ISCTE IUL
Instituto Universitário de Lisboa

Department of Information Science and Technology

Quality Assessment of 2D Image Rendering for 4D Light Field Content

Lourenço de Mértola Belford Correia da Silva

A Dissertation presented in partial fulfilment of the Requirements for the Degree of

Master in Telecommunications and Computer Engineering

Supervisor:
PhD. Paulo Jorge Lourenço Nunes, Assistant Professor,
ISCTE-IUL

Co-supervisor:
PhD. Tomás Gomes da Silva Serpa Brandão, Assistant Professor,
ISCTE-IUL

October 2018

Acknowledgements

I would like to dedicate this accomplishment to my parents, specially to my mother, who I lost in 2016 to cancer. These last two years were tough, but I knew that finishing my master's thesis was one of my mother's wishes and my father kept reminding me on a regular basis.

I'm extremely thankful for all they have done for me through out these years, for the education, principles and support they always gave me. They provided me the tools that I need to succeed, and they are the top two of my role models without any doubt.

Making clear that nothing good comes without focus, dedication and commitment was by far the biggest lesson I've learned and is the rule that I try to apply in everything that I do.

I would like to thank to my supervisors Professor Paulo Nunes and Professor Tomás Brandão. This work wouldn't be possible without their knowledge, guidance, patience and criticism.

A big thanks to my family and my close friends for their support and motivation.

"Make them proud every day"

Resumo

A tecnologia de campos de luz – *Light Field* (LF), composta por representações visuais de dados com grande quantidade de informação, pode ser usada para solucionar algumas das limitações atuais da tecnologia 3D, além de permitir novas funcionalidades que não são suportadas diretamente pela imagem 2D tradicional. No entanto, os dispositivos de visualização actuais não estão preparados para processar este tipo de conteúdo, o que significa que são necessários algoritmos de renderização para apresentar este tipo de conteúdo visual em versão 2D ou em versão 3D com múltiplas vistas.

No entanto, a qualidade visual do ponto vista da percepção do utilizador é altamente dependente da abordagem de renderização adotada. Portanto, a tecnologia de renderização LF requer avaliação de qualidade adequada com pessoas reais, já que não há maneira melhor e mais confiável de avaliar a qualidade deste tipo de algoritmos.

Neste contexto, esta dissertação tem como objetivo estudar, implementar e comparar diversos algoritmos e abordagens de renderização LF. A avaliação de desempenho é feita recorrendo a testes subjetivos de avaliação de qualidade para entender qual algoritmo que apresenta melhor desempenho em determinadas situações e a influência, em termos da qualidade subjetiva, de alguns parâmetros de input em certos algoritmos. Além disso, também é avaliada uma comparação de abordagens de renderização com focagem em apenas um plano versus renderização com focagem em todos os planos.

Palavras-Chave: Campo de Luz 4D; Algoritmos de Renderização 2D; Computação do Tamanho de *Patch* de Micro Imagens; Estimativa de Disparidade; Testes de Avaliação de Qualidade; Apresentação em Simultâneo de Comparação em Pares.

Abstract

Light Field (LF) technology, comprising visual data representations with huge amount of information, can be used to solve some of the current 3D technology limitations while enabling also new image functionalities not straightforwardly supported by traditional 2D imaging. However, current displays are not ready to process this kind of content, which means that rendering algorithms are necessary to present this type of visual content in 2D or 3D multi-view displays.

However, the visual quality experienced by the user is highly dependent on the rendering approach adopted. Therefore, LF rendering technology requires appropriate quality assessment tests with real people, as there is no better and reliable way to assess the quality of these type of algorithms.

In this context, this dissertation aims to study, implement, improve and compare various LF rendering algorithms and rendering approaches. Performance evaluation is done through subjective quality assessment tests aiming to understand which algorithm performs better in certain situations and the subjective quality impact of some of those algorithm parameters. Additionally, a comparison of single plane of focus versus all-in-focus LF rendering approaches is also evaluated.

Keywords: 4D Light Field; 2D Light Field Rendering; Micro Image Patch Size Computation; Disparity Estimation; Quality Assessment Tests; Pair Comparison Simultaneous Presentation.

Index

Acknowledgements .... **i**

Resumo .... **iii**

Abstract .... **v**

Index .... **vii**

List of Tables .... **xi**

List of Figures .... **xiii**

List of Abbreviations .... **xvii**

List of Symbols .... **xix**

Chapter 1 – Introduction **1**

1.1. Context and Motivation 1

1.2. Objectives 1

1.3. Research Questions 2

1.4. Research Method 2

1.5. Structure of the Dissertation 2

Chapter 2 – Review of Basic Concepts and Technologies **5**

2.1. Basic Concepts 5

2.1.1. Light Field 5

2.1.2. Radiance 6

2.1.3. Focal Length and Angle of View 7

2.1.4. Aperture and Depth of Field 8

2.1.5. 4D Light Field 9

2.1.6. Light Field Image 10

2.2. Light Field Camera Models 11

2.2.1. Basics on Traditional 2D Cameras 11

2.2.2. Basics on 4D Light Field Camera – Plenoptic Camera 1.0 11

2.2.3. Basics on 4D Light Field Camera – Plenoptic Camera 2.0 12

2.2.4. First generation Lytro Light Field Camera 13

2.2.5. Second generation Lytro Light Field Camera – Lytro Illum 14

2.2.6. Raytrix R42 Light Field Camera 14

2.3. Light Field Camera Arrays 15

2.4. Light Field Standardization Initiatives 15

2.4.1. MPEG-I 15

2.4.2. JPEG Pleno 15

Chapter 3 – Light Field Rendering **17**

3.1. Rendering Solutions Input 17

3.2. Texture Based 2D Image Rendering Solutions ... 18
3.2.1. Angle of View Based 2D Image Rendering (AV) ... 18
3.2.2. Single-Size Patch Based 2D Image Rendering (SSP) ... 19
3.2.3. Single-Size Patch Blending Based 2D Image Rendering (SSPB) ... 21
3.3. Disparity Based 2D Image Rendering Solutions ... 25
3.3.1. Disparity Estimation ... 25
3.3.2. Disparity Estimation – Minimizing Errors with α ... 29
3.3.3. Disparity Map Based 2D Image Rendering (DM) ... 32
3.3.4. Disparity Blending Based 2D Image Rendering (DB) ... 34
Chapter 4 – Developed Rendering Application ... 35
4.1. Technologies ... 35
4.2. Graphical User Interface ... 36
4.2.1. Overview ... 37
4.2.2. Rendering Algorithms Input Parameters ... 38
4.3. Block Diagram ... 40
Chapter 5 – Proposed Quality Assessment Tests ... 41
5.1. Questions ... 41
5.2. Methodology ... 42
5.3. Grading Scale ... 42
5.4. Test Participants ... 43
5.5. Developed Test Application ... 44
5.6. Test Material ... 44
5.7. Tests Cases ... 45
5.7.1. Algorithm Comparison: SSP vs SSPB ... 46
5.7.2. SSPB: The Influence of the Patch Size Parameter ... 47
5.7.3. SSPB: The Influence of the Window Size Parameter ... 48
5.7.4. Algorithm Comparison: DM vs DB ... 49
5.7.5. DB: The Influence of the Window Size Percentage Parameter ... 50
5.7.6. Algorithm Comparison: SSPB vs DB ... 52
5.7.7. Test Versions ... 53
Chapter 6 – Quality Assessment Results and Analysis ... 55
6.1. Algorithm Comparison: SSP vs SSPB ... 58
6.2. SSPB: The Influence of the Patch Size Parameter ... 60
6.3. SSPB: The Influence of the Window Size Parameter ... 67
6.4. Algorithm Comparison: DM vs DB ... 69
6.5. DB: The Influence of the Window Size Percentage Parameter ... 72
6.6. Algorithm Comparison: SSPB vs DB ... 74

Chapter 7 – Conclusions and Future Work 77
7.1. Conclusions.. 77
7.2. Future Work 79
Appendices 81
Appendix A – The relation between Sigma and Window Size 81
Appendix B – The impact of the α parameter in the Disparity Estimation 82
Bibliography... 87

List of Tables

Table 1: Seven level comparison grading scale [28]. 42
Table 2: Parameters of the generated images used for the SSP vs SSPB test. 46
Table 3: Parameters of the generated images used for the SSPB: influence of the PS test. 47
Table 4: Parameters of the generated images used for the SSPB: Influence of the WS test. 49
Table 5: Parameters of the generated images used for the DM vs DB test. 50
Table 6: Parameters of the generated images used for the DB: Influence of WSP test. 51
Table 7: Parameters of the generated images used for the SSPB vs DB test. 52
Table 8: Table with all the images existing in each one of the four test versions. 53
Table 9: Winning individual analysis with avg. algorithm confidence value and avg. overall score. 58
Table 10: Winning group analysis for SSP vs SSPB. 58
Table 11: Group analysis combinations results for Bike. 60
Table 12: Winning individual analysis results for Bike. 60
Table 13: Group analysis combinations results for Fredo. 61
Table 14: Winning individual analysis results for Fredo. 61
Table 15: Group analysis combinations results for Laura. 62
Table 16: Winning individual analysis results for Laura. 62
Table 17: Group analysis combinations results for Jeff. 62
Table 18: Winning individual analysis results for Jeff. 62
Table 19: Group analysis combinations results for Seagull. 63
Table 20: Winning individual analysis results for Seagull. 63
Table 21: Group analysis combinations results for Sergio. 64
Table 22: Winning individual analysis results for Sergio. 64
Table 23: Group analysis combinations results for Zhengyun1. 64
Table 24: Winning individual analysis results for Zhengyun1. 65
Table 25: Individual analysis results for Fredo. 67
Table 26: Individual analysis results for Jeff. 67
Table 27: Individual analysis results for Laura. 67
Table 28: Individual analysis results for Seagull. 68
Table 29: Individual analysis results for Sergio. 68
Table 30: Individual analysis results for Zhengyun1. 68
Table 31: Average individual analysis results. 68
Table 32: Winning individual analysis with avg. algorithm confidence value and avg. overall score. 69
Table 33: Winning group analysis for DM vs DB. 70
Table 34: Individual analysis results for Fredo. 72
Table 35: Individual analysis results for Jeff. 72
Table 36: Individual analysis results for Laura. 72
Table 37: Individual analysis results for Seagull. 73
Table 38: Individual analysis results for Sergio. 73
Table 39: Individual analysis results for Zhengyun1. 73
Table 40: Average individual analysis results. 73
Table 41: Winning individual analysis with avg. algorithm confidence value and avg. overall score. 74
Table 42: Winning group analysis for SSPB vs DB. 75

