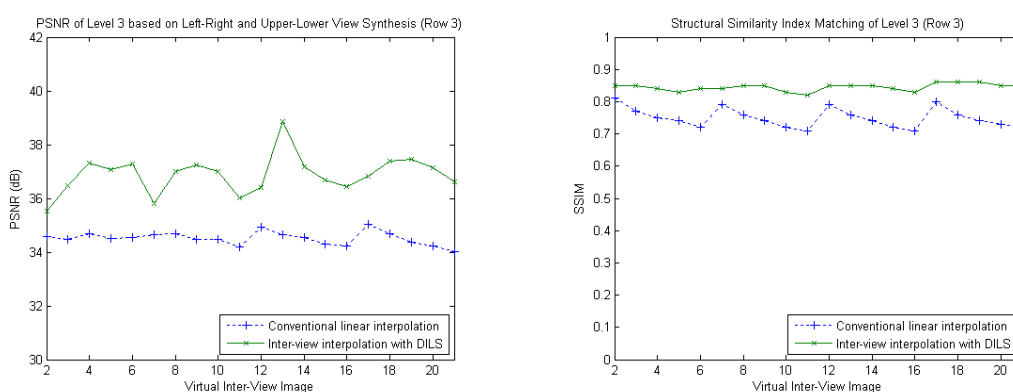


Figure 6.24: The comparison of PSNR for the Level 3 of the MLVS by using Upper-to-Lower (UL) and Left-to-Right (LR) Matching in the group of array 1.



(a) PSNR of inter-view synthesis image at Row 3

(b) SSIM index of inter-view synthesis image at Row 3 for conventional linear interpolation and DILS algorithm

Figure 6.25: Results on PSNR and SSIM for the MLVS at Level 3 compared with the conventional linear interpolation. The data sampled for the third row of Cookies multi-camera datasets.

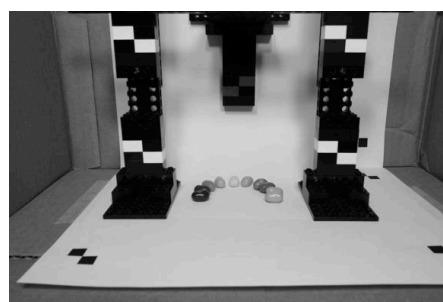
6.6 Lego Datasets Analysis

The next algorithm evaluation is based on the Lego multi-camera datasets originally with 17x17 views. For the MLVS experiment, only 5x5 real camera views of this datasets are used to reproduce the 17x17 views. The selected views for matching and synthesis are 1, 5, 9, 13 and 17 along the rows and columns. For this datasets, the camera arrays are divided into several groups of camera arrays. In the first group, it consists of 5x5 views for the first block of camera arrays, which is selected from the camera views along with the row 1-5 and column 1-5.

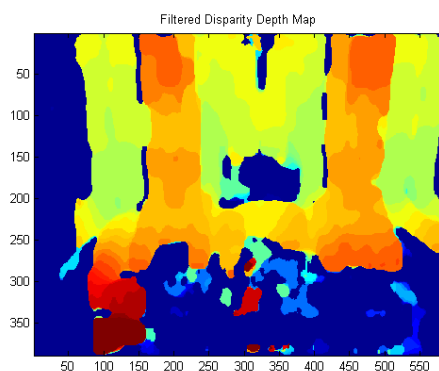
Figure 6.26(a) and Figure 6.26(b) show the original left image at camera 1 as the reference, and the right image at camera 5 as the target image in the first row of the camera arrays. Based on the stereo matching with the image pairs, the resulting disparity depth map is shown in Figure 6.26(c). The histogram distribution of the disparity depth map displayed in Figure 6.26(d) has a large amount for lower disparity range, because of the dark background in the disparity depth map and not much information on the texture.



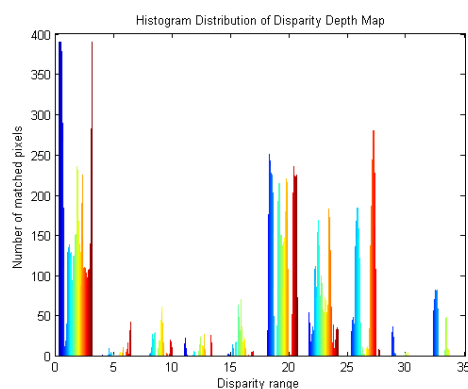
(a) Left image of Lego from camera 1_1



(b) Right image of Lego from camera 1_5



(c) Disparity depth map



(d) Histogram distribution of the disparity depth map

Figure 6.26: Lego stereo image pair in the first row of the group camera array datasets.

The disparity depth map is obtained from the left-to-right matching used the DILS algorithm in order to separate the layers of depth individually based on the histogram distribution. Within each layer, the new view interpolation image is created based on the disparity level and camera baseline ratio between left and right image pairs. Finally, the layers are flattened into a single image.

6.6.1 MLVS Level 1

In the first phase of MLVS, the image synthesis view contains holes along the occluded and boundary regions as shown in Figure 6.27(a). The errors in the inter-view image can be seen by comparing them with the original image. With the structural similarity metric, the SSIM is 0.76 (Figure 6.27(b)). This affects the image structure and it is not reliable for the second level of MLVS. Therefore, it is necessary for the MLVS image synthesis to be processed with a hole-fillings algorithm. Any holes within the image will be filled with boundary values. The new MLVS inter-view image is shown in Figure 6.27(c). The hole-filling procedure efficiently improved the MLVS inter-view image to SSIM=0.97 as shown in Figure 6.27(d).

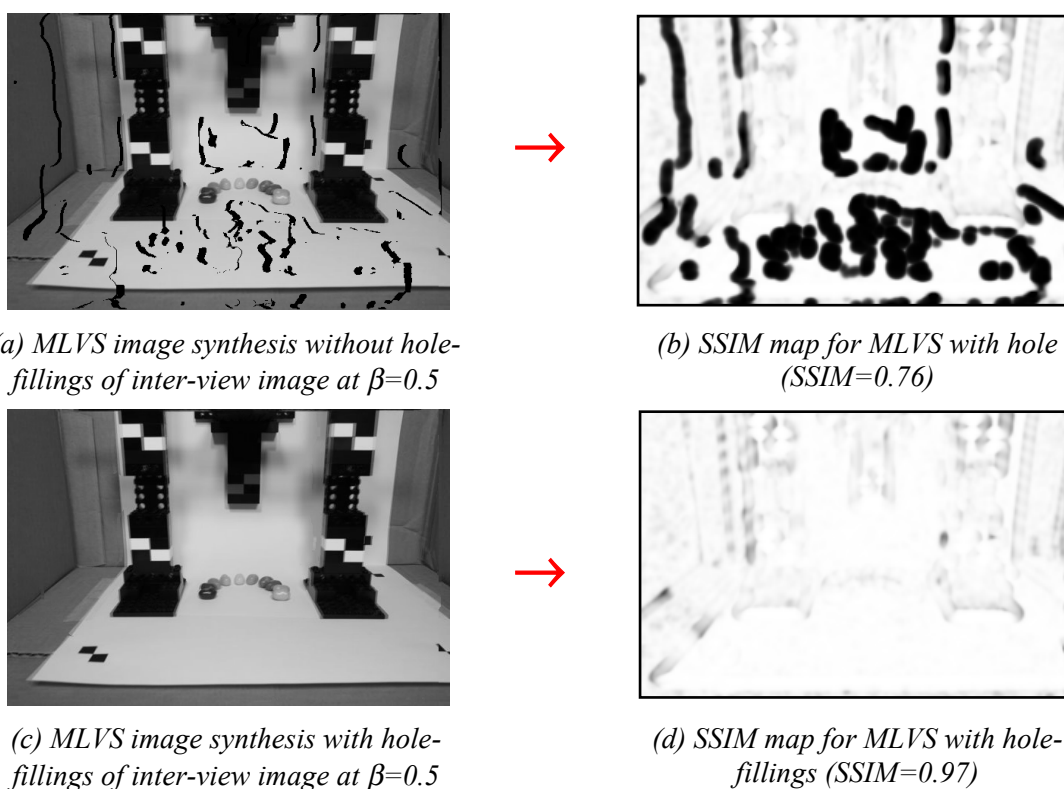


Figure 6.27: The comparison of MLVS (Level 1) without/with hole-filling algorithms. The sample of inter-view image and the structural similarity index map synthesized at $\beta=0.5$ (camera in row 1 column 3) for the Lego datasets.

Table 6.5 contains a sample of the results for the group of camera array 1, which is along the first and fifth rows in the Lego multi-camera datasets. The MLVS Level 1 is based on the left-to-right matching and synthesis, which is 1-5 for this example. The new image synthesis created with camera baseline ratio $\beta=0.25, 0.5, 0.75$ that represent the original camera location at column 2, 3 and 4 respectively in the first and fifth rows. The data has been obtained with the expanded matching and synthesis for the whole Lego datasets. The results illustrated as the graphs in Figure 6.28 for the PSNR and SSIM along the row 1 and 5.

The MLVS without the hole-filling techniques under-perform compared to the MLVS with the hole-filling in terms of the PSNR and the SSIM indices. While the MLVS without hole-filling fluctuates along the row, the improved version of MLVS with hole-filling provides a constant and good performance of PSNR and SSIM. The next section will discuss the MLVS in Level 2, which is the matching for the upper and lower images (or views).

Table 6.5: Comparison results of MLVS (Level 1) with and without using the hole-filling algorithms along the row 1 and 5 in a group of camera array.

Original image	Camera (β)	2(0.25)		3(0.5)		4(0.75)	
Left-to-right	Method	Hole	Hole-filled	Hole	Hole-filled	Hole	Hole-filled
1_1 to 1_5	PSNR	36.27	39.57	35.2	39.01	34.19	38.75
	MSE	15.36	7.17	19.67	8.16	24.8	8.66
	SSIM	0.76	0.97	0.76	0.97	0.71	0.96
5_1 to 5_5	PSNR	35.5	38.46	34.52	38.15	33.58	37.93
	MSE	18.34	9.28	22.98	9.96	28.53	10.48
	SSIM	0.74	0.96	0.74	0.96	0.69	0.95

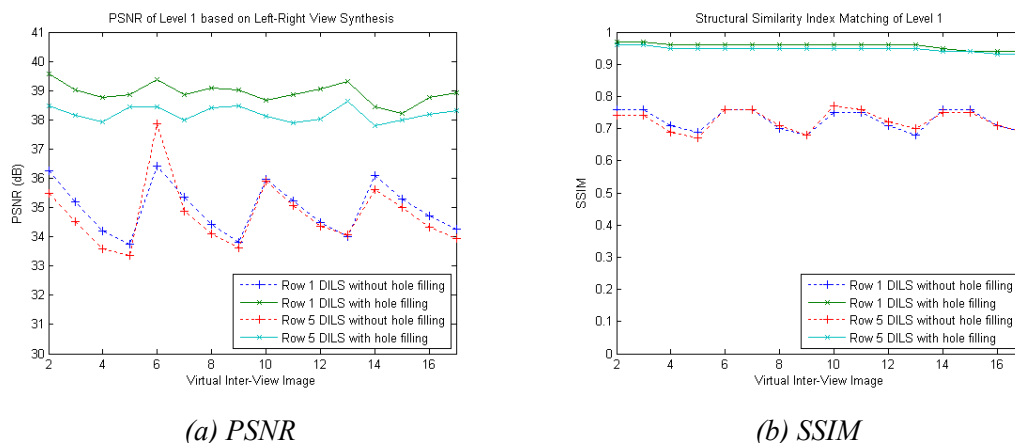


Figure 6.28: PSNR and SSIM of MLVS (Level 1) without and with using hole-filling algorithm in the image view synthesis at row 1 and 5 for Lego multi-camera datasets.

6.6.2 MLVS Level 2

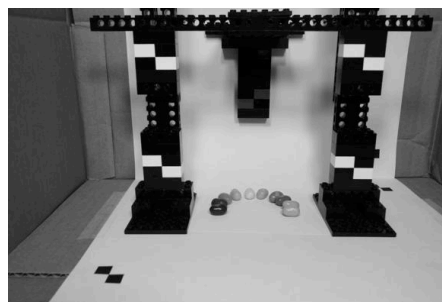
The MLVS Level 2 is based on upper-to-lower matching between the reference and target image pairs, where Figure 6.29(a) and Figure 6.29(b) are the original upper left image (camera 1 as the reference) and lower left image (camera 5 as the target image) respectively in the first column of camera arrays. The image pairs are matched with the upper-lower matching and the disparity depth map obtained is shown in Figure 6.29(c). The histogram distribution of the disparity depth map is displayed in Figure 6.29(d).

Similar to the MLVS Level 1, the disparity depth map is separated into several layers via the DILS algorithm to synthesize the inter-view interpolation image within the layers. When all the layers have been interpolated, the layers are flattened into a single image as the new view synthesis image. The samples of inter-view image synthesis with $\beta=0.5$ are shown in Figure 6.30. The $\beta=0.5$ represents the original camera location at column 1, row 3 with the upper-to-lower matching (1-1 to 5-1). In the matching and synthesis of (1-5 to 5-5), $\beta=0.5$ indicates the original camera location at the same row but different column, which is at 5.

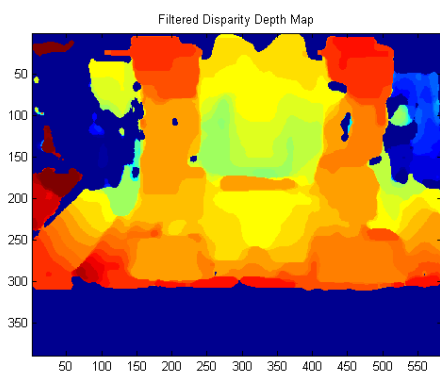
The inter-view image synthesis and SSIM in Figure 6.30(a) and Figure 6.30(b) shows that the image suffers with huge errors due to the holes. The holes appeared because of the occlusion and boundary region in the disparity depth map. The translation process of DILS also contributes to this hole. Therefore, the hole-fillings method is essential to improve the appearance of the image synthesis as shown in Figure 6.30(c). The new image provides a high SSIM index, which is 0.95 (in Figure 6.30(d)).



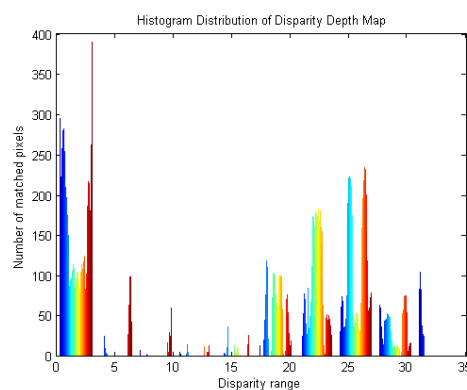
(a) Upper left image of Lego camera 1_1



(b) Lower left image of Lego camera 5_1



(c) Disparity depth map



(d) Histogram distribution of the disparity depth map

Figure 6.29: Lego stereo image pair in the first column of the group camera array datasets

Table 6.6 is almost similar to the results presented in Table 6.5. Instead of representing results for the left-to-right matching, this table is specifically for the matching between upper-to-lower images. The image matching is along the upper-left (1-1) and lower-left (5-1). It also provides the matching for the upper-right (1-5) and lower-right (5-5). This selection of data can be grouped as the group of camera array 1 in the Lego multi-camera datasets. The new image syntheses created with $\beta=0.25$, 0.5 and 0.75 represent the original camera locations at rows 2, 3 and 4 respectively.