Table 43: Metrics extracted from the disparity estimation processes of the image Zhengyun1 with different values of α. 82
Table 44: Metrics extracted from the disparity estimation processes of the image Jeff with different values of α... 82
Table 45: Metrics extracted from the disparity estimation processes of the image Fredo with different values of α... 83
Table 46: Metrics extracted from the disparity estimation processes of the image Laura with different values of α... 83
Table 47: Metrics extracted from the disparity estimation processes of the image Seagull with different values of α.. 84
Table 48: Metrics extracted from the disparity estimation processes of the image Sergio with different values of α... 84

List of Figures

Figure 1: Plenoptic function [2], introduced by Adelson, in 1991. 5
Figure 2. Electromagnetic and visible spectrum. 6
Figure 3. Lens with a small focal length and a wide angle of view. 7
Figure 4. Lens with a big focal length and a small angle of view. 7
Figure 5. Examples of photographs taken from the same place with different focal lengths. 7
Figure 6. Different apertures and their measures. 8
Figure 7. Scheme of the depth of field with different apertures. 8
Figure 8. Depth of field of the same light field scene with different apertures. 9
Figure 9. Radiance along a ray remains constant if there are no objects blocking. 9
Figure 10. Light Field image / array of micro images example. 10
Figure 11. Traditional camera simplified optical components representation. 11
Figure 12. Light field camera 1.0 simplified optical components representation. 12
Figure 13. Light field camera 2.0 simplified optical components representation. 13
Figure 14. First generation Lytro Light Field Camera [14]. 13
Figure 15. Second generation Lytro Light Field Camera – Lytro Illum [14]. 14
Figure 16. Light field camera Raytrix R42 [15]. 14
Figure 17. Light field camera array [17]. 15
Figure 18. Light field image, with 6 micro images, schematic representation. 17
Figure 19. Micro image, with 9 pixels, schematic representation. 17
Figure 20. The 9 different rendered perspectives. 18
Figure 21. Rendering with the AV algorithm, with the developed software tool. 19
Figure 22. The 4 different rendered perspectives with a patch size of 2. 20
Figure 23. Rendering with the SSP algorithm, with the developed software tool. 21
Figure 24. Extracted block from a MI. 22
Figure 25. Graph with the influence of the sigma parameter in a Gaussian function. 22
Figure 26. Light field image with only 2 micro images. 23
Figure 27. Resulting image with overlapping pixels from neighbor blocks and invalid pixels. 23
Figure 28. Resulting image with overlapping pixels from neighbor blocks. 23
Figure 29. Determine final pixel color using (11), with a weight of 0.6 for color pixels and 0.4 for white pixels. 24
Figure 30. Algorithm output image with smooth transitions between blocks. 24
Figure 31. Rendering with the SSPB algorithm, with the developed software tool. 24
Figure 32. Light field image with 9 micro images of size 7 by 7. 25
Figure 33. Estimation of the disparity of micro image 1, 1 with their neighbors. 26
Figure 34. Estimation of the disparity of micro image 1, 1 with its right neighbor. 26
Figure 35. Disparity Map after the estimation of the disparity for each micro image and the calculations performed to obtain those values. 28
Figure 36. Disparity Map of the image Laura, row = 24, col = 24 and block size = 27. 28
Figure 37. Image Sequence used to estimate the disparity maps with different α values. 30
Figure 38. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Laura. 30
Figure 39. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Fredo. 30
Figure 40. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Jeff. 31

Figure 41. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Seagull. 31
Figure 42. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Sergio. 31
Figure 43. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Zhengyun1. 31
Figure 44. Fredo image using the DM algorithm with α = 0.0. 33
Figure 45. Fredo image using the DM algorithm with α = 0.0625. 33
Figure 46. Fredo image using the DB algorithm with α = 0.0625 and WSP = 2.0. 34
Figure 47. Print screen of the main desktop application. 36
Figure 48. Main desktop application divided in the five most important parts. 37
Figure 49. Block diagram of the main steps to execute a certain rendering algorithm. . 40
Figure 50. Snellen chart used before the assessment to test the visual acuity [28]. 43
Figure 51. Ishihara plates used before the assessment to test the color blindness [28]. . 43
Figure 52. Developed application used for the quality assessment tests. 44
Figure 53. Test material used for the assessment tests. 44
Figure 54. Fredo image rendered by SSP and SSPB, PS=11, WS=15. 46
Figure 55. Jeff image rendered by SSPB, PS=7, WS=9 and PS=11, WS=15. 48
Figure 56. Seagull image rendered by SSPB, PS=9, WS=9 and PS=9, WS=21. 49
Figure 57. Zhengyun1 image rendered by DM and DB, α=0.015625, WSP=1.5. 50
Figure 58. Laura image rendered by DB, α=0.0625, WSP=1.25 and α=0.0625, WSP=2.00. 51
Figure 59. Sergio image rendered by SSPB (PS=11, WS=23) and DB (α=0.0625, WSP=2.00). 52
Figure 60. Jeff image rendered by SSP (PS=11) and SSPB (PS=11, WS=15). 59
Figure 61. Seagull image rendered by SSP (PS=9) and SSPB (PS=9, WS=13). 59
Figure 62. This chart illustrates the winning grouped pairs percentage of this test case. 59
Figure 63. Bike image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15). 60
Figure 64. Fredo image with 3 images with different focal planes: Background (PS=9, WS=13), Middle (PS=11, WS=15), Foreground (PS=13, WS=19). 61
Figure 65. Laura image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13). 61
Figure 66. Jeff image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15). 62
Figure 67. Seagull image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13). 63
Figure 68. Sergio image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15). 63
Figure 69. Zhengyun1 image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13). 64
Figure 70. Images with 2 focal planes comparison. 65
Figure 71. Images with 3 focal planes comparison. 65
Figure 72. Images with different blur intensities comparison. 69
Figure 73. Sergio image rendered by DM (α=0.0625) and DB (α=0.0625, WSP=1.5). 70
Figure 74. Sergio image rendered by DM (α=0.0) and DB (α=0.0, WSP=1.5). 70
Figure 75. Zhengyun1 image rendered by DM (α=0.0625) and DB (α=0.0625, WSP=1.5). 71

Figure 76. This chart illustrates the winning grouped pairs percentage of this test case. 71
Figure 77. Images with different blur intensities comparison. 74
Figure 78. This chart illustrates the winning grouped pairs percentage of this test case. 75
Figure 79. This chart illustrates half of the Gaussian function to understand the relation between the pixel weight G(x) and half of the window size (x). 81
Figure 80. This chart illustrates the minimum estimated patch size for different disparity estimation processes, using different values of α. 85
Figure 81. This chart illustrates the maximum estimated patch size for different disparity estimation processes, using different values of α. 85
Figure 82. This chart illustrates the average estimated patch size for different disparity estimation processes, using different values of α. 85
Figure 83. This chart illustrates the median of the estimated patch size for different disparity estimation processes, using different values of α. 86
Figure 84. This chart illustrates the mode of the estimated patch size for different disparity estimation processes, using different values of α. 86
Figure 85. This chart illustrates the standard deviation of the estimated patch size for different disparity estimation processes, using different values of α. 86

List of Abbreviations

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional
5D	Five Dimensional
7D	Seven Dimensional
BSI	Back Side Illumination
CPU	Central Processing Unit
DoF	Depth of Field
FPS	Frames per Second
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HDR	High Dynamic Range
LF	Light Field
MI	Micro Image
MLA	Micro Lens Array
MP	Mega Pixel
PS	Patch Size
WS	Window Size
WSP	Window Size Percentage
BS	Block Size
PoV	Point of View
AV	Angle of view rendering
SSP	Single-Size Patch rendering
SSPB	Single-Size Patch Blending rendering
DM	Disparity Map rendering
DB	Disparity Blending rendering

List of Symbols

w	Width
h	Height
a	Disparity Estimation Process Regulation Input Parameter
K	Disparity Value
M	Number of Individual Pairs
N	Number of Participants
x_{ij}	Score Given for Image Pair *i* by Participant *j*
$\bar{u}(i)$	Average Score of Pair *i*
$\sigma(i)$	Standard Deviation of Pair *i*
$IS(i)$	Individual Score of Pair *i*
$INC(i)$	Individual Negative Confidence Value of Pair *i*
$IPC(i)$	Individual Positive Confidence Value of Pair *i*
$IC(i)$	Individual Confidence Value of Pair *i*
$IW(i)$	Individual Winner of Pair *i*
$\overline{GC}(k)$	Average Group Confidence Value of Group of Pairs *k*
$\overline{GS}(k)$	Average Group Score of Group of Pairs *k*
$\overline{GW}(k)$	Average Group Winner of Group of Pairs *k*
$\overline{AC}(alg)$	Average Algorithm Confidence Value of Algorithm *alg*

Chapter 1 – Introduction

1.1. Context and Motivation

Recent breakthroughs in light field (LF) technologies for acquiring and manipulating light fields, in areas such as optics and image processing, are creating a revolution in the way of how to acquire, manipulate, share and consume photographic images, a lot further than what is possible today with traditional photographs.

The higher amount of information acquired in 4D light field images, when compared with traditional 2D photographs, allows to change the focal plane, change the viewing perspective of the scene or manipulate the depth of field; functionalities that are not supported by traditional 2D imaging. These breakthroughs are opening new horizons to the human creativity associated to the act of “capture a photo”.

Current display technologies, however, are still not compatible with 4D LF image formats, which means that image rendering algorithms must be used to convert 4D LF content into traditional 2D images, compatible with existing displays.

Consumer-grade 4D LF displays will lead to an authentic breakthrough in terms of visual content consumption and immersive user experiences. However, high quality 2D LF rendering algorithms need to be developed and properly evaluated, which is the main motivation for the work developed in the scope of this dissertation.

In this context, the workplan defined for this dissertation consisted on the development of software tools that allow the user to process and manipulate 4D LF images in an interactive and creative way and on the evaluation of the images produced by these tools, using subjective quality assessment methodologies.