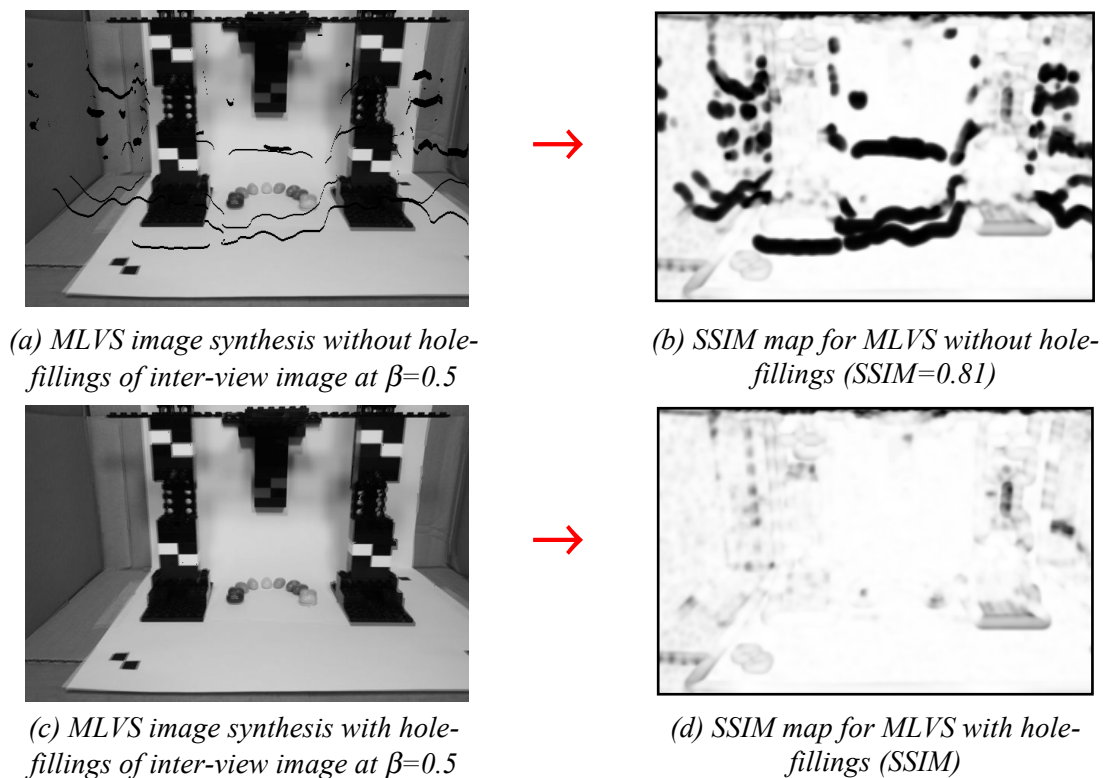


Figure 6.30: The comparison of MLVS (Level 2) with and without using the hole-filling method. The sample of inter-view image and the structural similarity index map synthesized at $\beta=0.5$ (camera in row 3 of column 1) for the Lego datasets

Table 6.6: Results comparison of MLVS (Level 2) with and without using the hole-filling methods along the column 1 and 5 in a group of camera array.

Original image	Camera (β)	2(0.25)		3(0.5)		4(0.75)	
Upper-to-lower	Method	Hole	Hole-filled	Hole	Hole-filled	Hole	Hole-filled
1_1 to 5_1	PSNR	36.49	37.59	35.9	38.11	34.91	37.38
	MSE	14.58	11.33	16.71	10.04	20.99	11.89
	SSIM	0.84	0.95	0.81	0.95	0.77	0.93
1_5 to 5_5	PSNR	36.17	37.29	35.54	37.78	34.75	37.39
	MSE	15.7	12.13	18.15	10.84	21.8	11.85
	SSIM	0.83	0.95	0.79	0.94	0.75	0.92

The full results along columns 1 and 5 for the Lego multi-camera datasets plotted in Figure 6.31 for the PSNR and SSIM. The graphs show the results between the MLVS with and without using the hole-filling technique. As illustrated in the graphs of the PSNRs and SSIMs, the results of MLVS with hole-filling outperform the image synthesized without the hole-filling techniques. In addition, the SSIM relatively constant more than 0.9 index, which indicates that the synthesized inter-view images are

similar to the original images. The next part is on the MLVS Level 3, which concludes the whole algorithm by obtaining the image synthesis in the middle part of the group of camera array.

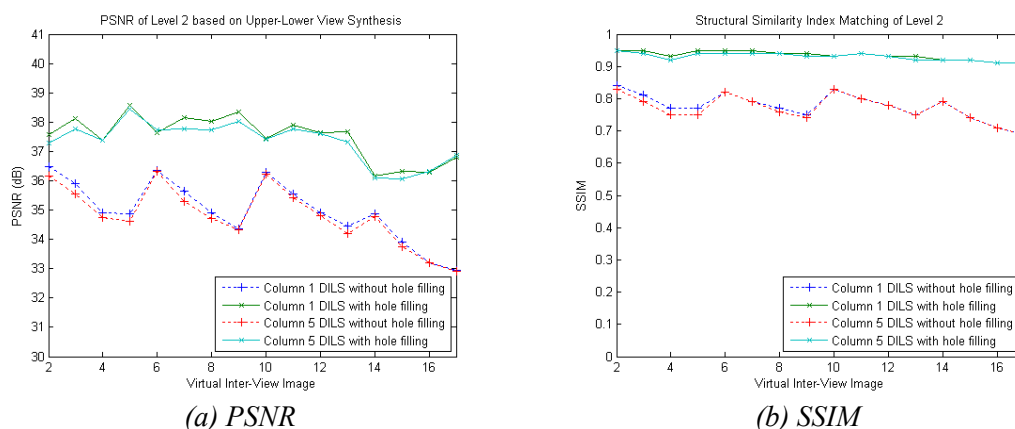


Figure 6.31: PSNR and SSIM of MLVS (Level 2) without and with hole-fillings in the image view synthesis at column 1 and 5 for Lego multi-camera datasets.

6.6.3 MLVS Level 3

Level 3 is the final part of the MLVS algorithm that corresponds to the synthesis of the middle image between the upper-lower and left-right images. The image synthesis in Level 3 is created based on the results obtained either from Level 1 or 2. Since there is no significant difference within the image quality either by using Level 1 or 2 results, the MLVS in the Level 3 uses left-to-right matching and synthesis for the DILS implementation. Figure 6.32(a) shows the sample of image synthesis based on $\beta=0.5$ for the camera at row 7, column 3. The image contains errors (holes) due to the occlusion and translation along the layers in the DILS algorithm. The SSIM for this image is quite low, which is 0.72 (as shown in Figure 6.32(b)). The image improved with the hole-filling technique as shown in Figure 6.32(c). Additionally, the SSIM index also increased to 0.92, which removes all the black regions as errors in the SSIM map (Figure 6.32(d)).

Table 6.7 presents the details on every inter-view image synthesis for $\beta=0.25, 0.5, 0.75$ (image that located in camera column 2, 3 and 4 respectively). The table shows a sample results for the synthesized images in rows 7 and 11. Generally, the error resulting from the MLVS without hole-filling techniques is high compared with the corrected images.

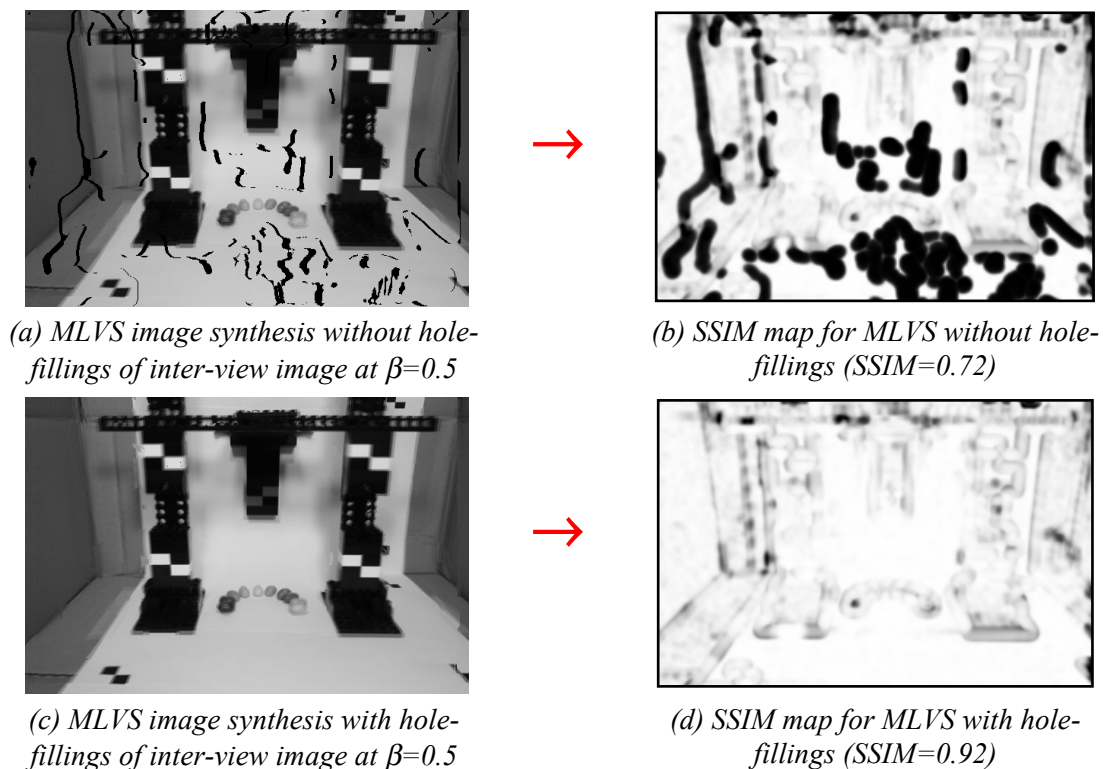


Figure 6.32: The comparison of MLVS (Level 3) with and without hole-fillings method. The sample of inter-view image and the structural similarity map synthesized at (camera in row 7 of column 3) for the Lego datasets.

Table 6.7: Results comparison of MLVS (Level 3) with and without the hole-filling methods along the row 7 and 11 in a group of camera array.

Original image	Camera (β)	2(0.25)		3(0.5)		4(0.75)	
Row	Method	Hole	Hole-filled	Hole	Hole-filled	Hole	Hole-filled
7_1 to 7_17	PSNR	34.9	37.17	34.05	36.89	33.21	36.81
	MSE	21.03	12.46	25.62	13.3	31.05	13.55
	SSIM	0.72	0.92	0.72	0.92	0.67	0.92
11_1 to 11_17	PSNR	34.82	36.86	34	36.5	33.28	36.65
	MSE	21.45	13.41	25.9	14.59	30.53	14.08
	SSIM	0.72	0.91	0.72	0.91	0.67	0.91

The full data of rows 7 and 11 for MLVS (Level 3) is plotted in the PSNR and SSIM graph as shown in Figure 6.33. It shows that the inter-view images corrected with hole-filling techniques provide good results compared to the images without the hole-fillings. The PSNR and SSIM of the corrected inter-view images are consistently stable from virtual camera location 2 to the maximum, which is 17. Meanwhile, the inter-view image synthesis with holes and errors fluctuate along the horizontal multi-camera arrays. It signifies that the hole-filling method is necessary for the MLVS in each level

of the implementation.

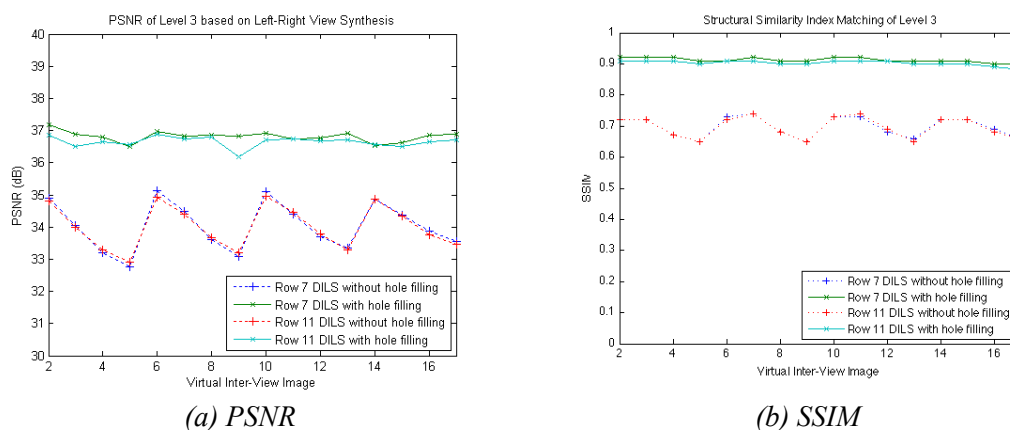


Figure 6.33: PSNR and SSIM of MLVS (Level 3) without and with using hole-filling algorithm in the image view synthesis at row 7 and 11 for Lego multi-camera datasets.

6.7 Results Evaluation with Different Baseline Images

In this evaluation, the Lego data is compared with different groups of arrays. Previously, the group consists of 5x5 views for the first block of camera arrays. What happens if we increase the disparity baseline, from 1-to-5 to 1-to-9? For this case, we conduct another experiment based on a group of camera arrays for 9x9 views in the first block for the similar Lego multi-camera datasets.

6.7.1 MLVS Level 1

Figure 6.34 presents the inter-view image syntheses between the matching of 1-to-5 and 1-to-9 baseline matching. The synthesized images between the two approaches are quite similar. However, the SSIM map indicates the MLVS by 1-to-9 baseline matching contained errors (as highlighted in the dark regions) with SSIM=0.88. Similar inter-view images synthesized by the 1-to-5 baseline matching yield better results, which is 0.95 for the SSIM index. Table 6.8 comprised the differences on the PSNR, MSE and SSIM for the baselines 1-to-5 and 1-to-9 in rows 1 and 9. The comparison was made for three inter-view image cameras located at column 3, 7 and 9. Based on these results, it shows that the increment of baseline matching reduces the accuracy of the image synthesis.

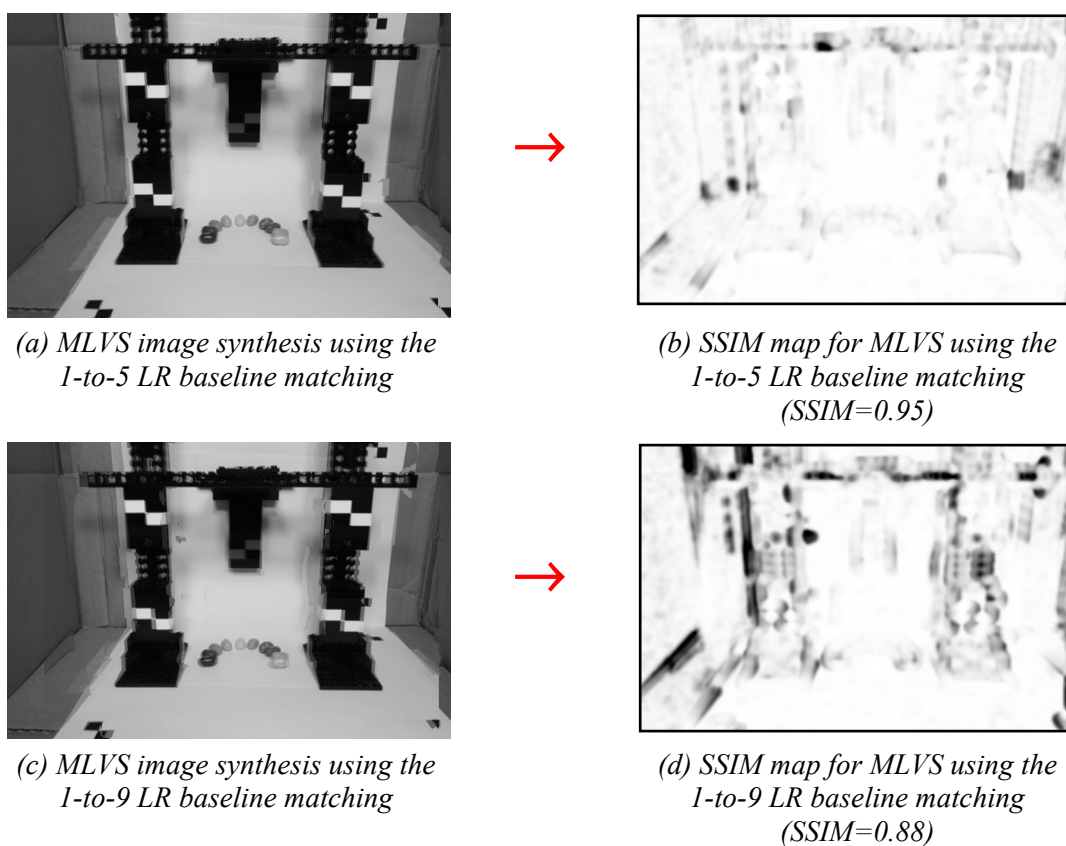


Figure 6.34: The comparison of the MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching. The sample of inter-view image and the structural similarity map synthesized at (camera in row 9 of column 3) for the Lego datasets.