1.2. Objectives

The main goals of this dissertation are, therefore, to study, to implement and to evaluate the rendering process of 2D images from 4D LF content. To reach these goals, two desktop software tools were developed. The first software application, with an easy to use graphical user interface (GUI), allows the user to test different 2D LF rendering algorithms, with the possibility to display and save the rendered images.

The second software application, with a simpler interface, allows to realize quality assessment tests of the rendered 2D images generated in the first software application.

Using these two tools, it was possible to determine which of the implemented 2D rendering algorithms can provide better results for diverse types of 4D LF content and rendering scenarios.

1.3. Research Questions

After the work done in this dissertation, valid answers should be provided for the following research questions:

- What type of rendering algorithms will perform better in terms of subjective quality perception?
- What will be the impact of certain algorithm parameters in terms of subjective quality perception?

1.4. Research Method

The research method followed during the development of this dissertation is the Design Science Research Methodology Process Model (DSRMPM), which consists in the following six steps:

- Problem Identification;
- Define Objectives for a solution;
- Design and Development;
- Demonstration;
- Evaluation;
- Communication.

1.5. Structure of the Dissertation

This dissertation is organized into seven chapters that intend to reflect the different work phases until its conclusion.

After the Introduction, Chapter 2 covers the literature review, introducing the most relevant light field concepts, since it is the focus of this dissertation.

Chapter 3 reviews and explains different image rendering algorithms that allow the conversion from a 4D light field image into a 2D image. The chapter also suggests an improvement to one of those algorithms that proved to be able to reduce some visual artifacts, increasing the image quality.

Chapter 4 contains a description of the developed rendering application, its implemented algorithms, its architecture and core functionalities.

Chapter 5 describes the subjective quality assessment experiments performed, the used methodology and their setup. The analysis and the conclusions of those quality assessment experiments are presented and discussed in Chapter 6.

Finally, Chapter 7 presents the conclusions of this study and suggests topics for future work.

Chapter 2 – Review of Basic Concepts and Technologies

2.1. Basic Concepts

In this chapter, basic concepts such as light field, radiance, focal length, depth of field, 4D light field and light field imaging will be introduced.

2.1.1. Light Field

The idea that a light field should be treated as a field, such as the magnetic field, was first introduced in 1846 by Michael Faraday [1] in a lecture titled “Thoughts on Ray Vibrations”. Faraday defined a 7D function able to capture the evolution of the radiance of all the moving light rays that go through every point in space, in any angular direction, for any wavelength, through time. The function parameterization consisted in a (*x*, *y*, *z*) to specify the 3D spatial position, (θ, φ) to specify the angular direction, one dimension to represent the wavelength and one dimension to represent the time.

In 1991, Adelson [2] was able to simplify the number of the light field function parameters by turning the dimension for the wavelength and time into constants, describing a function able to represent a scene in a single wavelength and instant of time, named plenoptic function, as can be seen in Figure 1.

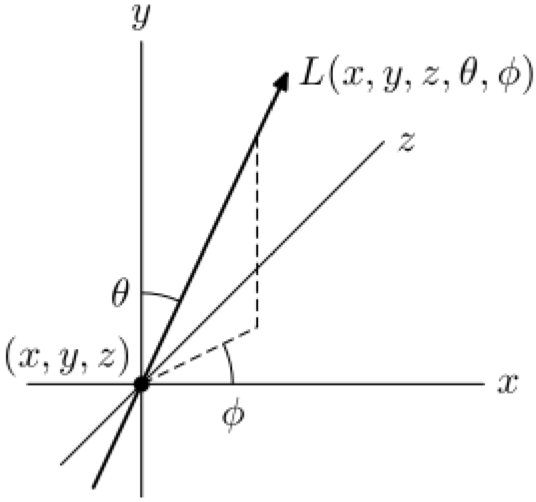
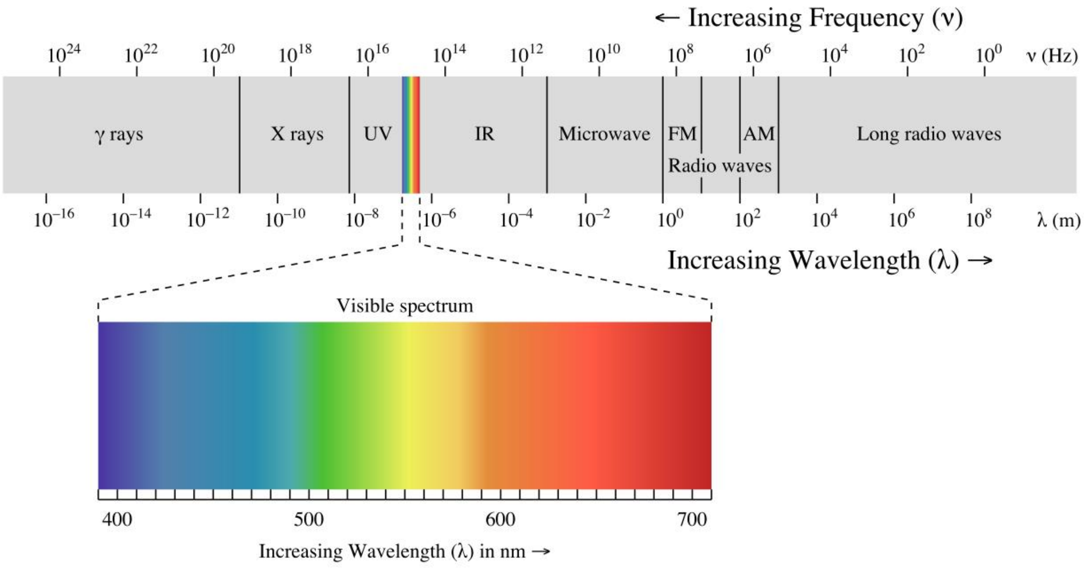


Figure 1: Plenoptic function [2], introduced by Adelson, in 1991.

2.1.2. Radiance

Electromagnetic field is carried by elementary particles named photons and the propagation of these particles is called electromagnetic propagation and it is made over electromagnetic waves.

Figure 2 describes the electromagnetic spectrum, where the visible spectrum corresponds to the portion of the electromagnetic field that the human eye can see. The visible spectrum, also known as light rays, correspond to every wavelength from 390 to 700 nm.



← Increasing Frequency (ν)
10^{24} 10^{22} 10^{20} 10^{18} 10^{16} 10^{14} 10^{12} 10^{10} 10^{8} 10^{6} 10^{4} 10^{2} 10^{0} ν (Hz)
γ rays
X rays
UV
IR
Microwave
FM
AM
Radio waves
Long radio waves
10^{-16} 10^{-14} 10^{-12} 10^{-10} 10^{-8} 10^{-6} 10^{-4} 10^{-2} 10^{0} 10^{2} 10^{4} 10^{6} 10^{8} λ (m)
Increasing Wavelength (λ) →
Visible spectrum
400
500
600
700
Increasing Wavelength (λ) in nm →

Figure 2. Electromagnetic and visible spectrum.

Radiance [3] corresponds to the amount of electromagnetic energy that is emitted, reflected, transmitted or received by a certain surface with a specific angle and direction.

Light field cameras use a 2D radiance sensor to be able to capture the amount of light hitting a certain area of the sensor that is measured in (1).

$$L = \frac{d^2\Phi}{dAd\Omega \cos\theta} \tag{1}$$

In the above equation, *L* is the radiance of a surface, *d* is the partial derivative symbol, φ is the radiant flux emitted, reflected, transmitted or received, Ω is the solid angle and *A cos* θ is the projected area.

2.1.3. Focal Length and Angle of View

The focal length [4][5][6], usually represented in millimeters (mm), it is a calculation of an optical distance from the point where light rays converge to form a sharp image of an object to the digital sensor at the focal plane in the camera. Angle of view is the visible extent of the scene captured by the image sensor/film of the camera, which means the bigger the angle of view is, the bigger the area captured is.

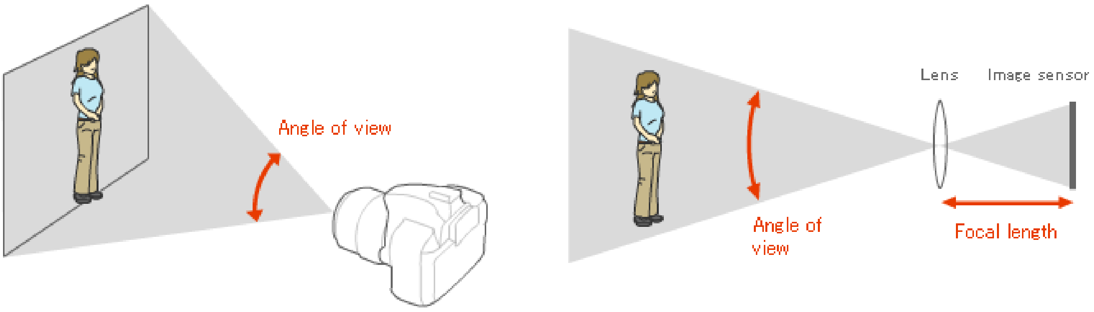


Figure 3. Lens with a small focal length and a wide angle of view.

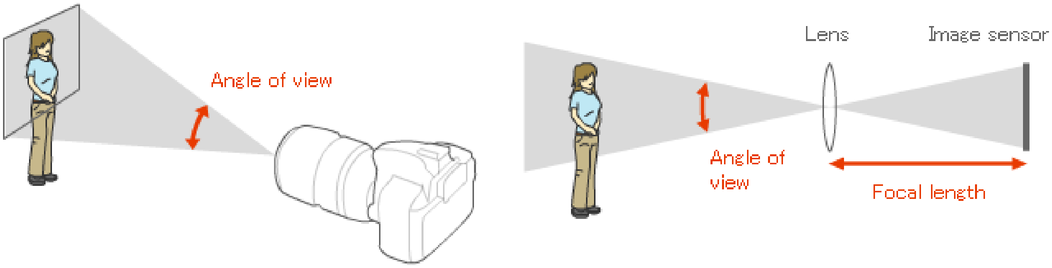


Figure 4. Lens with a big focal length and a small angle of view.

As illustrated by Figure 3 and 4, the angle of view is dependent of the focal length. As the focal length increases, the smaller the angle of view is, and vice versa.