Table 6.8: Comparison results of MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching along the rows 1 and 9

Original image	Camera (Column)	3		7		9	
		1-to-5	1-to-9	1-to-5	1-to-9	1-to-5	1-to-9
1	PSNR	39.01	36.18	38.87	36.09	38.85	35.75
	MSE	8.16	15.66	8.43	16	8.48	17.31
	SSIM	0.97	0.9	0.96	0.9	0.96	0.87
9	PSNR	37.81	35.44	38.07	35.45	38.06	35.64
	MSE	10.77	18.59	10.15	18.54	10.15	17.75
	SSIM	0.95	0.88	0.95	0.87	0.95	0.86

The complete results in the Lego multi-camera dataset for row 1, 9 and 17 plotted in the PSNR and SSIM graph (MLVS Level 1) are shown in Figure 6.35. This graph clearly indicates that the performance of the 1-to-9 baseline matching deteriorates in the PSNR compared to the 1-to-5 matching. With this PSNR output, the SSIM in the 1-to-5 matching is expected to outperform the 1-to-9 baseline matching as shown in Figure 6.35(b).

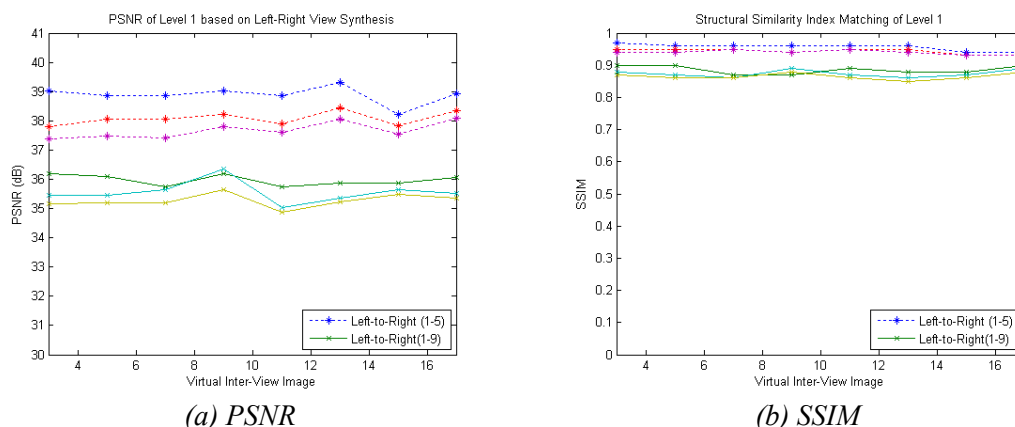


Figure 6.35: Comparison PSNR and SSIM results of the MLVS (Level 1) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching in the image view synthesis at rows 1, 9 and 17 for Lego multi-camera datasets.

6.7.2 MLVS Level 2

Figure 6.36(a) and Figure 6.36(c) show the image synthesis based on the upper-to-lower matching in MLVS Level 2 with the SSIM map images respectively (as shown in Figure 6.36(b) and Figure 6.36(d)). The inter-view image based on 1-to-9 matching consists of interpolation image errors in the upper part of the image. These errors are caused by the occlusion on the upper-to-lower matching by increasing the baseline. Moreover, the matching algorithm is based on fixed window area-matching which is unable to deal with the textureless region for the stereo pairs. The SSIM image map indicates the MLVS for baseline 1-to-9 in the Level 2 contained errors in the upper part of the image, with SSIM=0.84 (in Figure 6.36(d)). Table 6.9 shows the results based on the baseline matching of 1-to-5 and 1-to-9 for the upper-lower matching along columns of 1 and 9.

The full results on the MLVS (Level 2) for the columns 1, 9 and 17 are plotted in the graph of PSNR and SSIM as shown in Figure 6.37. This graph shows that the smaller baseline clearly provides better results compare with bigger step of baseline (1-to-9).

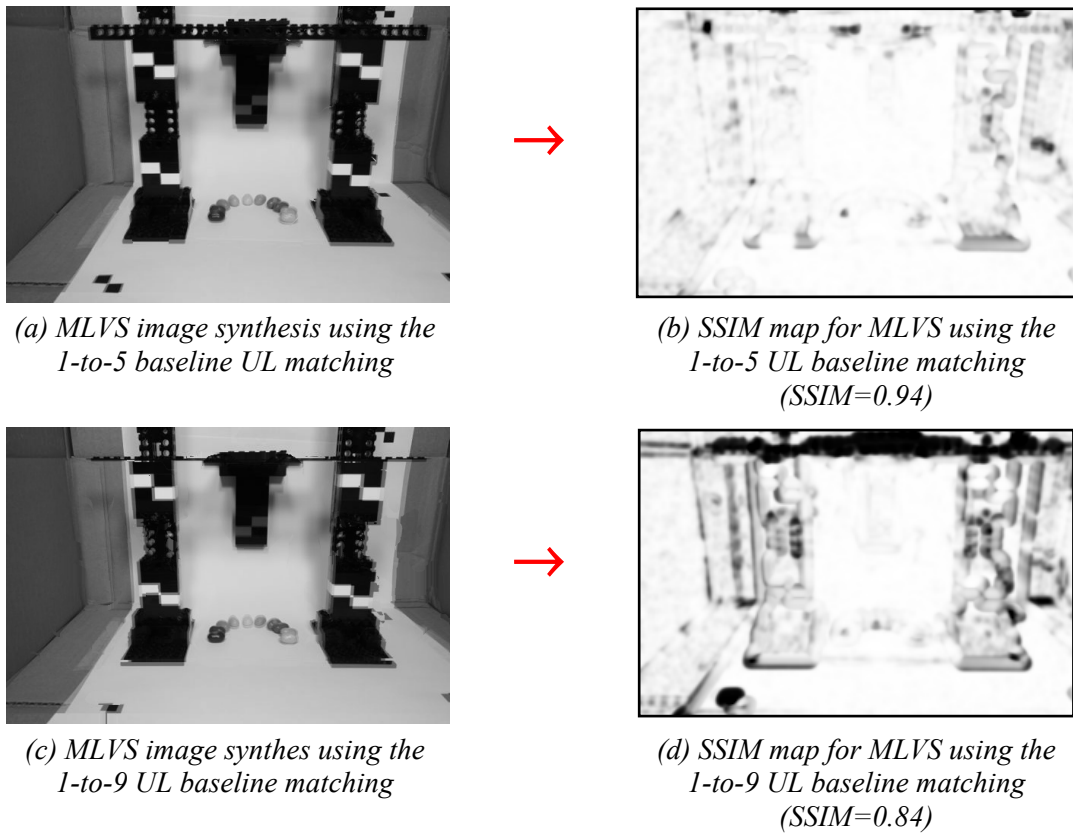


Figure 6.36: The comparison of the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 7 of column 1) for the Lego datasets.

Table 6.9: Comparison results of the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching along the columns 1 and 9

Original image	Camera (Row)	3		7		9	
Column	Matching	1-to-5	1-to-9	1-to-5	1-to-9	1-to-5	1-to-9
1	PSNR	38.11	35.56	38.56	35.64	38.16	36.91
	MSE	10.04	18.09	9.06	17.74	9.94	13.25
	SSIM	0.95	0.87	0.95	0.84	0.95	0.84
9	PSNR	37.71	35.46	38.31	35.79	37.74	36.57
	MSE	11.01	18.5	9.58	17.12	10.94	14.33
	SSIM	0.94	0.87	0.94	0.84	0.94	0.84

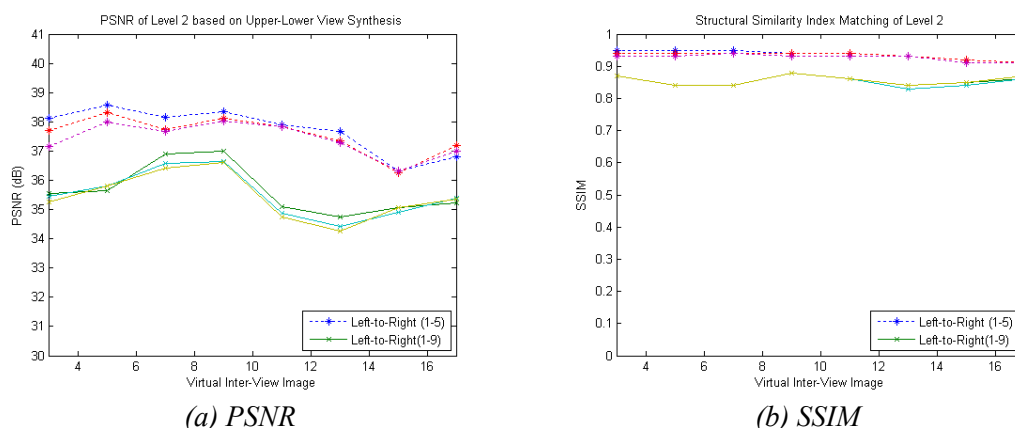


Figure 6.37: PSNR and SSIM results for the MLVS (Level 2) using the 1-to-5 and 1-to-9 Upper-Lower (UL) baselines matching at columns 1, 9 and 17 for Lego multi-camera datasets.

6.7.3 MLVS Level 3

The final evaluation on the MLVS is based on the last stage, which is the Level 3. As described earlier, the results obtained from Levels 1 and 2 will be used to synthesize the inter-view images in this level. If the created synthesis inter-view images contain errors, the new synthesis views in this level will suffer the same errors. For example, Figure 6.38 shows the results based on the 1-to-5 and 1-to-9 baseline matching in MLVS Level 3. While the image synthesis in 1-to-5 matching provide good results with SSIM=0.9, the image that synthesized in 1-to-9 suffers with several errors as shown in the SSIM map in Figure 6.38(d).

The sample of PSNR, MSE and SSIM for row 7 and 11 presented in Table 6.10, which comprises a comparison with the original image at cameras located in columns 3, 7 and 9. The complete results tabulated in the graphs of PSNR and SSIM are shown in Figure 6.39. The figure suggested that the image synthesis through small baseline matching provide better results compared with a larger baseline matching. For this case, the smaller baseline matching requires more cameras needed to capture the scene while producing virtual inter-view images. By contrast, the bigger baseline matching needs fewer cameras to produce the light-field images rendering and free-viewpoint video applications.

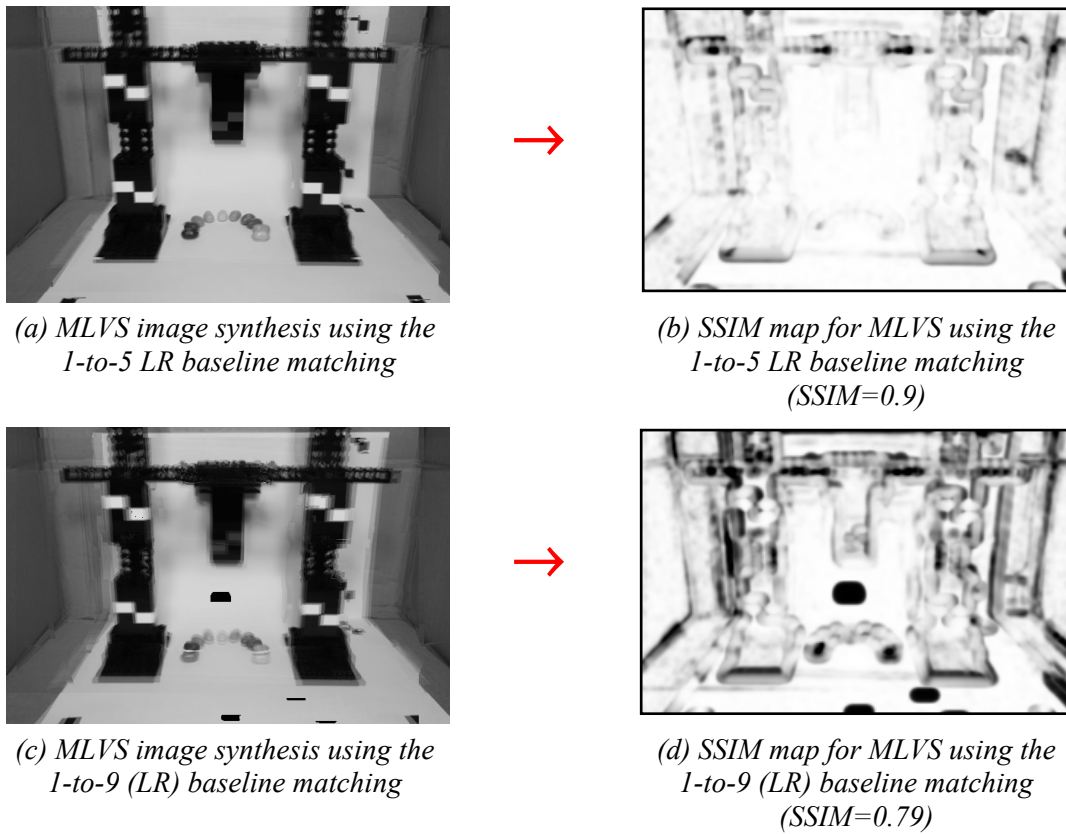


Figure 6.38: The comparison of the MLVS (Level 3) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching. The sample of inter-view image and the structural similarity index map synthesized at (camera in row 11 of column 7) for the Lego datasets.