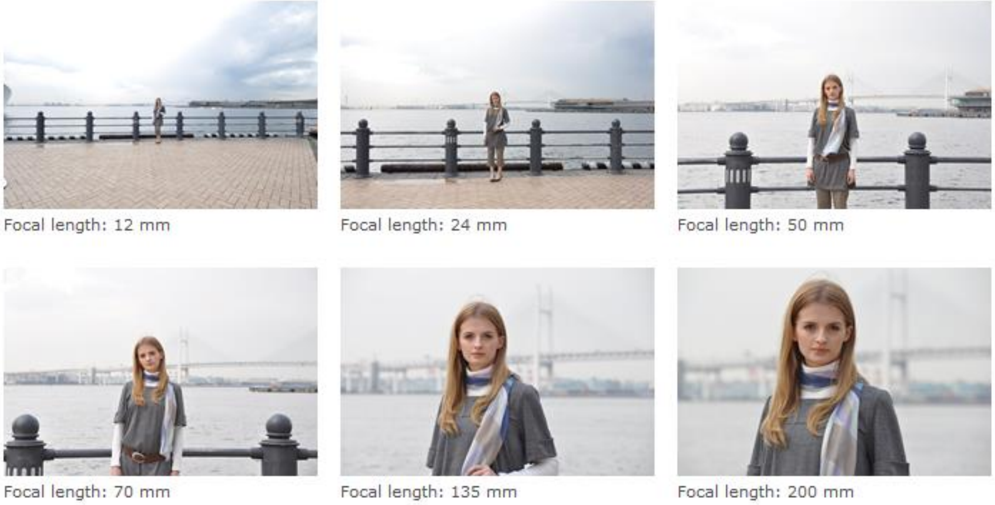


Figure 5. Examples of photographs taken from the same place with different focal lengths.

2.1.4. Aperture and Depth of Field

Aperture [7][8] is one of the main components in a camera optical system as it defines the size of the opening in the lens that can be adjusted to control the amount of light that reaches the camera sensor. The size of the aperture is measured in millimeters (mm), but aperture is normally described in f-stops and moving from one f-stop to the next doubles or halves the size of the amount of opening in your lens and consecutively the amount of light getting through, as can be seen in Figure 6.

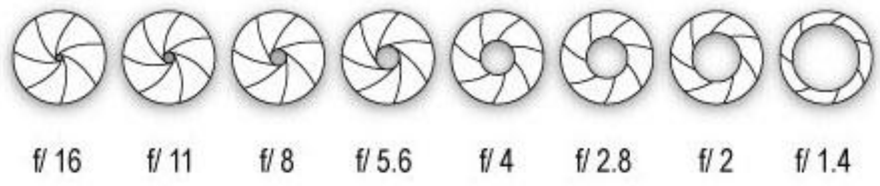


Figure 6. Different apertures and their measures.

Depth of Field [9][10] corresponds to the range of distance that appears acceptably sharp focus in a photograph. Some areas before and after the optimal focal plane are also going to be in focus.

The depth of field outside the sharp focus region will have a gradual blurry transition, even if this fact can be seed by the human eye. Images with small areas of focus are called shallow depth of field, while images with a larger area of focus are called deep depth of field, as can be seen in Figure 7.

The selection of the best depth of field for a certain situation might may vary according to the photographer. The selection of depth of field is therefore a subjective choice.

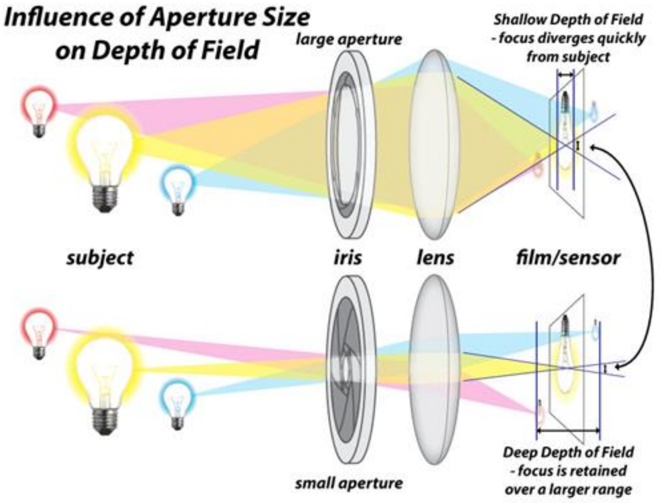

Influence of Aperture Size on Depth of Field
large aperture
Shallow Depth of Field - focus diverges quickly from subject
subject
iris
lens
film/sensor
small aperture
Deep Depth of Field - focus is retained over a larger range

Figure 7. Scheme of the depth of field with different apertures.

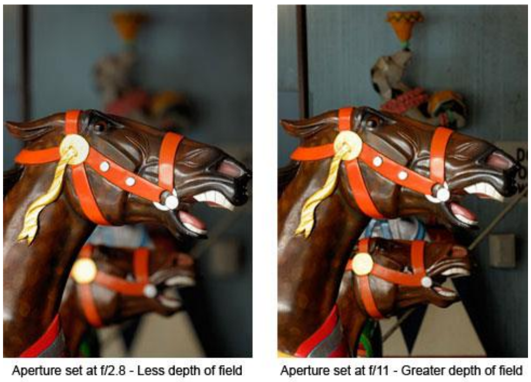

Aperture set at f/2.8 - Less depth of field
Aperture set at f/11 - Greater depth of field

Figure 8. Depth of field of the same light field scene with different apertures.

Figure 8 demonstrates the depth of field of two images of the same light field scene with two different apertures. The image on the left has a bigger aperture which result in less depth of field. The image on the right has a smaller aperture which results in a greater depth of field. The higher the aperture is, the bigger the opening in the lens is, the greater the depth of field, and the sharper the background is.

2.1.5. 4D Light Field

As presented before, the plenoptic function has 5 dimensions. It would be extremely hard to capture all the light ray information with current technology, as an extremely large number of sensors and storage devices would be necessary, as explained in [2].

Radiance along a light ray remains constant if there are no objects blocking, as can be seen in Figure 9. Which means the capture of all 5D light rays would have redundant information. That redundant information corresponds to one dimension, reducing the plenoptic 5D parameter function into a 4D function, this discovery was made by Parry Moon and named it photic field in 1981. Computer graphics researchers Levoy and Gortler named it 4D light field in 1996 or Lumigraph respectively, in 1996.

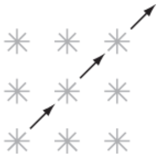


Figure 9. Radiance along a ray remains constant if there are no objects blocking.

This way the 4D light field image needs a (*x*, *y*) coordinate to specify the 2D spatial position and (θ, φ) coordinate to specify the angular direction.

Traditional 2D video needs a (*x*, *y*) coordinate to specify the 2D spatial position and one dimension to represent the time instant.

Traditional 2D image needs only a (*x*, *y*) to specify the 2D spatial position.

2.1.6. Light Field Image

The Light Field Image also known as plenoptic image, holoscopic image or integral image can be presented as a 2D array of 2D micro images. A possible way to store the 4D information of the light field is presented in Figure 10, in this example all the light field data is stored in the same image. The background image corresponds to a light field image and the image in foreground represents a small highlighted part of the 2D array, where 2D micro images, associated to different perspectives of the light field scene, are placed side by side.

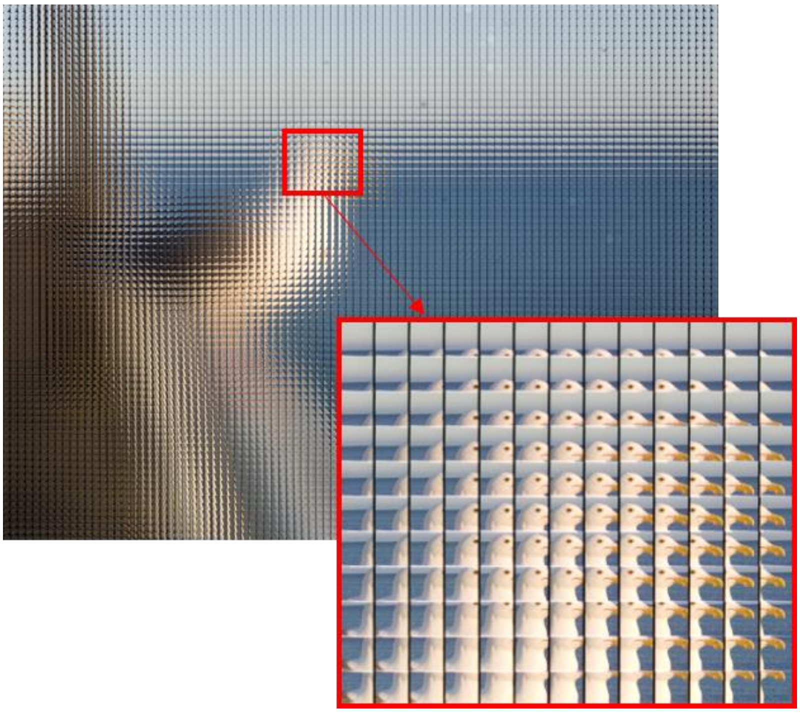


Figure 10. Light Field image / array of micro images example.

Different setups can be used, micro images resolution varies depending on the size of the micro lenses and the content displacement varies depending on the distance from a certain micro lens to their adjacent micro lenses.

This way, micro images resolution is predefined and content displacement from neighboring points of view will depend on the setup used inside the light field camera, used to capture the light field scene.

2.2. Light Field Camera Models

This section starts by introducing the basics of the traditional 2D light field cameras followed by a presentation of the basics behind 4D light field cameras. Also, some of the more representative light field cameras, from Lytro and Raytrix manufacturers, will be briefly reviewed.

2.2.1. Basics on Traditional 2D Cameras

As can be observed in the simplified representation depicted in Figure 11, in a traditional camera light rays (represented as arrows) with multiple orientations go through the lens and hit the sensor. The information captured by each pixel of the sensor corresponds to the light intensity from multiple rays hitting the corresponding area of the sensor, originating this way an intensity value of a pixel in the captured image. This process did not change much since the first digital cameras were invented.

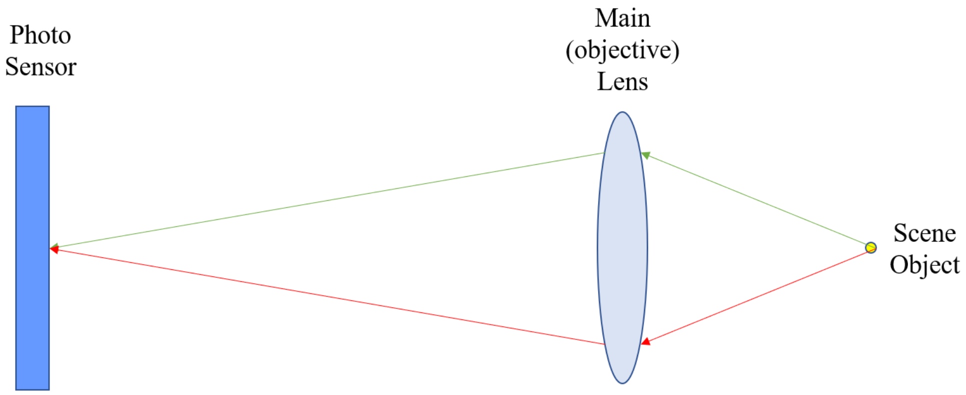


Figure 11. Traditional camera simplified optical components representation.

2.2.2. Basics on 4D Light Field Camera – Plenoptic Camera 1.0

Compact Light field cameras are relatively new. Their objective is to capture as much information of the light field scene as possible. To allow this task, a new component, known as micro lens array (MLA), has been added to the camera design, as can be seen in Figure 12. As explained in [11][12], light rays with different intensities and orientations go through the lens, and instead of hitting the sensor like in the traditional cameras, they will go through a certain micro lens, making it possible to store light rays with different orientations, in different parts of the sensor. This process will allow to know the light intensity and the orientation of every single ray. Light field images tend to look like a grid of micro images (MI) and have a lot more information, since they capture the intensity of light in a scene and the direction that the light rays are traveling in space. In this camera, the focal length *f* is the distance from the sensor to the MLA is fixed. The micro lenses are focused at infinity and completely defocused from the main lens image plane which result in blurry micro images.

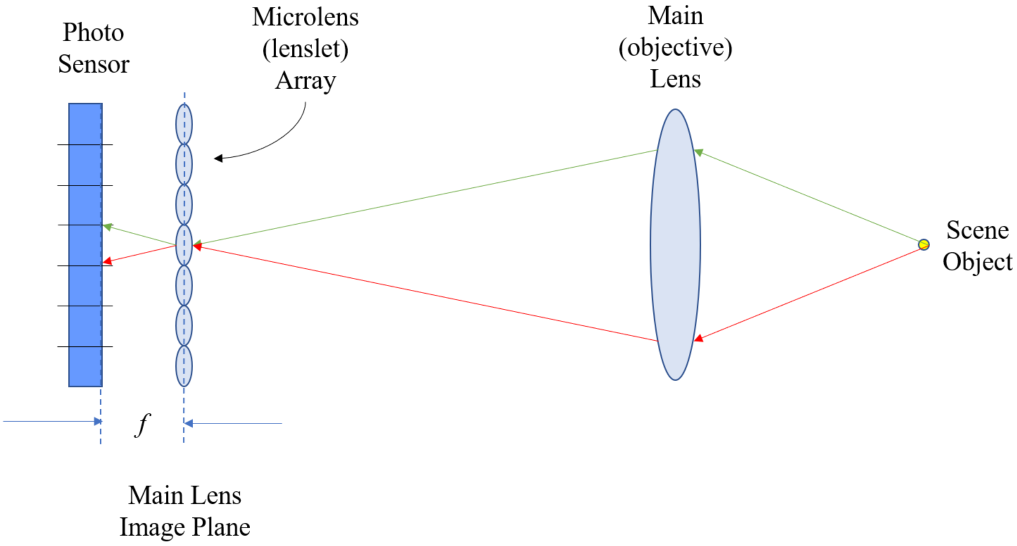


Figure 12. Light field camera 1.0 simplified optical components representation.

2.2.3. Basics on 4D Light Field Camera – Plenoptic Camera 2.0

As illustrated in Figure 13 this type of camera is an improvement from the Light Field Camera 1.0, thus includes all its elements. As described in [13] the main (objective) lens creates its image at a plane which is called image plane. In this camera the distance from the sensor to the MLA is fixed at a constant distance of *b*, the MLA must be at a certain distance *a* from the image plane in order to obtain a certain fixed focal length *f*, this relation is described by equation:

$$\frac{1}{a}+\frac{1}{b}=\frac{1}{f} \tag{2}$$

With this arrangement, the micro lenses satisfy the lens equation, making them exactly focused on the main lens image plane, resulting in sharp and inverted micro images.

New possibilities like changing the depth of field, changing the point of view and perform depth estimation are now possible to do with a single image, creating a lot of new possibilities to photographers and to the image processing community.

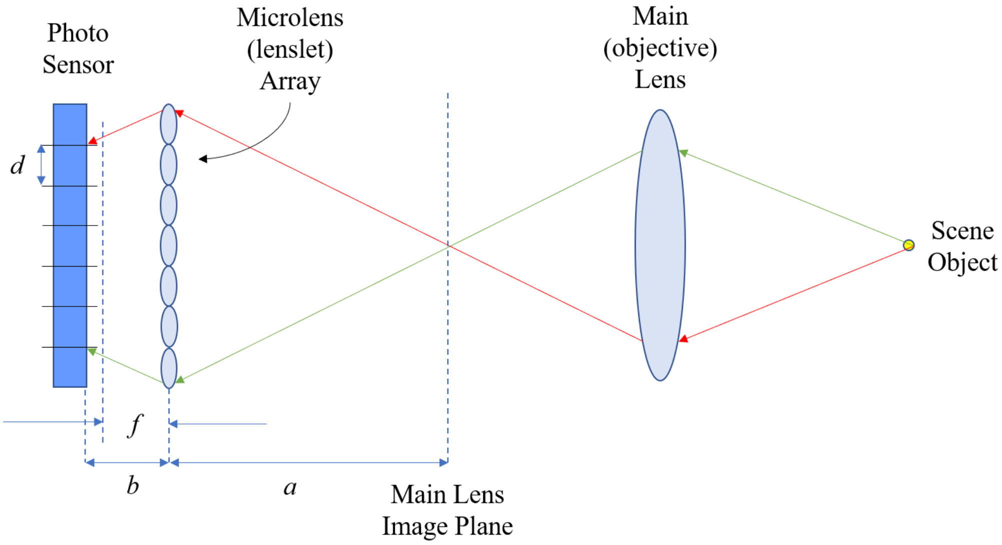


Figure 13. Light field camera 2.0 simplified optical components representation.

2.2.4. First generation Lytro Light Field Camera

Lytro, Inc is an American company founded in 2006 by Executive Chairman Ren Ng, whose PhD research on the field of computational photography / light field imaging won the prize for the best thesis in computer science from Stanford University in 2006.

Lytro was the first company releasing a first-generation light field camera into the consumer electronics market, in 2012. More information about this light field 1.0 camera with 11 megaray sensor camera can be found in [14].

Figure 14. First generation Lytro Light Field Camera [14].

2.2.5. Second generation Lytro Light Field Camera – Lytro Illum

In 2014 their second-generation camera named Lytro Illum was also a light field 1.0 camera and was released with a 40 megaray sensor, with a lot more powerful processor and with a display overlay that shows the photographer the relative focus of all objects in the frame, and which elements are re-focusable.

Figure 15. Second generation Lytro Light Field Camera – Lytro Illum [14].

Lytro is expanding into cinematography, virtual reality and augmented reality areas and released a new camera called Lytro Immerge whose main innovation is the way of recording light field information [14].

2.2.6. Raytrix R42 Light Field Camera

Raytrix GmbH [15], in a German company founded in 2009. Since 2010 they are selling 3D cameras for industrial applications and research purposes. With a team of 15 people, their main goal is to improve the quality of their light field cameras and to explore new application areas.

At least seven different light field cameras are presented in Raytrix website [15], some of them are aimed for video, other for still imaging.

Figure 16. Light field camera Raytrix R42 [15].

2.3. Light Field Camera Arrays

A different way to capture the light field is using different arrangements of camera arrays [16], as illustrated in Figure 17. The reduced cost of cameras makes it possible to replace the monocular camera with an array of cameras in certain situations, as presented in [17]. Different arrangements will change the dynamic range, the resolution, seeing through occlusions and in the depth estimation of the scene.

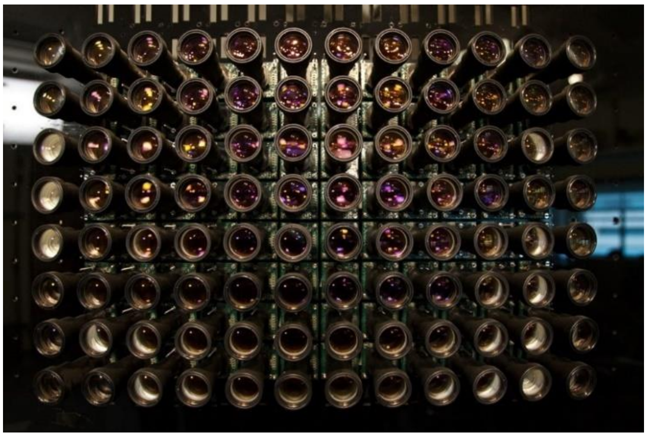


Figure 17. Light field camera array [17].

2.4. Light Field Standardization Initiatives

Light Field imaging has currently risen as a feasible and prospective technology for future image and video applications. New standardizations for light field imaging are emerging [18][19].

2.4.1. MPEG-I

MPEG-I is the name of the new work that was started by the Moving Picture Experts Group (MPEG) which targets future immersive applications. The goal of this new standard is to enable various forms of audio-visual immersion, including panoramic video with 2D and 3D audio, with various degrees of true 3D visual perception.

2.4.2. JPEG Pleno

JPEG Pleno standardization was launched by the Joint Photographic Experts Group (JPEG) and aims to provide a standard framework for representing new imaging modalities, like texture-plus-depth, light field, point cloud and holographic imaging.