Table 6.10: Results comparison of the MLVS (Level 3) using the 1-to-5 and 1-to-9 Left-Right (LR) baselines matching along the row 7 and 11

Original image	Camera (Column)	3		7		9	
		1-to-5	1-to-9	1-to-5	1-to-9	1-to-5	1-to-9
7_1 to 7_17	PSNR	36.89	35.18	36.51	34.66	36.83	34.16
	MSE	13.3	19.72	14.53	22.25	13.51	24.95
	SSIM	0.92	0.78	0.91	0.76	0.92	0.74
11_1 to 11_17	PSNR	36.5	33.76	36.55	33.66	36.73	33.47
	MSE	14.59	27.37	14.39	28.02	13.82	29.24
	SSIM	0.91	0.81	0.9	0.79	0.91	0.78

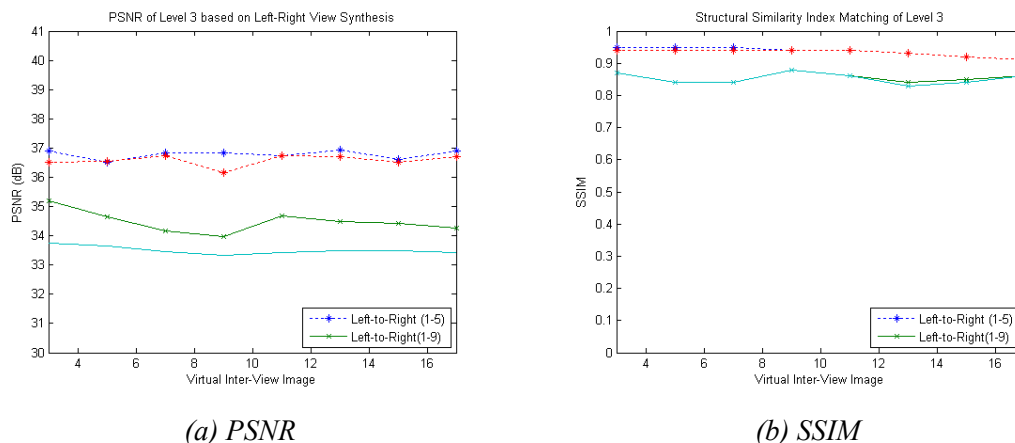


Figure 6.39: PSNR and SSIM results of the MLVS (Level 3) by using 1-to-5 and 1-to-9 Left-Right (LR) baselines matching in the image view synthesis at row 7 and 11 for Lego multi-camera datasets.

6.8 Conclusion

The aim of the MLVS algorithm is to create additional views along the camera arrays configuration in the light-field image rendering and free-viewpoint applications. With this approach, the number of used cameras can be reduced. The complexity of the handling multiple cameras, video storage, data compression and bandwidth can also be reduced. In this chapter we have presented a novel technique of inter-view synthesis based on the DILS algorithm for multi-camera array configurations. In this technique, novel multi-view syntheses were created based on a small number of cameras to create full dense multi-camera arrays. It reduces the number of required cameras to create dense images for FVV and light field imaging application. The principles of MLVS are to find the pixel correspondences and synthesis through three levels of matching and synthesis processes. This research also proposed an image view synthesis framework based on multi-view camera arrays configuration by matching and synthesizing inter-view image algorithms horizontally and vertically.

The first level of MLVS is to find the image synthesis through left-to-right matching along the horizontal plane. Second level of MLVS is to synthesis image based on upper-to-lower matching in the vertical camera arrays configuration. And lastly, the Level 3 is to create the inter-view images in the middle of the cameras. Based on experimental results, the inter-view image synthesis through MLVS provides good results compared with conventional inter-view interpolation method. The additional hole-fillings

technique in the DILS algorithm is quite essential to remove errors in the inter-view image synthesis.

The performance of the MLVS algorithm was tested on the Stanford Multi-Camera Array datasets and yielded high PSNR and SSIM index values. The quality of synthesized multi-view images is very impressive and satisfactory for free-viewpoint video and light-field imaging applications. The proposed method gives comparable performance to the conventional inter-view interpolation. In the experiments, it was demonstrated that it is possible to efficiently synthesize realistic new views even from bigger baseline matching that need fewer cameras. However, it can be performed with better results by using small baseline matching, where the reference and target image pair is not too far apart. The new structures and design are shown to offer improved performance and provide additional views with fewer cameras arrangement compared to the conventional high volume camera configurations for free-viewpoint video acquisition and light field imaging applications.

Chapter 7

Conclusion and Summary

7.1 Conclusions Overview

This thesis investigated image and video processing in multi-view image synthesis with applications to 3D vision and free-viewpoint video. The main focus of this work was to propose and create inter-view images that locate in the virtual viewpoints between source image viewpoints for 3D video systems.

This thesis presented the evolution of multi-view systems that have been an active research area in the field of computer vision from the basic stereoscopy to multi-view imaging. The number of publications on stereo increased with the online submission state-of-the-art algorithm in Middlebury Stereo Page that provides some common benchmark datasets and evaluation systems. The researchers can utilize the datasets and examine their proposed methods objectively and universally. Most of the stereo algorithms aim to produce accurate disparity maps and faster executions. Existing inter-view synthesis algorithms emphasize mostly in disparity estimation obtained from the stereo matching algorithms. The core components of stereo matching algorithms are highlighted and include matching cost computation, cost aggregation, disparity computation optimization and disparity refinement. The thesis investigated a few well-known techniques and algorithm published around these components, highlights the challenges faced and proposed some solutions to these challenges.

The development in display technology allows new applications to expand such as 3DTV, free-viewpoint TV, multi-view display and so on. The 3D video technologies enable various applications that can be integrated into a single 3D video system. However, efficient data compression is important in the multi-view video due to the huge amount of data for storage and transmission. This thesis discussed multi-view video coding that exploits redundancies that exist among multi-view images. Multi-

view imagery requires the contents to be captured by many cameras. The number of cameras or views on the multi-camera system can be reduced by the image view synthesis. Thus, the complexity of multi-camera configuration and high cost can be reduced with image view synthesis.

To develop techniques for image view synthesis, the fundamental concepts of image geometry plays an important role. This thesis describes the concepts of a 3D scene representation, view synthesis algorithms and image-based rendering. Due to the complexity of image synthesis for real scenes, this thesis presents techniques based on layered depth map.

A novel algorithm of Depth Image Layer Separation (DILS) algorithm was presented in this thesis. DILS focussed on image synthesis based on disparity depth map layers representation of stereo image pairs. The algorithm presented and exploited inter-view correlations to generate intermediate view synthesis images that locate in the virtual viewpoints between source image viewpoints. DILS featured a new paradigm that comprises a method to select interesting locations in the image based on depth analysis. It also represents a new image representation that allows description of the objects or parts of an image without the need of segmentation and identification. Image view synthesis can reduce the complexity of multi-camera array configuration for 3D imagery and free-viewpoint applications. The performance of the algorithm was tested on the Middlebury Database and yielded high PSNR and SSIM values. The quality of synthesized multi-view images is satisfactory for free-viewpoint applications. The proposed method gives comparable performance to the conventional inter-view interpolation. In the experiments, it was demonstrated that it is possible to synthesize realistic new views efficiently even from inaccurate depth information through the DILS algorithm. However, it can be performed using either simple or sophisticated stereo matching techniques to synthesize better quality inter-view images.

This thesis presented another novel technique of a post-processing stereo matching algorithm, which is the disparity refinement based on the DILS algorithm, known as Depth Layers Refinement (DLR). The core of the algorithm relies on a layer separation process based on different disparity ranges and the mapping between the layers and the segmented reference image. Each disparity layer and segmented reference image undergo a morphological process based on cross-path points from the mapping procedure. With this approach, the uniform areas and repetitive patterns can be grouped

in a single layer of depth. A comparative analysis of existing stereo matching algorithms with the proposed algorithm is conducted based on the common benchmark datasets and evaluation system in the Middlebury Stereo Vision Page. The results shown validate the proposed algorithm, where the disparity map obtained by a fixed window SAD improved significantly after being processed with the DLR algorithm. The algorithm is shown to refine the disparity depth maps efficiently and improves the pixel matching between the two images with a basic similarity metric. The main difference between this algorithm and other sophisticated algorithms is that this approach refines the disparity map and detects the depth discontinuities based on the layers separation.

Finally, this thesis presented a novel technique of image synthesis for multi-view images that can be used in 3D vision and free-viewpoint applications. This method is known as the Multi-Level View Synthesis (MLVS). MLVS exploits the advantage of the new inter-view interpolation algorithms based on the DILS algorithm by extending stereo to multiple camera configurations. In this technique, a novel multi-view synthesis is created based on a limited number of cameras for sparse camera arrays by finding the pixel matching correspondences and synthesis through three stages (levels) of processing. The first stage identifies the pixel correspondences and view synthesis based on the left-right image pair. Following this, the view synthesis image in the second stage is based on the upper-lower image pairs through vertical matching. The third stage used the obtained output in the first or second stages for the new inter-view synthesis to create full virtual multi-camera array image views. This approach reduces the number of cameras required to create dense images in the light field imaging applications. The new structures and design were shown to offer improved performance and provide additional views with fewer cameras in comparison to the conventional high volume camera configurations for free-viewpoint video acquisition.

To conclude, this thesis presented a number of stereo matching and inter-view image synthesis techniques for 3D vision and free-viewpoint applications. The techniques intended to recreate the real scene images from the stereo pair and multi-camera arrays scenarios. Due to the demand of dense images to be used in multi-view imaging displays, huge amount of cameras are required to feed the content creation. With the proposed algorithms, the number of cameras used can be reduced, which can reduce the camera costs, multi-camera complexity configuration problems, large data storage and

transmission requirements. The thesis also bridges a gap between efficient algorithm and practical implementation by highlighting problem issues in the stereo matching and image synthesis algorithms.

7.2 Future Work

The presented DILS was developed and tested with only static images, while in the 3D video and free-viewpoint environment most of the views involve motion. The DILS algorithm can be extended to cater for camera motions as well. To perform this task the disparity maps and interpolation of view synthesis for every frame are needed. In most of free-viewpoint video creations from multiple camera systems, cameras are assumed to be fixed. This is guaranteed by mounting the cameras on poles or tripods for the duration of capturing. The calibration is done only before the video acquisition starts. During video acquisition, cameras cannot be moved, zoomed or rotated. The field of view of each camera in these systems must be wide enough to cover the area in which the object moves. Therefore, the DILS could be adapted for synthesizing free-viewpoint videos in a natural scene from uncalibrated pure rotating and zooming cameras. The free-viewpoint video can be synthesized based on the reconstructed visual hull through DILS. The new view synthesis approach for a large baseline of DILS, where the cameras are separated sparsely is also an interesting field of research to be explored.

The proposed techniques of realising virtual images prove the fundamental concept of using disparity maps from stereo matching for image synthesis algorithm. However to investigate the best suitable options in specific environments and 3D computer vision applications are left as future work. Another limitation of the presented DILS algorithms lies in the assumption that the scene can be approximated by a set of layers and that is limited by the baseline distance between the source images. The DILS also assumed layers can be identified where the depth of the objects are not closely located for the whole image pairs. A more sophisticated model to separate the layers and interpolation model remains a topic for further work. Furthermore, we plan to extend our layered approach from stereo images to stereo videos of moving scenes. This will explore possibilities to employ layer extraction, segmentation and composition for both stereo matching and inter-frame motion tracking in order to develop techniques for video synthesis based on combined depth and motion information.

Although the proposed inter-view image synthesis techniques performed well for datasets it is desired that these techniques should be ported onto some hardware platform for real-time implementation. It is suggested that the algorithm presented can be ported to general-purpose processor or DSP-based embedded platform. It is also anticipated that the algorithms presented are optimized for hardware implementation for faster execution and memory requirement as the basic stereo matching cost mainly on SAD similarity metric. As suggested and proved by Stefano [85], the fixed window computation can be optimized through memory iteration.

The technique of refining the disparity map with DLR algorithm could be improved by selecting a better technique on the edge boundary, such as salient detection algorithm. The DLR and DILS can also be extended to include other motion and visual descriptors such as motion trajectory descriptor, camera motion descriptors and shape descriptors for video processing applications. The implementation of three proposed algorithms in this thesis is also recommended by using the TOF depth camera and Microsoft's Kinect system as another possible implementation.

Appendix A: Camera Calibration

Camera calibration involves the estimation of both extrinsic and intrinsic camera parameters. The first stage of camera calibration procedure is to establish correspondences between 2D points in the image and the 3D points on the checkerboard, known as point-correspondences. The basic steps for camera calibration can be summarized as follows:

- i. Capture N (at least 3) images with the checkerboard pattern and estimate point-correspondences.
- ii. Estimate and correct the radial lens distortion.
- iii. For each captured image, calculate the N homography transform.
- iv. Using the N homography transform, calculate the intrinsic and extrinsic parameters.
- v. Refine the calculated camera parameters.

A.1 Camera Parameters

A.1.1 Internal Camera Parameters

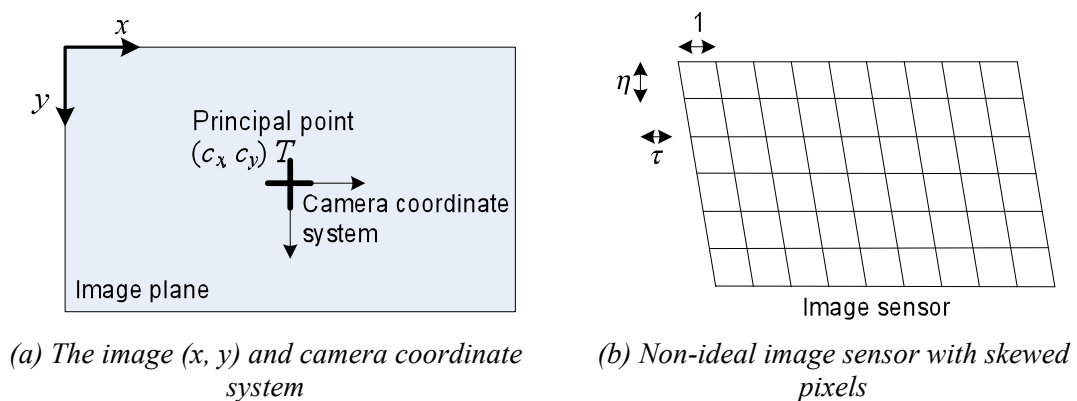


Figure A.1: Pixel coordination system

The internal camera parameters consist from the perspective projection of the camera, principal-point offset, image sensor characteristics, radial lens distortion and tangential lens distortion. For the principle-point offset, most of the current imaging systems

define the origin of the pixel coordinate system at the top-left pixel of the image. However, the origin of the pixel coordinate system corresponds to the principal point (c_x, c_y) , located at the centre of the image as shown in Figure A.1(a) [37].

Two new parameters, c_x and c_y , are introduced to model a possible displacement (away from the optic axis) of the centre of coordinates on the projection screen. The result is that a relatively simple model in which a point Q in the physical world, whose coordinates are (X, Y, Z) , is projected onto the screen at some pixel location given by (x_{screen}, y_{screen}) in accordance with the following equations [37],

$$x_{screen} = f \left(\frac{X}{Z} \right) + c_x, \quad y_{screen} = f \left(\frac{Y}{Z} \right) + c_y \quad (\text{A.1})$$

A necessary conversion of coordinates done by using homogeneous coordinates where the principal-point position can be integrated into the projection matrix. The perspective projection equation becomes as [37],

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (\text{A.2})$$

In some particular cases, pixels of the image sensor are not square, which are skewed, depending on the camera manufacturer. The pixel grid may be skewed due to an inaccurate synchronization of the pixel-sampling process. The imperfection of the imaging system can be taken into account in the camera model, using the parameter η (pixel aspect ratio) and τ (skew of the pixels as shown in Figure A.1(b)). The projection mapping updated as [37],

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & \tau & c_x & 0 \\ 0 & \eta f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [K \quad 0_3] P \quad (\text{A.3})$$

with $P = (X, Y, Z, 1)^T$ being a 3D point defined with homogeneous coordinates. In practice, with recent digital cameras, it can be assumed that the pixels are squared ($\eta=1$) and non-skewed ($\tau=0$). The projection matrix that includes the intrinsic parameters is denoted as K . The all zero element vector is denoted by 0_3 . In this case Equation (A.3)

will be used.