Chapter 3 – Light Field Rendering

This chapter introduces the developed lenslet rendering solutions, for plenoptic 2.0 cameras, that were based in some references from the state of the art and allow to render 4D LF content into 2D displays. All the algorithms were implemented from scratch, some of the difficulties and decisions made in the development process will be explained. Other implemented ideas to improve the algorithms quality will also be described.

3.1. Rendering Solutions Input

This section introduces the schematic representations of a light field image and of a micro image that will be used in further sections, where different rendering solutions that allow to render 4D light field contents into 2D displays will be presented.

Light Field technology allows a user to take a 4D photo, convert to 2D using a rendering algorithm such as the ones proposed in [19]-[23], change the light field perspective of the scene, estimate the depth of the scene and manipulate the depth of field.

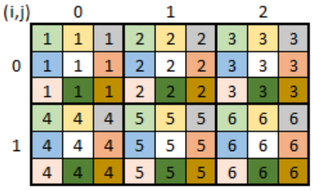


Figure 18. Light field image, with 6 micro images, schematic representation

Figure 18 is a schematic representation of a 4D light field image that can also be a 2D array of micro images, where each micro image is associated to a certain position (*i*, *j*). Each micro image is the result of several different light rays that went through a certain micro lens and captured by the radiance sensor. The valid values for *i* and *j* are:

$$\begin{gathered} 0 \leq i < MLA.height \\ 0 \leq j < MLA.width \end{gathered} \tag{3}$$

where *MLA.height* and *MLA.width* represent the vertical and horizontal dimensions of the micro lens array, respectively.

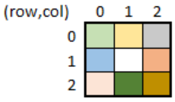


Figure 19. Micro image, with 9 pixels, schematic representation.

As can be seen in Figure 19, inside each micro image (MI) there are several pixels, where each pixel is associated to a certain position (*row*, *col*), the valid *row* and *col* values are described in (4).

$$\begin{gathered} 0 \leq row < MI.height \\ 0 \leq col < MI.width \end{gathered} \tag{4}$$

3.2. Texture Based 2D Image Rendering Solutions

3.2.1. Angle of View Based 2D Image Rendering (AV)

The idea behind this algorithm [19] starts by choosing the scene perspective to be seen, in other words, a valid position (*row*, *col*) inside a micro image, must be chosen. To create the new image, which will have the same size as the micro lens array (MLA), the intensity value of the pixel in the selected position (*row*, *col*) will be extracted from each micro image (*i*, *j*) and stored in the corresponding position (*i*, *j*) of the new image. This way, a certain perspective (*row*, *col*) of the light field scene will be rendered.

Assuming Figure 18 is the light field image this algorithm is using as input, it will be able to render the 9 different and possible perspectives represented in Figure 20, as this is the number of pixels inside each MI, with the size of the MLA, in this case, 2 rows and 3 columns.

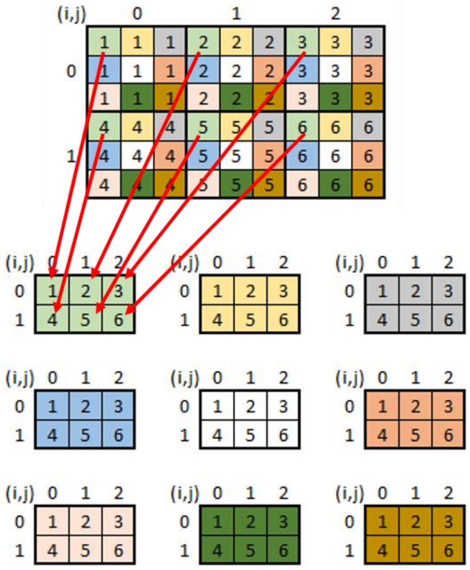


Figure 20. The 9 different rendered perspectives.

Advantages: Since only one pixel is extracted from each MI to render one of the perspectives, this algorithm has the high number of perspectives that can be rendered.

Disadvantages: Since only one pixel is extracted from each MI to render one of the perspectives, the resolution of the rendered images will be always equal to the size of the

MLA, and typically these values are very small compared to images seen daily all over the internet.

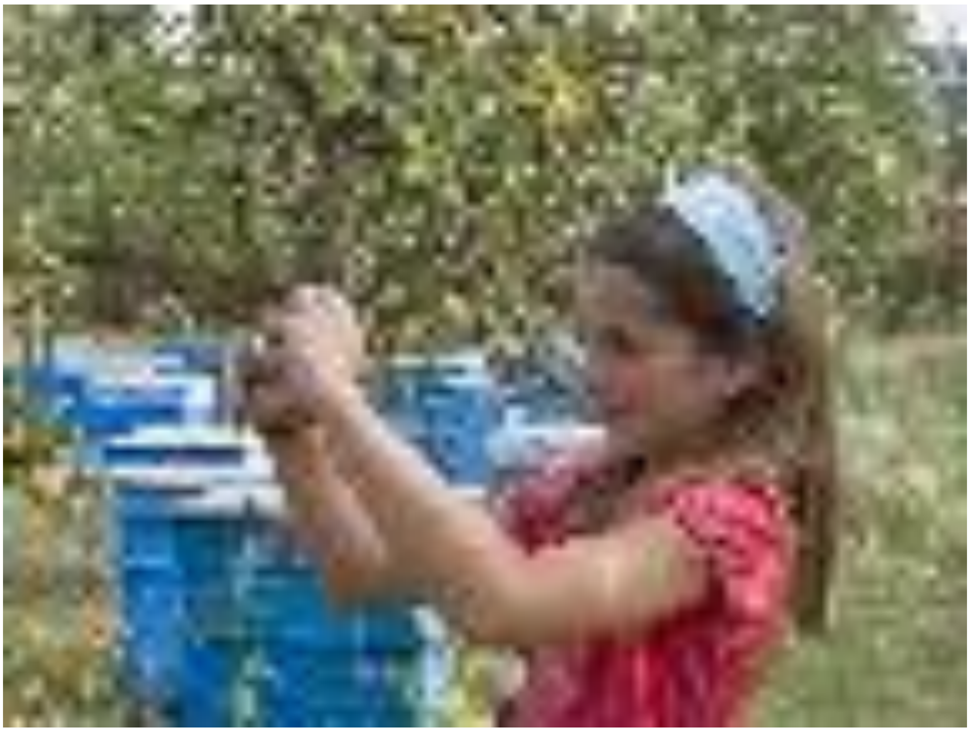


Figure 21. Rendering with the AV algorithm, with the developed software tool.

As can be seen by Figure 21, a generated perspective from the light field scene from this rendering algorithm have been up scaled in size by a factor of five due to the small resolution of these output images. The images have a bad quality and are very pixelated, being hard to understand the details of their content.

3.2.2. Single-Size Patch Based 2D Image Rendering (SSP)

This algorithm was developed by Todor Georgiev [20], in 2010, and is very similar to the one presented in the previous section but instead of extracting a single pixel from each MI, will extract a squared block of pixels with a fixed size, referred as patch size (PS), which can assume the values:

$$1 < PS < \min(MI.height, MI.width) \tag{5}$$

A position (*row*, *col*) must be selected and will correspond to the top left pixel of the block that will be extracted. The valid values for the *row* and *col* vary depending of the selected patch size, as defined in:

$$\begin{gathered} 0 \le row < MI.height\ - PS \\ 0 \le col < MI.width\ - PS \end{gathered} \tag{6}$$

The patch size parameter which corresponds to the number of pixels extracted from each MI, which means the bigger the patch size is, the bigger the resolution will be, making the resolution of the resulting image proportional to the value of the patch size as described by the following equation:

$$\begin{gathered} Perspective.height\ = MLA.height \times PS \\ Perspective.width\ = MLA.width \times PS \end{gathered} \tag{7}$$

The number of different perspectives will also be dependent of the same parameter, as the bigger the patch size is, the smaller the number of different perspectives that can be rendered will be as shown in equation:

$$\#Perspectives\ = \left[MI_{Height} - (Path_{Size} - 1)\right] \times \left[MI\_Width - (Path\_Size\ -\ 1)\right] \tag{8}$$

Assuming Figure 18 is the light field image this algorithm is using as input and the patch size is equal to 2, it will be able to render 4 different perspectives as defined in (8) and represented in Figure 22, with the double of the MLA size, as defined in (7).

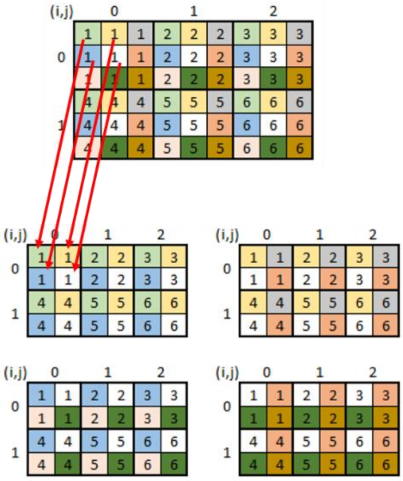


Figure 22. The 4 different rendered perspectives with a patch size of 2.

Advantages: The rendered images will no longer have small resolutions as the ones obtained with the AVe algorithm, because this algorithm extracts more than one pixel per MI.

Disadvantages: The bigger the patch size is, a bigger number of pixels will be extracted from each MI, which will reduce the number of different rendered perspectives.