As described earlier, the relation that maps to the points Q_i in the physical world with coordinates (X_i, Y_i, Z_i) to the points on the projection screen with coordinates (x_i, y_i) known as the projective transform. In this case, the image plane is the projective space and it has two dimensions, therefore it can be represent the points on that plane as three-dimensional vectors $q = (q_1, q_2, q_3)$. Recalling that all points having proportional values in the projective space are equivalent, the actual pixel coordinates can be recovered by dividing through by q_3 . This allows to arrange the parameters that define the camera (f , c_x and c_y) into a single 3-by-3 matrix, which known as the camera intrinsic matrix. The projection of the points in the physical world into the camera is now summarized by the following form [37],

$$q = MQ, \quad \text{where} \quad q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (\text{A.4})$$

By multiplying this equation, $w = Z$. And since the point q is in homogenous coordinates, the equation divide through by w (or Z) in order to recover the earlier definitions. With the ideal pinhole, it is useful model for some of the three-dimensional geometry of vision. However, in practice such arrangement would make for very slow imaging. For a camera to form images at a faster rate, a lot of light over a wider area and focus that light to converge at the point of projection must be gathered. In order to accomplish this, a lens is used. A lens can focus a large amount of light on a point to provide fast imaging, but it comes with distortions.

A.1.2 Lens Distortion

Theoretically, it is possible to define a lens with no distortions. However, in practice no lens is perfect. During the manufacturing process, the lenses undergo spherical lens rather than ideal parabolic lens. It is also difficult to mechanically align the lens and imager exactly. In this section, two main lens distortions will be described that are radial and tangential distortions. Radial distortions arise as a result of the shape of lens, whereas tangential distortions arise from the assembly process of the camera as a whole.

The lenses of real cameras often noticeably distort the location of pixels near the edges of the imager. This bulging phenomenon is the source of the barrel or fish-eye effect.

Figure A.2 [44] shows how radial distortion occurs. The radial lens distortion appears more visible at the image edges, where the radial distance is high. The rays further from the centre of the lens are bent too much compared to rays that pass closer to the centre. Thus, the sides of a square appear to bow out on the image plane and will caused the straight lines to be mapped curved lines. This is also known as barrel distortion. Barrel distortion is particularly noticeable in cheap web cameras but less apparent in high-end cameras, where a lot of effort is put into fancy lens systems that minimize radial distortion.

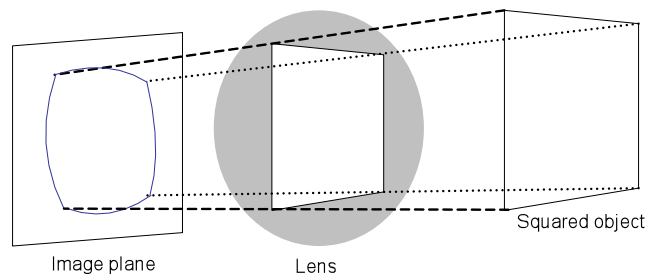


Figure A.2: Radial distortion that causes straight lines to be bended

For radial distortions, the distortion is 0 at the centre of the imager and increases as move toward the periphery. In practice, this distortion is small and can be characterized by the first few terms of a Taylor series expansion around $r = 0$. For cheap web cameras, generally the first two of such terms will be use; the first of which is conventionally called k_1 and the second k_2 . For highly distorted cameras such as fish-eye lenses, the third radial distortion term, k_3 can be used. In general, the radial location of a point on the imager will be rescaled according to the following equations [44]:

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (\text{A.5})$$

Here, (x, y) is the original location on the imager of the distorted point and $(x_{corrected}, y_{corrected})$ is the new location as a result of the correction.

The second largest common distortion is tangential distortion. This distortion is due to manufacturing defects resulting from the lens not being exactly parallel to the imaging plane as illustrated in Figure A.3 [44]. Tangential distortion is minimally characterized by two additional parameters, p_1 and p_2 , that derived from [44]:

$$\begin{aligned} x_{corrected} &= x + [2p_1 y + p_2(r^2 + 2x^2)] \\ y_{corrected} &= y + [p_1(r^2 + 2y^2) + 2p_2 x] \end{aligned} \quad (\text{A.6})$$

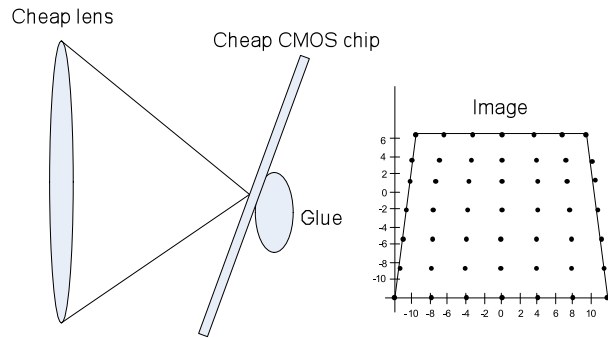


Figure A.3: Tangential camera lens distortion

Thus, in total there are five distortion coefficients that required. Since all five are necessary in most of the calibration process and coding routines that use them, they are typically bundled into one distortion vector. This is just a 5-by-1 matrix containing k_1 , k_2 , p_1 , p_2 and k_3 . Besides that, there are many other kinds of distortions that occur in imaging systems, but they are typically having lesser effects than radial and tangential distortions.

A.1.3 External Camera Parameters

The extrinsic parameters indicate the external position and orientation of the camera in the 3D world. For each image, the camera takes of a particular object, the pose of the object relative to the camera coordinate systems can be described in terms of a rotation and a translation as shown in Figure A.4 [37, 44, 45], which reproduced from Figure 2.4 in Chapter 2.

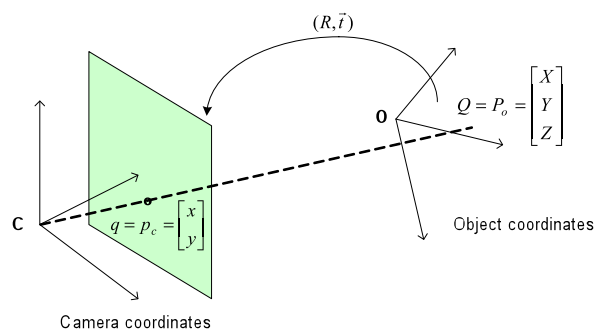


Figure A.4: Converting from object to camera coordinate systems

In general, a rotation in any number of dimensions can be described in terms of multiplication of a coordinate vector by a square matrix of the appropriate size. A rotation is equivalent to introducing a new description of a point's location in a different coordinate system. Rotating the coordinate system by an angle θ is equivalent to counter-rotating the target point around the origin of that coordinate system by the same

angle, θ . The representation of a two-dimensional rotation as matrix multiplication is shown in Figure A.5. Rotation in three-dimensions can be decomposed into a two-dimensional rotation around the x , y and z axis in sequence with respective rotation angles ψ , φ and θ . The result is a total rotation matrix R that is given by the product of the three matrices $R_x(\psi)$, $R_y(\varphi)$ and $R_z(\theta)$ [44], where

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \quad (\text{A.7})$$

$$R_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} \quad (\text{A.8})$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.9})$$

Thus, $R = R_z(\theta), R_y(\varphi), R_x(\psi)$.

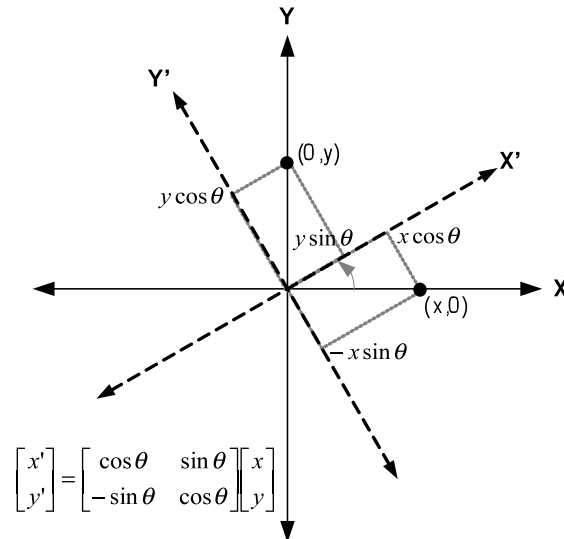


Figure A.5: Rotating points by θ

The rotation matrix R has the property that its inverse is its transpose: hence, $R^T R = R R^T = I$, where I is the identity matrix consisting of 1s along the diagonal and 0s everywhere else.

The translation vector is how a shift can be represented from one coordinate system to another system whose origin is displaced to another location. The translation vector is just the offset from the origin of the first coordinate system to the origin of the second coordinate system. Thus, to shift from a coordinate system centred on an object to one centred at the camera, the appropriate translation vector is simply $T = origin_{object} - origin_{camera}$. With reference to Figure A.5, a point in the object (or world) coordinate frame P_o has coordinate P_c in the camera coordinate frame [44]:

$$P_c = R(P_o - T) \quad (A.10)$$

Combining this equation for P_c above with the camera intrinsic corrections will form the basic system of equations to solve in the camera calibration parameters.

Appendix B: Predictive Coding

B.1 Hierarchical Encoding

Two basic types of coded pictures are possible: intra and inter pictures. Intra pictures are coded independently of any other image. Meanwhile, inter pictures depend on one or more reference pictures that have been encoded previously. By design, an intra picture does not exploit the redundancies among the multi-view images. But an inter picture is able to make use of these redundancies by choosing one or more reference pictures and generating a motion and/or disparity compensated image for efficient predictive coding. The basic ideas of motion-compensated predictive coding are presented in [178] by Flierl. This technique can be expanded to hierarchical encoding which offers not only temporal multi-resolution representation but also high coding efficiency.

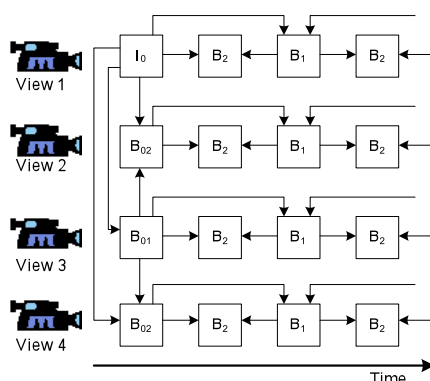


Figure B.1: Hierarchical encoding of a matrix of pictures (MOP) with bi-predictive pictures

Figure B.1 depicts a possible hierarchical encoding of a MOP with $N=4$ image sequences, each comprising $K=4$ temporally successive pictures. Each MOP is encoded with one intra picture and $NK-1$ bi-predictive pictures. Each MOP is decomposed in view direction at the first time instant only. Therefore, the sequences have view decompositions at every K^{th} time instant. The intra picture I_0 in each MOP represents the lowest view resolution. The next view resolution level is attained by including the bi-predictive picture B_{01} . The highest view resolution is achieved with the bi-predictive B_{02} . Then, the reconstructed N view images at every K^{th} time instant are

now used as reference for multi-resolution decompositions with bi-predictive pictures in temporal direction. The decomposition in view direction at every K^{th} time instant represents already the lowest temporal resolution level. The next temporal resolution level is attained by including the predictive picture B_1 . The highest temporal resolution is achieved with the predictive pictures B_2 . Thus, hierarchical encoding of each MOP with bi-predictive pictures generates a representation with multiple resolutions in time and view direction [70].

The concept of hierarchical B pictures was introduced by Schwarz [179]. A typical hierarchical prediction structure with three stages of a dyadic hierarchy is depicted in Figure B.2. The first picture of a video sequence is intra-coded as IDR picture and so-called key pictures are coded in regular intervals. A key picture and all pictures that are temporally located between the key picture and the previous key picture are considered to build a group of pictures (GOP), as illustrated in Figure B.2 for a GOP of eight pictures.

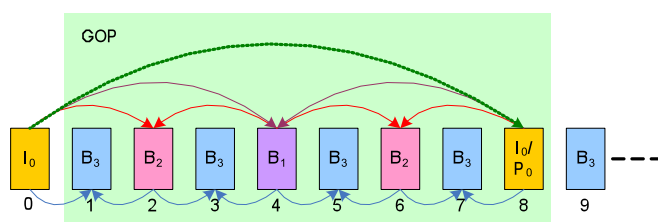


Figure B.2: Hierarchical reference picture structure for temporal prediction

The hierarchical B pictures can easily be applied to multi-view video sequences as illustrated in Figure B.3 for a sequence with 8 cameras and a GOP length of 8, where S_n denotes the individual view sequences and T_n the consecutive time-points. To allow synchronization and random access, all key pictures are coded in intra mode. Simulcast coding with hierarchical B pictures will be used as a reference to compare highly efficient temporal prediction structures with prediction structures that additionally use inter-view prediction.

Video coding based in intra mode, where no reference pictures are available for prediction, results in considerable higher bit rates than in inter prediction. By replacing intra-coded (or I) pictures with inter-coded (P or B) pictures has the potential to achieve a substantial coding gain. Adapting this approach to the multi-view video example in Figure B.3 leads to the prediction scheme in Figure B.4.

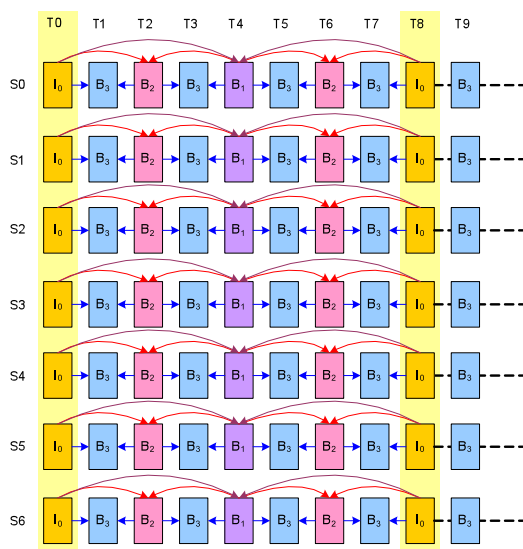


Figure B.3: Temporal prediction using hierarchical B pictures

The prediction structure of the first view S_0 remains the same and is called base view, as it is identical to the simulcast prediction structure with hierarchical B pictures for temporal prediction only. However, for the other views, all intra-coded key pictures are replaced by inter-coded pictures using inter-view prediction. For the remaining pictures of each GOP, the prediction structure does not change and remains to be temporal prediction with hierarchical B pictures. The analysis of temporal and inter-view prediction efficiency by Merkle [71] indicates that using temporal and inter-view reference frames at the same time improved the coding efficiency. In order to exploit all statistical dependencies within a multi-view test data set, inter-view prediction can be extended to non-key pictures.

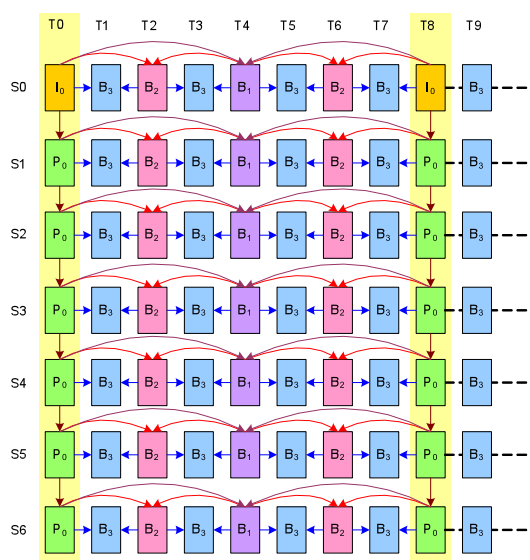


Figure B.4: IPP inter-view prediction for key pictures

Multi-view video coding is investigated by the Joint Video Team (JVT). The JVT is developing a Joint Multi-view Video Model (JMVM) [12] which is based on the video coding standard ITU-T Recommendation H.264-ISO/IEC 14496-10 AVC. The current JMVM proposes illumination change-adaptive motion compensation and prediction structures with hierarchical bi-predictive pictures. The JMVM uses the block-based coding techniques of H.264/AVC to exploit both temporal redundancies and view redundancies. The coding structure has been discussed in this part and investigated in [71]. The standard codec H.264/AVC is a hybrid video codec and incorporates an intra-frame codec and a motion-compensated inter-frame predictor. A survey of coding algorithms and transport methods for 3DTV has been discussed by [66, 180].

Predictive coding schemes are technologically well advanced and offer good quality at low bit rates, in particular with the advent of the latest standard H.264/AVC. However, these schemes required inherent constraint of sequential coding which affect the subsequent coding decisions. This affects overall coding efficiency and produces limited flexibility of the multi-view video coding.

Besides the JMVM, the JVT also developing a Joint Scalable Video Model (JSVM) [181] that supports adaptive lifted wavelets. Subband coding schemes offer flexible representations for multi-view imagery. Further examples for multi-view wavelet video coding are given in [41]. However, decompositions of the motion and disparity compensated lifted wavelets usually suffer compensation mismatch through predict and update steps for multi-connected motion and disparity fields. This compensation mismatch alters properties that are offered by the corresponding non-adaptive wavelet transforms [178].

B.2 H.264/AVC Different Modes of Operation

The multi-frame referencing is the key property of the H.264/AVC standard that enables prediction of blocks of a P-frame being coded using a previous I-frame of multiple previous coded P-frames. The fact that there are high correlations among different views of a multi-view sequence led the development of a H.264/AVC based on multi-view video coding technique with 5 modes of operation by MMRG H.264 Multi-view Extension [76]. Five different mode of operation of the H.264/AVC based on multi-view video coding scheme are explained in the following.

A block diagram of Mode 1 of operation is shown in Figure B.5(a), where the previous frames of closest camera sequence in addition to previous frames of the encoded camera sequence. Figure B.5(b) shows Mode 2 operation, where the latest frame from one nearby camera and latest frame from encoded camera sequence are used. For Figure B.5(c), the latest frames from two nearby cameras and latest frame from encoded camera sequence are used. Meanwhile Figure B.5(d) illustrates the only the previous frames of the encoded camera sequence are used in Mode 4. Lastly, in Mode 5, which is shown in Figure B.5(e), the latest frames from all the cameras in addition to one more frame from one of the closest cameras are used.

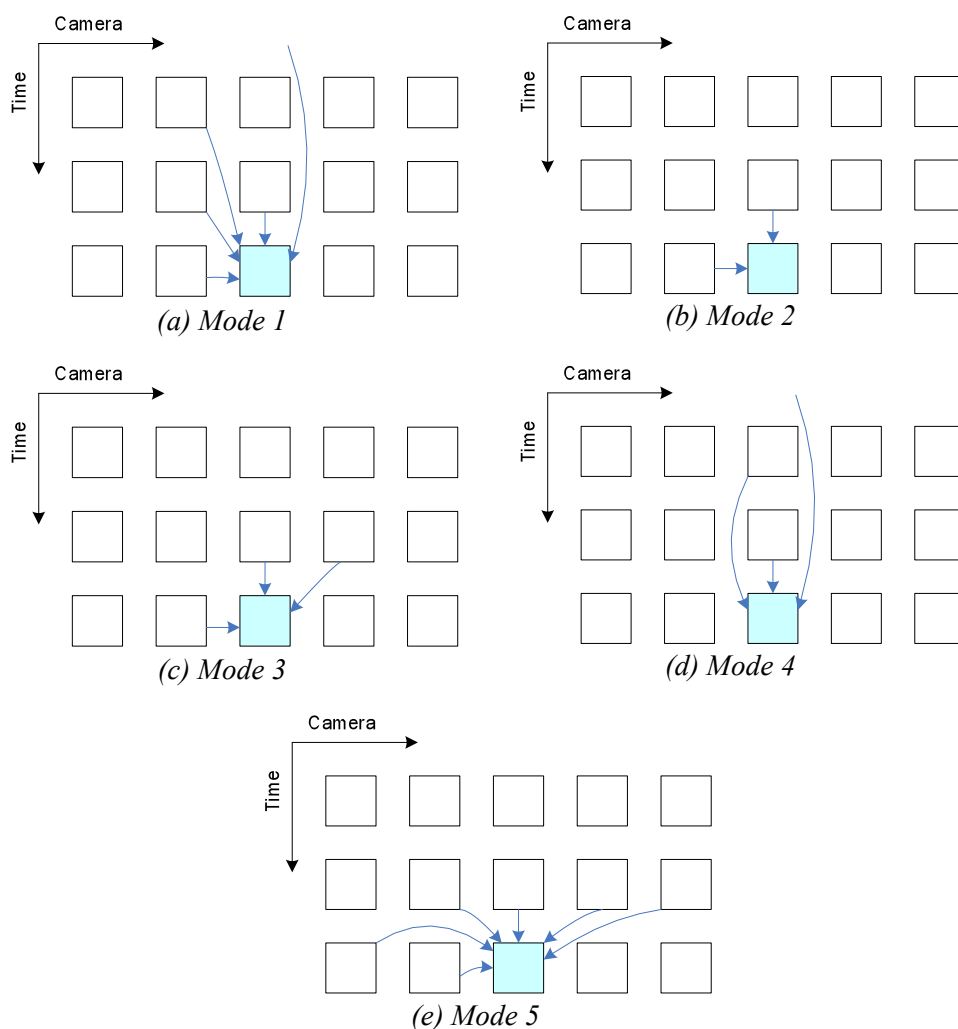


Figure B.5: Reference modes in H.264/AVC multi-view extension codec

The parameters for multi-view video coding have been set in the encoder configuration file. The sample of the configuration displayed in the following is a part of the source code file shown in Figure B.6. The Multi View Coding parameter section in Figure B.6

defined the *MultiViewCount* that indicates the number of cameras in the multiview configuration. If it is 1 the encoder works exactly as the monoscopic case. Otherwise the input file names and reconstructed file names are configured by the entries “*InputFileStructure*” and “*ReconFileStructure*”.

```
#####
# Files
#####
InputFile           = "book0.yuv" # Input sequence
InputHeaderLength   = 0 # If the inputfile has a header
StartFrame          = 0 # Start frame for encoding. (0-N)
FramesToBeEncoded   = 50 # Number of frames to be coded
FrameRate           = 15.0 # Frame Rate per second (0.1-100.0)
SourceWidth         = 352 # Frame width
SourceHeight        = 288 # Frame height
TraceFile           = "trace_book.txt"
ReconFile            = "outbook.yuv"
OutputFile          = "avc_book.264"

.. .. (other configurations) .. ..

#####
# Multi View Coding
#####
MultiViewCount      = 4 # 1 monoscopic, else number of cameras
InputFileStructure  = book%d.YUV # overwrites inputFile - Starts
# from 0
ReconFileStructure  = outbook%d.YUV # overwrites ReconFile - Starts
# from 0
ReferenceMode       = 2 # 1, 2, 3, 4, 5
AnalyzeFile         = "analyze_book"
StandardCompatible  = 0
```

Figure B.6: Multi-view video coding parameter configuration file

The *InputFileStructure* configures the file name structure of the input yuv sequences from the individual cameras. Input file structure should be in the form “input%nd.yuv” where *n* is the number of digits that indicates the camera number in the sequence. In this case, the “input” is the head of the input video files and named as “book”. This entry does not have any effect in the input file name if “*MultiViewCount*” is set to 1. The *ReconFileStructure* configures the reconstructed file names in the same way as the “*InputFileStructure*”. The *ReferenceMode* used to set the 5 reference modes in the MMRG H.264 Multiview Extension, which have been explained in Figure B.5. Different modes can be added in the later versions.

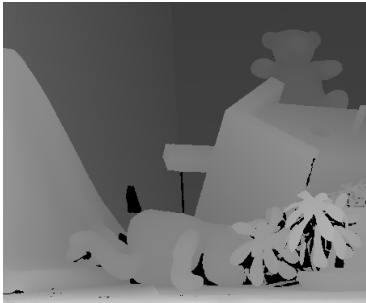
Lastly, *AnalyzeFile* configures the name for the analyze files which are generated for all encoded sequences. These analyze files can be used with the MMRG H.264 Analyzer in order to view encoding information for each macroblock, whether it is intra coded, skipped or which camera sequence it is referred from.

Appendix C: Middlebury Stereo Evaluation

C.1 Middlebury Stereo Datasets



(a) *Teddy left image*



(b) *Teddy ground truth*



(c) *Venus left image*



(d) *Venus ground truth*



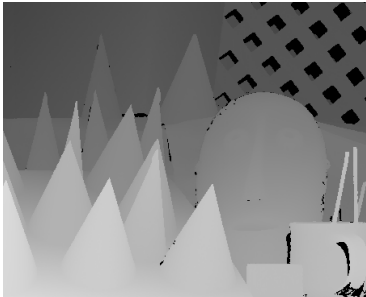
(e) *Tsukuba left image*



(f) *Tsukuba ground truth*



(g) *Cones left image*



(h) *Cones ground truth*

Figure C.1: Middlebury stereo image pairs dataset for left images with their corresponding ground truth disparity maps [22]

C.2 Middlebury Stereo Ranking System

The Middlebury Stereo Evaluation [177] rank is sorted based on which algorithms could give the smallest percentage value of bad pixels for each dataset. The sample of the screenshot for the results is given by C.2. The small blue numbers to the right of each result (e.g., in the first line, the number 15 to the right of 1.07) are the specific ranks for each algorithm for the given dataset. In this case for example it means that the *ADCensus* algorithm was ranked 15th best amongst all algorithms in the database when dealing with the ‘Tsukuba’ non-occlusion dataset. Similarly, the *CoopRegion* algorithm (2nd row) was ranked first for all regions in ‘Tsukuba’, as it produce the smallest percentage error of just 1.16 (shown in bold). However, this algorithm could not give the minimum percentage error in comparison with the other datasets and performance measures.










Algorithm	Avg.	Tsukuba <small>ground truth</small>			Venus <small>ground truth</small>			Teddy <small>ground truth</small>			Cones <small>ground truth</small>			Average percent of bad pixels (explanation)
	Rank ▼	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
ADCensus [94]	8.3	<u>1.07</u> ₁₅	1.48 ₁₃	5.73 ₁₇	<u>0.09</u> ₂	0.25 ₇	1.15 ₃	<u>4.10</u> ₈	6.22 ₃	10.9 ₇	<u>2.42</u> ₈	7.25 ₇	6.95 ₉	 3.97
CoopRegion [41]	10.1	<u>0.87</u> ₄	1.16 ₁	4.61 ₃	<u>0.11</u> ₄	0.21 ₃	1.54 ₇	<u>5.16</u> ₁₈	8.31 ₁₂	13.0 ₁₅	<u>2.79</u> ₂₁	7.18 ₆	8.01 ₂₇	 4.41
AdaptingBP [17]	10.2	<u>1.11</u> ₁₉	1.37 ₇	5.79 ₁₉	<u>0.10</u> ₃	0.21 ₄	1.44 ₆	<u>4.22</u> ₁₀	7.06 ₇	11.8 ₁₁	<u>2.48</u> ₁₀	7.92 ₁₄	7.32 ₁₃	 4.23
RVbased [116]	13.3	<u>0.95</u> ₉	1.42 ₁₁	4.98 ₈	<u>0.11</u> ₆	0.29 ₁₁	1.07 ₁	<u>5.98</u> ₂₆	11.6 ₃₅	15.4 ₃₂	<u>2.35</u> ₆	7.61 ₈	6.81 ₈	 4.88
RDP [102]	13.8	<u>0.97</u> ₁₀	1.39 ₉	5.00 ₉	<u>0.21</u> ₂₄	0.38 ₁₉	1.89 ₁₄	<u>4.84</u> ₁₂	9.94 ₂₁	12.6 ₁₃	<u>2.53</u> ₁₁	7.69 ₁₀	7.38 ₁₄	 4.57
DoubleBP [35]	13.9	<u>0.88</u> ₆	1.29 ₄	4.76 ₆	<u>0.13</u> ₈	0.45 ₂₆	1.87 ₁₃	<u>3.53</u> ₆	8.30 ₁₁	9.63 ₄	<u>2.90</u> ₂₇	8.78 ₃₅	7.79 ₂₁	 4.19
OutlierConf [42]	14.6	<u>0.88</u> ₅	1.43 ₁₂	4.74 ₅	<u>0.18</u> ₁₇	0.26 ₉	2.40 ₂₄	<u>5.01</u> ₁₄	9.12 ₁₈	12.8 ₁₄	<u>2.78</u> ₂₀	8.57 ₂₇	6.99 ₁₀	 4.60
SubPixDoubleBP [30]	19.6	<u>1.24</u> ₂₇	1.76 ₃₁	5.98 ₂₃	<u>0.12</u> ₇	0.46 ₂₈	1.74 ₁₀	<u>3.45</u> ₅	8.38 ₁₃	10.0 ₆	<u>2.93</u> ₃₀	8.73 ₃₂	7.91 ₂₃	 4.39
SurfaceStereo [79]	19.8	<u>1.28</u> ₃₂	1.65 ₂₁	6.78 ₃₉	<u>0.19</u> ₁₉	0.28 ₁₀	2.61 ₃₅	<u>3.12</u> ₃	5.10 ₁	8.65 ₁	<u>2.89</u> ₂₈	7.95 ₁₆	8.26 ₃₅	 4.06

Figure C.2: Sample of Middlebury Stereo Evaluation Results, where ‘nonocc’ (for non-occluded regions), ‘all’ (for all regions) and ‘disc’ (for discontinuities regions)

For ‘Teddy’ dataset, the minimum percentage error for all regions and discontinuities obtained by the *SurfaceStereo* algorithm are 5.10 and 8.65 respectively, and therefore it was ranked first for this specific datasets. The average rank value (*Avg. Rank*) is calculated using the total specific ranks for each dataset (in all twelve columns shown). As shown in Figure C.2, first overall rank is given to *ADCensus* algorithm, average rank value of 8.3. This can be obtained the average of all the specific rank values in the twelve columns. The *CoopRegion* algorithm was 2nd with a 10.1 average rank value. We noticed that the *SurfaceStereo* algorithm is not first in the ranking even though it produced the smallest percentage bad pixels error in the ‘Teddy’ datasets. While the average rank provides a reasonable way of ordering the methods, the average error

visualized by the error bar shows just how close together the algorithms are in comparison.

Author's Publication

1. Nurulfajar Abd Manap, John J. Soraghan. "Depth Image Layers Separation Algorithms for Novel View Synthesis". *Image and Vision Computing Journal*. 2012. *Submitted*.
2. Nurulfajar Abd Manap, John J. Soraghan. "Disparity Refinement Based on Depth Image Layers Separation for Stereo Matching Algorithms". *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. Vol. 4, No. 1. 2012.
3. Gaetano Di Caterina, Nurulfajar Abd Manap, Masrullizam Mat Ibrahim, John J. Soraghan. "Real Time Door Access Event Detection and Notification in a Reactive Smart Surveillance System". *International Conference on Image and Signal Processing (ICISP)*. 2012. Agadir, Morocco.
4. Nurulfajar Abd Manap, Gaetano Di Caterina, Masrullizam Mat Ibrahim, John J. Soraghan. "Collaborative Surveillance Cameras for High Quality Face Acquisition in a Real-Time Door Monitoring System". *European Workshop on Visual Information Processing (EUVIP 2011)*. July 2011. Paris, France.
5. Nurulfajar Abd Manap, John J. Soraghan. "Novel View Synthesis Based on Depth Map Layers Representation". *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-Con 2011)*. May 2011. Antalya, Turkey.
6. Nurulfajar Abd Manap, Gaetano Di Caterina, John Soraghan, Vijay Sidharth, Hui Yao. "Face Detection and Stereo Matching Algorithms for Smart Surveillance System with IP Cameras". *European Workshop on Visual Information Processing (EUVIP 2010)*. July 2010. Paris, France.
7. Nurulfajar Abd Manap, Gaetano Di Caterina, John Soraghan, Vijay Sidharth, Hui Yao. "Smart Surveillance System Based on Stereo Matching Algorithms with IP and PTZ Cameras". *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-Con 2010)*. June 2010. Tampere, Finland.