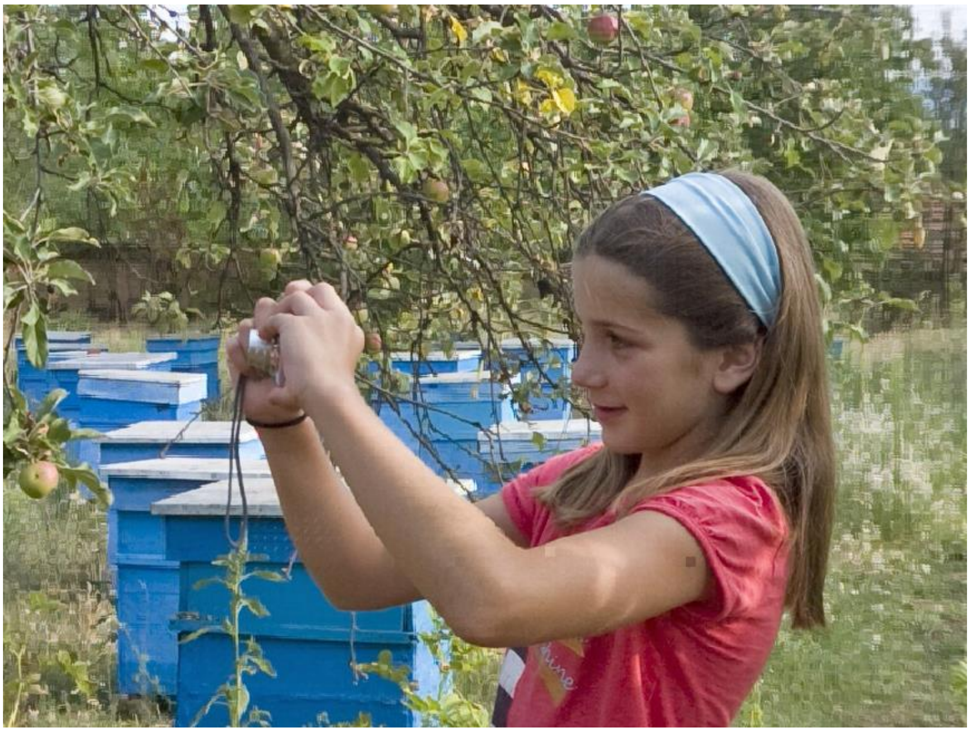


Figure 23. Rendering with the SSP algorithm, with the developed software tool.

The rendering in Figure 23 has a lot better quality when compared to the rendering from the AVe algorithm. The transitions between neighbor squared blocks are clear, especially in the background. Looking to the girl arm there was a problem with one of the micro lenses.

3.2.3. Single-Size Patch Blending Based 2D Image Rendering (SSPB)

This algorithm is identic to the one presented in the previous section, was developed by Todor Georgiev and Andrew Lumsdaine and is described in [21][22]. The patch size parameter already introduced, will represent, once again, the size of the block of pixels that will be extracted from each MI. A selected valid position (*row, col*) will mark the starting point where the block will be extracted. Instead of joining side by side the extracted blocks in the new image, a process called blending will be used, where the main goal is to smooth the transitions by calculating the average between pixels, with different weights, from neighbor blocks.

To be able to make this blending process a few more parameters will be needed. A new parameter like the patch size will be needed, which will be named window size (WS). In this algorithm the patch size and window size will be assumed as both being odd numbers and the following conditions must be valid:

$$1 < PS \leq WS < \min(MI.height, MI.width) \tag{9}$$

In Figure 24 a representation of a block extracted from a MI is presented with a patch size of 3 and a window size of 5.

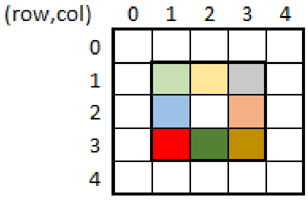


Figure 24. Extracted block from a MI.

The first step is to associate each pixel of the extracted block with a certain weight, this means the further away a pixel is from the center of the block, the smallest the weight of that pixel will have. To make this possible, the weight of each pixel is given by:

$$Weight(row, col) = 1 \times e^{-\frac{(row-centerRow)^2}{2\times sigma_Y{}^2}} \times e^{-\frac{(col-centerCol)^2}{2\times sigma_X{}^2}} \tag{10}$$

In equation (10), *weight* corresponds to the weight of a certain pixel, given by the influence of horizontal and vertical gaussian function, (*row*, *col*) correspond to the position of a certain pixel, (center*Row*, *centerCol*) correspond to the position of the pixel in the center of the block (centered in the middle of the extracted block of size WS), in this case position (*centerRow* = *2*, *centerCol* = *2*) and (*sigmaY*, *sigmaX*) have a big influence in the format of the horizontal and vertical gaussian functions, used in equation (10), as can be seen in Figure 25. For more information about the relation between the WS and the components of the *sigma* parameter consult Appendix A.

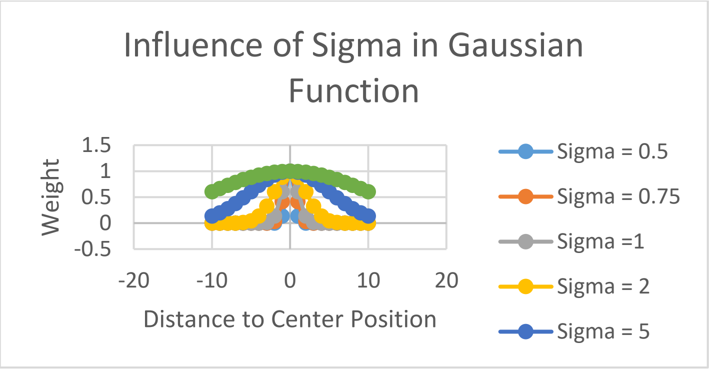
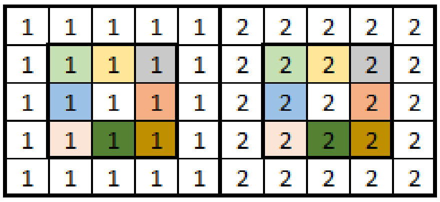


Figure 25. Graph with the influence of the sigma parameter in a Gaussian function.

Figure 25 shows the influence of the parameter sigma in the format of the Gaussian function as the smaller the sigma is, the fastest the weight of the pixels closest to the center will decrease.

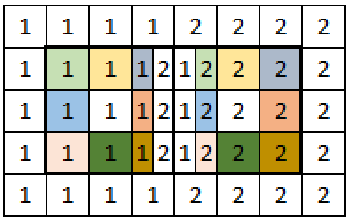


1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2

Figure 26. Light field image with only 2 micro images.

Assuming the light field images used by this algorithm was the one presented in Figure 26, with a patch size of 3 and a window size of 5, a resulting image is returned with the size of the chosen patch size times the number of micro images, which means an image with 3 rows and 6 columns.

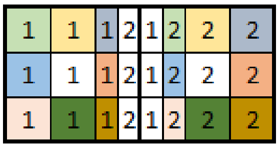
As the output image will have the size of multiple patch sizes, some of the pixels inside the window size, presented as the white pixels in the margins, will overlap with pixels of the neighbor micro image, as presented in Figure 27.



1	1	1	1	2	2	2	2
1	1	1	1 2	1 2	2	2	2
1	1	1	1 2	1 2	2	2	2
1	1	1	1 2	1 2	2	2	2
1	1	1	1	2	2	2	2

Figure 27. Resulting image with overlapping pixels from neighbor blocks and invalid pixels.

As the white pixels after the margin will be removed as they won't have a pixel position inside the output image, as can be seen in Figure 28.



1	1	1 2	1 2	2	2
1	1	1 2	1 2	2	2
1	1	1 2	1 2	2	2

Figure 28. Resulting image with overlapping pixels from neighbor blocks.

As shown in Figure 28 there are some pixels that have contributions of *n=2* sub pixels from the input image (the *n* can vary depending on the relation between PS and WS), in these situations the following equation should be applied:

$$pixelColor = \frac{\sum_{i=0}^{n}(color_i \times weight_i)}{\sum_{i=0}^{n} weight_i} \tag{11}$$

To calculate the average of all the neighbor contributions with different weights as can be seen in Figure 29 and place the result as the final pixel color, resulting in an output image illustrated in Figure 30.

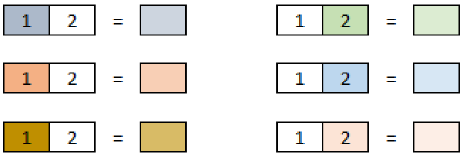


Figure 29. Determine final pixel color using (11), with a weight of 0.6 for color pixels and 0.4 for white pixels.

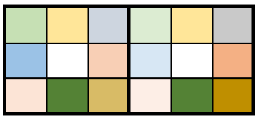


Figure 30. Algorithm output image with smooth transitions between blocks.

Advantages: The smooth transitions between blocks makes harder for the user to see artifacts when compared to the SSP algorithm where the outline of each block could be easily seen.

Disadvantages: Comparing this algorithm with AV and SSP, this is the algorithm with less perspectives due the fact that is the one taking more information (patch size and window size) from the input image to render a perspective of the light field scene. This is also the less efficient algorithm as is the most complex one so far.

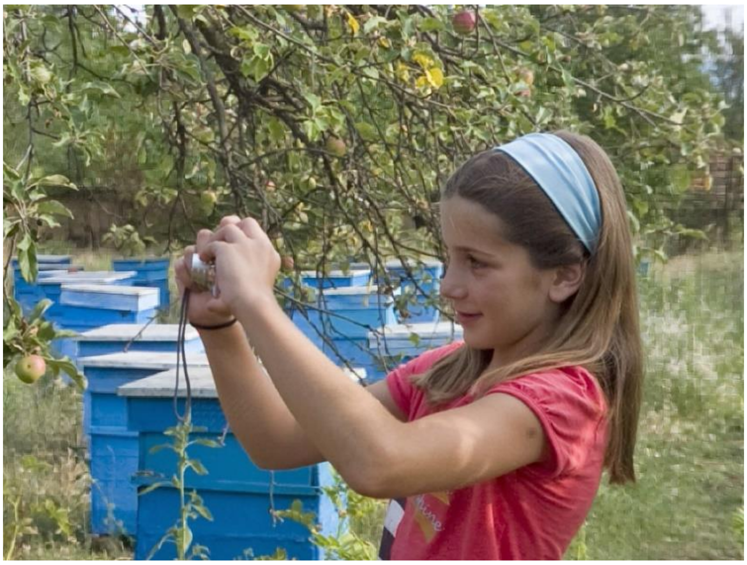


Figure 31. Rendering with the SSPB algorithm, with the developed software tool.

The rendering in Figure 31 is the one with better quality between the three algorithms presented. The transitions between neighbor squared blocks are less clear, except in the background. Looking to the girl arm there was a problem with one of the micro lenses, but with the blending process the reason is no longer clear if that is a problem or if it is a spot on her skin.

3.3. Disparity Based 2D Image Rendering Solutions

In this section, two models of how to estimate the disparity map are presented and then two algorithms that use the disparity map estimated to get an all-in-focus rendered 2D image are introduced.

3.3.1. Disparity Estimation

This disparity estimation model is explained by T. Georgiev and A. Lumsdaine in [20].