8. Nurulfajar Abd Manap, John Soraghan. "Multi-view Video Coding for Wireless Channel". *4th International Symposium on Broadband Communications (ISBC 2010)*. July 2010. Melaka, Malaysia.
9. Nurulfajar Abd Manap, John Soraghan. "Multi-view Video Coding for 3DTV". *Malaysia Glasgow Doctoral Colloquium (MGDC)*. Jan 2010. Glasgow, UK.
10. Nurulfajar Abd Manap, Gaetano Di Caterina, John Soraghan. "Low Cost Multi-view Video System for Wireless Channel". *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-Con 2009)*. May 2009. Potsdam, Germany.

References

- [1] I. Ahmad, "Multi-View Video: Get Ready for Next-Generation Television," *Distributed Systems Online, IEEE*, vol. 8, pp. 6-6, 2007.
- [2] A. Kubota, *et al.*, "Multiview Imaging and 3DTV," *IEEE Signal Processing Magazine*, vol. 24, pp. 10-21, November 2007.
- [3] J. Berent and P. L. Dragotti, "Plenoptic Manifolds," *Signal Processing Magazine, IEEE*, vol. 24, pp. 34-44, 2007.
- [4] C. Theobalt, *et al.*, "High-Quality Reconstructon from Multiview Video Streams," *IEEE Signal Processing Magazine*, vol. 24, pp. 45-57, November 2007.
- [5] M. Magnor, *et al.*, "Video-Based Rendering", *Proceeding ACM SIGGRAPH*, Course 16, 2005.
- [6] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proceeding ACM SIGGRAPH*, pp. 31-42, 1996.
- [7] P. E. Debevec, *et al.*, "Modeling And Rendering Architecture From Photographs: A Hybrid Geometry and Image-Based Approach," *Proceeding ACM SIGGRAPH*, pp. 11-20, 1996.
- [8] L. Zitnick, *et al.*, "High-Quality Video View Interpolation Using A Layered Representation," *ACM Trans. Graph*, vol. 23, pp. 600-608, 2004.
- [9] S. M. Seitz and C. R. Dyer, "Photorealistic Scene Reconstruction By Voxel Coloring," *IEEE Proceedings on Computer Vision and Pattern Recognition (CVPR)*, pp. 1067-1073, 1997.
- [10] K. Mueller, *et al.*, "Compressing Time-Varying Visual Content," *IEEE Signal Processing Magazine*, vol. 24, pp. 58-65, November 2007.
- [11] M. Flierl and B. Girod, "Multiview Video Compression," *IEEE Signal Processing Magazine*, vol. 24, pp. 66-76, 2007.
- [12] A. Vetro, *et al.*, "Joint Multiview Video Model JMVM 2.0," *ITU-T and ISO/IEC Joint Video Team. Document JVT-U207*, 2006.
- [13] ISO/IEC/JTC1/SC29/WG11, "Survey of Algorithms used for Multi-view Video Coding (MVC)," January 2005.
- [14] M. Zwicker, *et al.*, "Resampling, Antialiasing and Compression in Multiview 3D Display," *IEEE Signal Processing Magazine*, pp. 88-96, November 2007.
- [15] J. Konrad and M. Halle, "3D Displays and Signal Processing," *IEEE Signal Processing Magazine*, vol. 24, pp. 97-111, November 2007.
- [16] B. G. Blundell and A. J. Schwarz, "The Classification of Volumetric Display Systems: Characteristics and Predictability of The Image Space," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, pp. 66-75, 2002.
- [17] G. E. Favalora, "Volumetric 3D Displays and Application Infrastructure," *Computer*, vol. 38, pp. 37-44, 2005.

- [18] M. Flierl and B. Girod, "Coding of Multi-View Image Sequences with Video Sensors," *IEEE International Conference on Image Processing (ICIP)*, pp. 609-612, 2006.
- [19] N. A. Dodgson, "Autostereoscopic 3D Displays," *IEEE Computer Society*, vol. 38, pp. 31-36, August 2005.
- [20] Philips 3D Solutions, "3D Interface Specifications, White Paper," *Philips Electronics Nederland B.V.*, October 2007.
- [21] C. Wang, *et al.*, "A Novel Intermediate View Synthesis Method Based on Disparity Estimation," *Proceeding of the First IEEE International Conference on Information Science and Engineering (ICISE 09)*, pp. 1079-1082, 2009.
- [22] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7-42, 2002.
- [23] L. D. Stefano and S. Mattocchia, "Real-Time Stereo Within the VIDET Project," *Elsevier Science Ltd: Real-Time Imaging*, vol. 8, pp. 439-453, 2002.
- [24] P. Fua, "Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities," *International Joint Conference on Artificial Intelligence*, pp. 1292-1298, August 1991.
- [25] L. D. Stefano and S. Mattocchia, "Fast Stereo Matching for the VIDET System using a General Purpose Processor with Multimedia Extensions," *Proceedings of the Fifth IEEE International Workshop on Computer Architectures for Machine Perception (CAMP)*, pp. 356-362, 2000.
- [26] J. Zhu, *et al.*, "Fusion Of Time-Of-Flight Depth and Stereo for High Accuracy Depth Maps," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 231-236, 2008.
- [27] Y. S. Kang and Y.-S. Ho, "Disparity Map Generation for Color Image using TOF Depth Camera," *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 1-4, 2011.
- [28] A. Kolb, *et al.*, "Time-of-Flight Cameras in Computer Graphics," *Computer Graphics Forum*, vol. 29, pp. 141-159, February 2010.
- [29] J. Tscherrig, "Activity Recognition with Kinect," Technical Report, Trier University, 2011.
- [30] H. Y. Shum, *et al.*, "Survey of Image-Based Representations and Compression Techniques," *IEEE Trans. Circuits Systems Video Technology*, vol. 13, pp. 1020-1037, Nov. 2003.
- [31] J. Kilner, *et al.*, "Objective Quality Assessment in Free-Viewpoint Video Production," *Signal Processing: Image Communication*, vol. 24, pp. 3-16, January 2009.
- [32] D. Scharstein, "Stereo Vision for View Synthesis," *Proc. of Conference on Computer Vision and Pattern Recognition*, pp. 852-858, 1996.
- [33] V. Vaish, *et al.*, "Reconstructing Occluded Surfaces Using Synthetic Apertures: Stereo, Focus and Robust Measures," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2331-2338, 2006.
- [34] B. Wilburn, *Et Al.*, "High Performance Imaging Using Large Camera Arrays," *ACM Transactions on Graphics*, vol. 24, pp. 765-776, July 2005.
- [35] M. Tanimoto, "FTV (Free-Viewpoint TV)," *IEEE International Conference on Image Processing (ICIP)*, 2393-2396, 2010.

- [36] M. Tanimoto, "Overview of Free Viewpoint Television," *Signal Processing: Image Communication*, vol. 21, pp. 454-461, 2006.
- [37] Y. Morvan, "Acquisition, Compression and Rendering of Depth and Texture for Multi-view Video," PhD Thesis, Eindhoven University of Technology, 2009.
- [38] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," *Computational Models of Visual Processing*, pp. 3-20, 1991.
- [39] J. Shade, *et al.*, "Layered Depth Images," *Proceedings Of The 25th Annual Conference On Computer Graphics And Interactive Techniques*, pp. 231-242, 1998.
- [40] J. U. Garbas, *et al.*, "4D Scalable Multi-View Video Coding Using Disparity Compensated View Filtering and Motion Compensated Temporal Filtering," *IEEE 8th Workshop on Multimedia Signal Processing*, pp. 54-58, 2006.
- [41] W. Yang, *et al.*, "4D Wavelet-Based Multiview Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 1385-1396, 2006.
- [42] J. Berent, *et al.*, "Adaptive Layer Extraction for Image Based Rendering," *International Workshop on Multimedia Signal Processing*, pp. 266-271, 2009.
- [43] P. Merkle, *et al.*, "Efficient Compression Of Multi-View Depth Data Based On MVC," *Proceedings of 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2007.
- [44] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed.: O'Reilly Media, Inc., 2008.
- [45] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*: Cambridge University Press, 2003.
- [46] A. Bovik, *Handbook of Image and Video Processing*, Second Ed.: Elsevier Academic Press, 2005.
- [47] Y. Morvan, *et al.*, "System Architecture for Free-Viewpoint Video and 3D-TV," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 925-932, May 2008.
- [48] A. Smolic, *et al.*, "Development of MPEG Standards for 3D and Free Viewpoint Video," *SPIE Conference Optics East 2005: Communications, Multimedia & Display Technologies*, vol. 6014, pp. 262-273, Nov 2005.
- [49] J. Kilner, "Free-Viewpoint Video for Outdoor Sporting Events," Technical Report, University of Surrey, 2006.
- [50] B. Mendiburu, *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*: Focal Press, 2009.
- [51] S. C. Chan, *et al.*, "Image-Based Rendering and Synthesis Technological Advances and Challenges," *IEEE Signal Processing Magazine*, vol. 24, pp. 22-33, November 2007.
- [52] H. Y. Shum, *et al.*, *Image-Based Rendering*: Springer, 2007.
- [53] M. Magnor, *Video-Based Rendering*: A. K. Peters, 2005.
- [54] A. Smolic and D. McCutchen, "3DAV Exploration of Video-Based Rendering Technology in MPEG," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 14, no. 4, pp. 348-356, 2004.
- [55] R. Szeliski, *Computer Vision Algorithms and Applications*: Springer, 2011.
- [56] H. M. Ozaktas, *et al.*, *Three-Dimensional Television: Capture, Transmission, Display*: Springer Publishing Company, Incorporated, 2007.

- [57] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*: Prentice Hall PTR, 1998.
- [58] J. Y. Bouguet, "Camera Calibration Toolbox with Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc, 2008. Last accessed: May 7, 2012.
- [59] S. A. Gudmundsson, *et al.*, "Fusion of stereo vision and Time-of-Flight imaging for improved 3D estimation," *Int. J. Intelligent Systems Technologies and Applications*, vol. 5, pp. 425-433, 2008.
- [60] S. Hussmann, *et al.*, "A Performance Review of 3D TOF Vision Systems in Comparison to Stereo Vision Systems," *Stereo Vision*, I-Tech Education and Publishing, pp. 103-120, 2008.
- [61] D. Herrera, *et al.*, "Accurate and Practical Calibration of a Depth and Color Camera Pair," *Computer Analysis of Images and Patterns - 14th International Conference*, vol. 6855, pp. 437-445, 2011.
- [62] Y. Mori, *et al.*, "View Generation With 3D Warping Using Depth Information for FTV," *Signal Processing: Image Communication*, vol. 24, pp. 65-72, 2009.
- [63] S. Shimizu, *et al.*, "View Scalable Multi-View Video Coding Using 3-D Warping with Depth Map," *IEEE Transactions on Circuits and Systems for Video Technology*, pp.1-13, 2007.
- [64] B.-B. Chai, *et al.*, "Depth Map Compression for Real-Time View-Based Rendering," *Elsevier Science: Pattern Recognition Letters*, vol. 25, pp. 755-766, 2004.
- [65] Y. Morvan, "Acquisition, Compression and Rendering of Multi-view Video," *NXP semiconductor, workshop on Perspectives on Vision Networks*, Technical Report, August 2007.
- [66] A. Smolic, *et al.*, "Coding Algorithms for 3DTV - A Survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, pp. 1606-1621, 2007.
- [67] A. Smolic, *et al.*, "3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards," *IEEE International Conference on Multimedia and Expo*, pp. 2161-2164, 2006.
- [68] ITU-T and ISO/IEC, "Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," *ITU-T Rec. H.222.0 ISO/IEC 13818-1 (MPEG 2 Systems)*, Nov 1994.
- [69] C. Fehn, *et al.*, "An Evolutionary And Optimised Approach On 3D-TV," *Proceedings of International Broadcast Conference*, pp. 357-365, 2002.
- [70] M. Flierl, *et al.*, "Motion and Disparity Compensated Coding for Video Camera Arrays," *Proceedings of Picture Coding Symposium*, pp. 1-11, 2006.
- [71] P. Merkle, *et al.*, "Efficient Prediction Structures for Multi-View Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1461-1475, 2007.
- [72] G. Li and Y. He, "A Novel Multi-View Video Coding Scheme Based On H.264," *Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing*, vol. 1, pp. 15-18, 2003.
- [73] U. Fecker and A. Kaup, "H.264/AVC - Compatible Coding Of Dynamic Light Fields Using Transposed Picture Ordering," *13th European Signal Processing Conference*, pp. 1-4, 2005.
- [74] C. Fehn, "3D-TV Using Depth-Image-Based Rendering (DIBR)," *Visualization, Imaging and Image Processing (VIIP)*, pp. 1-6, 2003.

- [75] E. Martinian, *et al.*, "View Synthesis for Multiview Video Compression," *Picture Coding Symposium (PCS)*, pp. 2981-2984, 2006.
- [76] C. Bilen, *et al.*, "A Multi-view Video Codec Based on H.264," *IEEE ICIP*, pp. 1-4, 2006.
- [77] A. S. Akbari, *et al.*, "A Novel H.264/AVC Based Multi-View Video Coding Scheme," *Proceedings of 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2007.
- [78] A. Manta, "Multiview Imaging and 3DTV. A Survey," Technical Report, TU Delft, 2008.
- [79] Y. Boykov and O. Veksler, "Graph Cuts in Vision and Graphics: Theories and Applications," *In Handbook of Mathematical Models in Computer Vision*, pp. 100-119, 2006.
- [80] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2004*, vol. 1, pp. 261-268, 2004.
- [81] A. Klaus, *et al.*, "Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure," *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 3, pp. 15-18, 2006.
- [82] N. Grammalidis and M. G. Strintzis, "Disparity and Occlusion Estimation In Multiocular Systems and Their Coding for the Communication Of Multiview Image Sequences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 328-344, 1998.
- [83] M. C. Sung, *et al.*, "Stereo Matching Using Multi-directional Dynamic Programming," *International Symposium on Intelligent Signal Processing and Communications (ISPACS)*, pp. 697-700, 2006.
- [84] D. Tzovaras, *et al.*, "Disparity Field And Depth Map Coding For Multiview 3D Image Generation," *Signal Processing: Image Communication*, vol. 11, pp. 205-230, Jan 1998.
- [85] L. D. Stefano, *et al.*, "A Fast Area-based Stereo Matching Algorithm," *Proceedings from the 15th International Conference on Vision Interface*, vol. 22, pp. 983-1005, October 2004.
- [86] R. Szeliski and R. Zabih, "An Experimental Comparison of Stereo Algorithms," *Vision Algorithms: Theory and Practice, number 1883 in LNCS*, pp. 1-19, 1999.
- [87] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1088-1095, 1991.
- [88] J. Li, *et al.*, "Robust Stereo Image Matching Using a Two-Dimensional Monogenic Wavelet Transform," *OPTICS LETTERS*, vol. 34, pp. 3514-3516, November 2009.
- [89] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [90] H. Bay, *et al.*, "SURF: Speeded Up Robust Features," in *Computer Vision -- ECCV 2006* vol. 3951, pp. 404-417, 2006.
- [91] E. Delponte, *et al.*, "SVD-Matching Using SIFT Features," *Graphical Models*, vol. 68, pp. 415-431, 2006.
- [92] A. Basharat, *et al.*, "Content Based Video Matching Using Spatiotemporal Volumes," *Computer Vision and Image Understanding*, vol. 110, pp. 360-377, 2008.

- [93] Y. S. Heo, *et al.*, "Mutual Information-Based Stereo Matching Combined With SIFT Descriptor In Log-Chromaticity Color Space," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 445-452, 2009.
- [94] D. Wan and J. Zhou, "Stereo Vision Using Two PTZ Cameras," *Computer Vision and Image Understanding*, vol. 112, pp. 184-194, 2008.
- [95] M. Stommel, *et al.*, "A Fast, Robust And Low Bit-Rate Representation for SIFT and SURF Features" *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 278-283, 2011.
- [96] E. Tola, *et al.*, "A Fast Local Descriptor For Dense Matching," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [97] L. D. Stefano, *et al.*, "A PC-Based Real-Time Stereo Vision System," *Machine Graphics & Vision*, vol. 13, pp. 197-220, 2004.
- [98] B. Cyganek and J. P. Siebert, *An Introduction to 3D Computer Vision Techniques and Algorithms*: John Wiley & Sons, Ltd, 2009.
- [99] S. Yoon, *et al.*, "Fast Correlation-Based Stereo Matching With The Reduction Of Systematic Errors," *Pattern Recognition Letters*, vol. 26, pp. 2221-2231, 2005.
- [100] M. Humenberger, *et al.*, "A Fast Stereo Matching Algorithm Suitable For Embedded Real-Time Systems," *Computer Vision and Image Understanding*, vol. 114, pp. 1180-1202, 2010.
- [101] B. Cyganek and J. Borgosz, "Maximum Disparity Threshold Estimation for Stereo Imaging Systems via Variogram Analysis," *Computational Science ICCS Springer Berlin*, vol. 2657, pp. 591-600, 2003.
- [102] M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 353-363, 1993.
- [103] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," *In Proceedings of European Conference on Computer Vision*, pp. 151-158, 1994.
- [104] K. Muhlmann, *et al.*, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation," *International Journal of Computer Vision*, vol. 47, pp. 79-88, 2002.
- [105] S. T. Barnard and M. A. Fischler, "Stereo Vision," *Encyclopedia of Artificial Intelligence*, vol. New York: John Wiley, pp. 1083-1090, 1987.
- [106] L. Zitnick and T. Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 675-684, 2000.
- [107] M. M. H. Chowdhury and M. A. A. Bhuiyan, "A New Approach for Disparity Map Determination," *Daffodil International University Journal of Science and Technology*, vol. 4, pp. 9-13, January 2009.
- [108] F. Tombari, *et al.*, "Classification And Evaluation of Cost Aggregation Methods For Stereo Correspondence," *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, 2008.
- [109] O. Veksler, "Fast Variable Window for Stereo Correspondence Using Integral Images," *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pp. 556-561, 2003.

- [110] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, May 2002.
- [111] V. Kolmogorov, "Graph Based Algorithms for Scene Reconstruction from Two or More Views," PhD Thesis, Cornell University, September 2003.
- [112] Z. F. Wang and Z. G. Zheng, "A Region Based Stereo Matching Algorithm Using Cooperative Optimization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, 2008.
- [113] R. Szeliski, *et al.*, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1068-1080, 2008.
- [114] B. Ozkalayci, *et al.*, "Multi-View Video Coding Via Dense Depth Estimation," *Proceedings of 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2007.
- [115] J. Sun, *et al.*, "Image Completion With Structure Propagation," *ACM SIGGRAPH*, pp. 851-858, 2005.
- [116] H. Hirschmuller and S. Gehrig, "Stereo Matching in the Presence of Sub-Pixel Calibration Errors," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 437-444, 2009.
- [117] S. Mattoccia, *et al.*, "Accurate And Efficient Cost Aggregation Strategy For Stereo Correspondence Based On Approximated Joint Bilateral Filtering," *Asian Conference on Computer Vision (ACCV)*, pp. 23-27, 2009.
- [118] M. Bleyer and M. Gelautz, "A Layered Stereo Matching Algorithm Using Image Segmentation And Global Visibility Constraints," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, pp. 128-150, 2005.
- [119] L. Hong and G. Chen, "Segment-based Stereo Matching Using Graph Cuts," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2004*, vol. 1, pp. 74-81, 2004.
- [120] F. Tombari, *et al.*, "Segmentation-Based Adaptive Support for Accurate Stereo Correspondence," *IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pp. 427-438, 2007.
- [121] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *ACM Graphics and Image Processing*, vol. 24, pp. 381-395, June 1981.
- [122] S. Mattoccia, "A Locally Global Approach to Stereo Correspondence," *IEEE International Conference on Computer Vision Workshop, ICCV Workshops*, pp. 1763-1770, 2009.
- [123] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *International Journal of Computer Vision*, vol. 35, pp. 269-293, 1999.
- [124] I. J. Cox, *et al.*, "A Maximum Likelihood Stereo Algorithm," *Computer Vision and Image Understanding*, vol. 63, pp. 542-567, 1996.
- [125] A. Fusiello, *et al.*, "Efficient Stereo with Multiple Windowing," *Conference on Computer Vision and Pattern Recognition*, pp. 858-863, 1997.
- [126] A. Banno and K. Ikeuchi, "Disparity Map Refinement and 3D Surface Smoothing Via Directed Anisotropic Diffusion," *Computer Vision and Image Understanding*, vol. 115, pp. 611-619, 2011.

- [127] L. De Maezdu, *et al.*, "Linear Stereo Matching," *International Conference on Computer Vision*, pp. 1708-1715, 2011.
- [128] B. Khaleghi, *et al.*, "An Improved Real-Time Miniaturized Embedded Stereo Vision System (MESVS-II)," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1-8, 2008.
- [129] Z. Zhang, *et al.*, "A Novel Algorithm for Disparity Calculation Based on Stereo Vision," *4th European DSP Education and Research Conference (EDERC)*, pp. 180-184, 2010.
- [130] X. Mei, *et al.*, "On Building An Accurate Stereo Matching System On Graphics Hardware," *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 467-474, 2011.
- [131] S. B. Stevenson and C. M. Schor, "Human Stereo Matching Is Not Restricted To Epipolar Lines," *Vision Research*, vol. 37, pp. 2717-2723, 1997.
- [132] N. Miyazaki, *et al.*, "Offset Vertical Stereo System for Real-Time Range-Finding to Preceding Vehicles," *IAPR Workshop on Machine Vision Applications*, pp. 459-462, 1998.
- [133] G. Caron and E.-M. Mouaddib, "Vertical Line Matching For Omnidirectional Stereovision Images," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2787-2792, 2009.
- [134] S. H. Yang, *et al.*, "Neural Network Based Stereo Matching Algorithm Utilizing Vertical Disparity," *36th Annual Conference on IEEE Industrial Electronics Society (IECON)*, pp. 1155-1160, 2010.
- [135] R. Yang and Z. Zhang, "Eye Gaze Correction with Stereovision for Video-Teleconferencing," in *Computer Vision ECCV*, vol. 2351, pp. 761-763, 2002.
- [136] S. B. Kang, *et al.*, "The Geometry-Image Representation Tradeoff for Rendering," *IEEE International Conference on Image Processing*, vol. 2, pp. 13-16, 2000.
- [137] J. Carranza, *et al.*, "Free-Viewpoint Video Of Human Actors," *ACM Trans. Graph.*, vol. 22, pp. 569-577, 2003.
- [138] T. Matsuyama, *et al.*, "Real-Time Dynamic 3-D Object Shape Reconstruction And High-Fidelity Texture Mapping for 3-D Video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, pp. 357-369, 2004.
- [139] S. Vedula, *et al.*, "Spatio-Temporal View Interpolation," *Proceedings of the 13th Eurographics Workshop on Rendering*, pp. 65-76, June 2002.
- [140] B. Bai and J. Harms, "A Multiview Video Transcoder" *Proceedings of The 13th Annual ACM International Conference on Multimedia* pp. 503-506, 2005.
- [141] T. Fujii, *et al.*, "Ray Space Coding for 3D Visual Communication," *Picture Coding Symposium*, pp. 447-451, March 1996.
- [142] S. J. Gortler, *et al.*, "The Lumigraph," *Proceedings Of The 23rd Annual Conference On Computer Graphics And Interactive Techniques*, pp. 43-54, 1996.
- [143] C. Buehler, *et al.*, "Unstructured Lumigraph Rendering," *Proceedings Of The 28th Annual Conference On Computer Graphics And Interactive Techniques*, pp. 425-432, 2001.
- [144] H. Y. Shum and L. W. He, "Rendering With Concentric Mosaics," *Proceedings of the 26th ACM SIGGRAPH*, pp. 299-306, 1999.

- [145] C. Zhang and T. Chen, "A Survey on Image-Based Rendering - Representation, Sampling and Compression," *Technical Report AMP*, 2003.
- [146] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proceedings Of The 22nd Annual Conference On Computer Graphics And Interactive Techniques*, pp. 39-46, 1995.
- [147] J. X. Chai, *et al.*, "Plenoptic Sampling," *Proceedings Of The 27th Annual Conference On Computer Graphics And Interactive Techniques*, pp. 307-318, 2000.
- [148] H. Y. Shum, *et al.*, "Pop-Up Light Field: An Interactive Image-Based Modeling And Rendering System," *ACM Trans. Graph.*, vol. 23, pp. 143-162, April 2004.
- [149] J. Pearson, *et al.*, "Accurate Non-Iterative Depth Layer Extraction Algorithm for Image Based Rendering," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 901-904, 2011.
- [150] P. Kauff, *et al.*, "Depth Map Creation and Image-Based Rendering for Advanced 3DTV Services Providing Interoperability And Scalability," *Signal Processing: Image Communication*, vol. 22, pp. 217-234, February 2007.
- [151] S. E. Chen and L. Williams, "View Interpolation for Image Synthesis," *International Conference on Computer Graphics and Interactive Techniques*, pp. 279-288, 1993.
- [152] J. H. Park and H. W. Park, "Fast View Interpolation Of Stereo Images Using Image Gradient And Disparity Triangulation," *Signal Processing: Image Communication*, vol. 18, pp. 381-384, 2003.
- [153] M. Lhuillier and L. Quan, "Image-Based Rendering By Joint View Triangulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 1051-1063, 2003.
- [154] A. Redert, *et al.*, "Correspondence Estimation In Image Pairs," *IEEE Signal Processing Magazine*, vol. 16, pp. 29-46, 1999.
- [155] H. Fan and K. N. Ngan, "Disparity Map Coding Based On Adaptive Triangular Surface Modelling," *Signal Processing: Image Communication*, vol. 14, pp. 119-130, 1998.
- [156] S. Sethuraman, *et al.*, "A Multiresolution Framework For Stereoscopic Image Sequence Compression," *IEEE International Conference Image Processing*, vol. 2, pp. 361-365, 1994.
- [157] S. Wang and Y. Wang, "Multiview Video Sequence Analysis, Compression, And Virtualviewpoint Synthesis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 397-410, 2000.
- [158] A. K. Jain, *et al.*, "Efficient Stereo-to-Multiview Synthesis," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 889-892, 2011.
- [159] M. Levoy, "Light Fields and Computational Imaging," *Computer*, vol. 39, pp. 46 -55, August 2006.
- [160] K. Takahashi and T. Naemura, "Layered Light-Field Rendering With Focus Measurement," *Signal Processing: Image Communication*, vol. 21, pp. 519-530, 2006.
- [161] Y. Li, *et al.*, "Rendering Driven Depth Reconstruction," *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP)*, vol. 4, pp. 780-784, 2003.
- [162] A. Smolic, *et al.*, "Intermediate View Interpolation Based on Multiview Video Plus Depth for Advanced 3D Video Systems," *International Conference on Image Processing (ICIP)*, pp. 2448-2451, 2008.

- [163] M. Ishii, *et al.*, "Joint Rendering and Segmentation of Free-Viewpoint Video," *EURASIP Journal on Image and Video Processing*, vol. 2010, p. 1-10, 2010.
- [164] M. Sjostrom, *et al.*, "Improved Depth-Image Based Rendering Algorithms," *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2011.
- [165] Z. Wang and A. Bovik, "Mean Squared Error: Love It or Leave It?," *IEEE Signal Processing Magazine*, vol. 26, pp. 98-117, 2009.
- [166] Z. Wang, *et al.*, "Image Quality Assessment: From Error Visibility To Structural Similarity," *Image Processing, IEEE Transactions on*, vol. 13, pp. 600-612, 2004.
- [167] L. Chaohui, *et al.*, "Efficient Stereo Disparity Estimation for Intermediate View Synthesis," *The 47th Midwest Symposium on Circuits and Systems, 2004 (MWSCAS)*, vol. 3, pp. 483-486, 2004.
- [168] C. Lu, *et al.*, "Virtual View Synthesis for Multi-view 3D Display," *Third International Joint Conference on Computational Science and Optimization (CSO)*, vol. 2, pp. 444-446, 2010.
- [169] J. D. Oh, *et al.*, "Disparity Estimation and Virtual View Synthesis From Stereo Video," *Proc. IEEE Symposium Circuits and Systems (ISCAS)*, pp. 993-996, 2007.
- [170] F. J. Halim and J. S. Jin, "View Synthesis by Image Mapping and Interpolation," *Proceedings of Pan-Sydney Area Workshop on Visual Information Processing*, vol. 11, pp. 25-30, 2001.
- [171] D. Farin, *et al.*, "View Interpolation Along a Chain of Weakly Calibrated Cameras," *IEEE Workshop on Content Generation and Coding for 3D-Television*, pp. 1-4, 2006.
- [172] K. A. Hunt, *The Art of Image Processing with Java*: CRC Press, 2010.
- [173] A. Fusiello, *et al.*, "Experiments with a New Area-Based Stereo Algorithm," *ICIAP '97 Proceedings of the 9th International Conference on Image Analysis and Processing*, vol. 1, pp. 669-679, 1997.
- [174] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679-698, 1986.
- [175] C. M. Christoudias, *et al.*, "Synergism In Low Level Vision," *Proceedings of 16th International Conference on Pattern Recognition*, pp. 1-6, 2002.
- [176] R. J. Fergusson, "Human Visual System based Object Extraction for Video Coding," PhD Thesis, University of Strathclyde, 1999.
- [177] Middlebury Computer Vision. Stereo Evaluation [Online]. Available: <http://vision.middlebury.edu/stereo/>. Last accessed: February 27, 2012.
- [178] M. Flierl, *et al.*, "Motion and Disparity Compensated Coding for Multiview Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1474-1484, 2007.
- [179] H. Schwarz, *et al.*, "Analysis of Hierarchical B Pictures and MCTF," *International Conference on Multimedia and Expo*, pp. 1929-1932, 2006.
- [180] G. B. Akar, *et al.*, "Transport Methods in 3DTV - A Survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, pp. 1622-1630, 2007.
- [181] A. Vetro, *et al.*, "Joint Draft 6.0 on Multiview Video Coding," *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, 13-18 January, 2008.