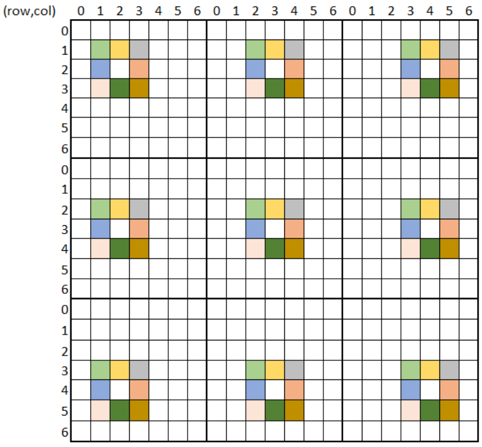


Figure 32. Light field image with 9 micro images of size 7 by 7.

If our 4D light field image is the one presented in the Figure 32, the first thing to do is to specify the *row*, *col* and the *block size* (BS) of pixels to compare with all the blocks of the adjacent micro images in order to find out the block with the most similarities, and that corresponds to the displacement value between the two micro images.

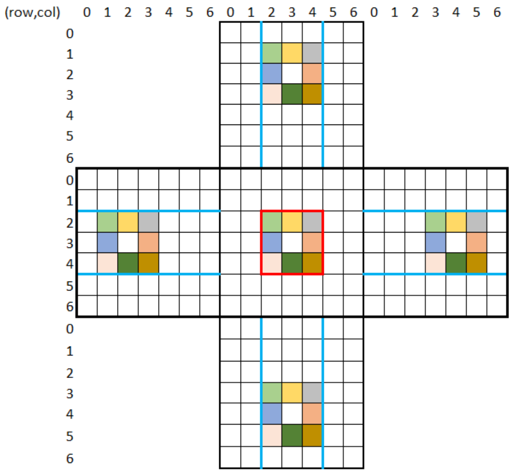


Figure 33. Estimation of the disparity of micro image 1, 1 with their neighbors.

Assuming the point of view corresponds to *row* = 2, *col* = 2 and the *block size* = 3, as presented in Figure 33, where for each micro image, a comparison between the red block of the micro image of which its disparity is tried to be estimated and all the blocks in the same vertical line for the top and bottom micro images and with all the blocks in the same horizontal line for the left and right micro images, represented by blue lines in Figure 33.

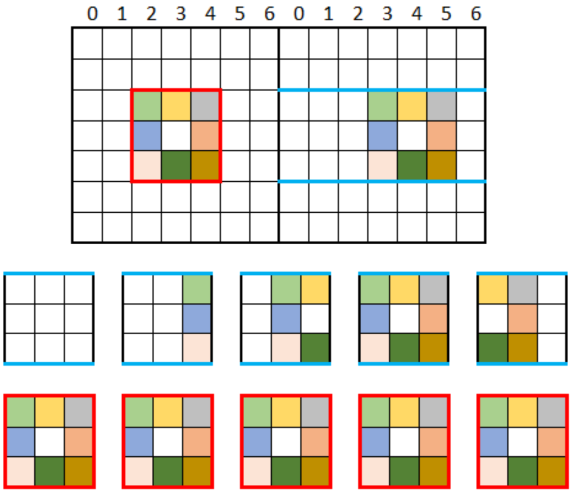


Figure 34. Estimation of the disparity of micro image 1, 1 with its right neighbor.

Figure 34 shows how the disparity value is calculated, representing all the possible comparisons that must be done in this case. For each comparison of blocks the following equation must be performed:

$$comparison = \frac{\frac{\sum_{r=0}^{rows}\sum_{c=0}^{cols}(A_{rc}-B_{rc})^2}{r\times c}}{255} \tag{12}$$

The equation will calculate the square differences of the blocks *A* and *B*, calculate the square difference value per pixel and divide it by *255* to get a value between one and two hundred and fifty-five, this way, the equation will return two hundred and fifty-five if the two blocks are completely different and zero if the two blocks are equal.

In this case, when a comparison of the red square is made with the five possible blocks from the right micro image, a calculation of the column value of the block that obtained the highest comparison value with the red block is needed. After comparing the red block with all the possible blocks, the equation:

$$K_x = \left|col - col_{\max\_comparition}\right| \tag{13}$$

should be used to calculate the horizontal disparity value between the two micro images. To calculate the vertical disparity value between the two micro images, the following equation should be used:

$$K_y = \left|row - row_{\max\_comparition}\right| \tag{14}$$

For this case, after performing all the comparisons, the *column = 3* would have the lowest comparison value, so *Kx = |2-3|=1.*

After performing the same logic to all the valid neighbors, the following equation should be used:

$$K = \frac{Kx_{left} + Kx_{right} + Ky_{top} + Ky_{bottom}}{\#\ of\ valid\ neighbors} \tag{15}$$

to calculate the average disparity value for each one of the nine micro images, as illustrated in Figure 35.

$$K_{00} = \frac{Kx_{right} + Ky_{bottom}}{2} = \frac{|2-3| + |2-3|}{2} = \frac{2}{2} = 1$$

$$K_{01} = \frac{Kx_{left} + Kx_{right} + Ky_{bottom}}{3} = \frac{|2-1| + |2-3| + |2-3|}{3} = \frac{3}{3} = 1$$

$$K_{02} = \frac{Kx_{left} + Ky_{bottom}}{2} = \frac{|2-1| + |2-3|}{2} = \frac{1+1}{2} = 1$$

$$K_{10} = \frac{Kx_{right} + Ky_{top} + Ky_{bottom}}{3} = \frac{|2-3| + |2-1| + |2-3|}{3} = \frac{3}{3} = 1$$

$$K_{11} = \frac{Kx_{left} + Kx_{right} + Ky_{top} + Ky_{bottom}}{4} = \frac{|2-1| + |2-3| + |2-1| + |2-3|}{4} = \frac{4}{4} = 1$$

$$K_{12} = \frac{Kx_{left} + Ky_{top} + Ky_{bottom}}{3} = \frac{|2-1| + |2-1| + |2-3|}{3} = \frac{3}{3} = 1$$

$$K_{20} = \frac{Kx_{right} + Ky_{top}}{2} = \frac{|2-3| + |2-1|}{2} = \frac{2}{2} = 1$$

$$K_{21} = \frac{Kx_{left} + Kx_{right} + Ky_{top}}{3} = \frac{|2-1| + |2-3| + |2-1|}{3} = \frac{3}{3} = 1$$

$$K_{22} = \frac{Kx_{left} + Ky_{top}}{2} = \frac{|2-1| + |2-1|}{2} = \frac{2}{2} = 1$$

	0	1	2
0	1	1	1
1	1	1	1
2	1	1	1

Figure 35. Disparity Map after the estimation of the disparity for each micro image and the calculations performed to obtain those values.

After performing the disparity estimation for all the existing micro images by placing the red block in the point of view *row = 2*, *col = 2* and comparing it with all the neighbors, a disparity value would show for each micro image, that would look like the disparity map in Figure 35.

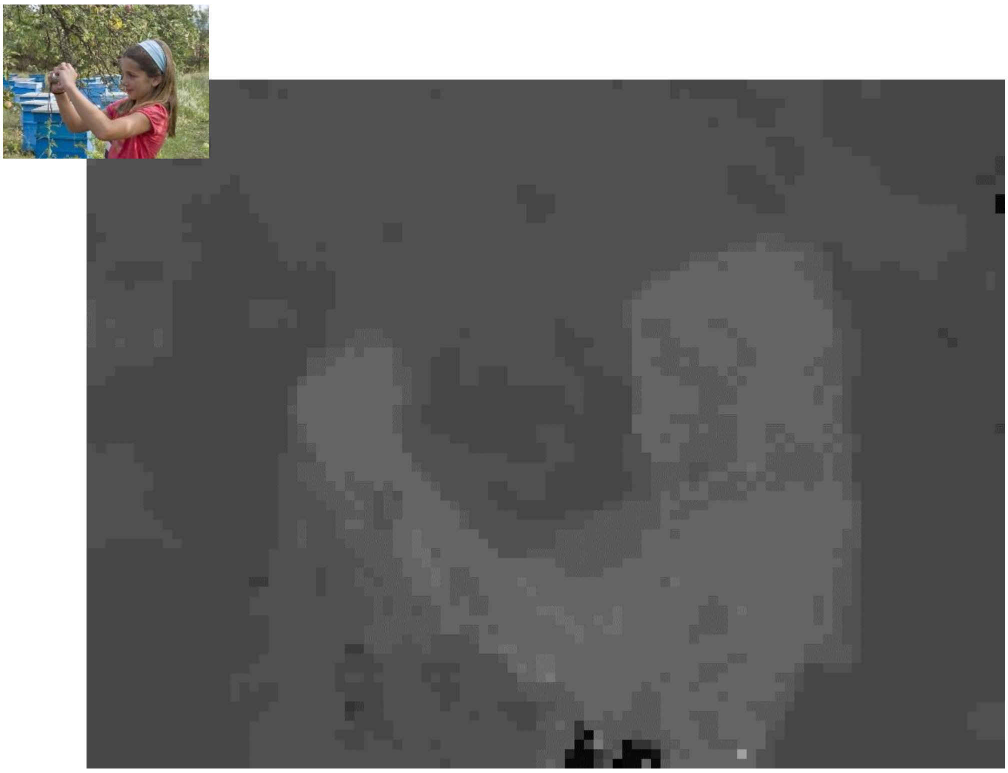


Figure 36. Disparity Map of the image Laura, row = 24, col = 24 and block size = 27.

Figure 36 illustrates the estimated disparity map for the image that was used to display the results from the previous introduced algorithms.

The original values of the disparity map were multiplied by a factor of ten to obtain bigger visual differences between the disparity values. The darker pixels correspond to small disparity values, that means they don’t change much their position from micro image to micro image and they are in the background of the image. The lighter pixels correspond to bigger disparity values that often change their position from micro image to micro image and they are in the foreground.

3.3.2. Disparity Estimation – Minimizing Errors with α

Small errors in the estimation of disparity create artifacts that can have a big visual impact for the user, this happens because when the estimated PS is wrong, the extracted blocks will not match after being upscaled and placed side by side.

So, a new adaptive estimation model that allows the user to have some control of the estimation process is presented by introducing a new parameter called alpha (α). The difference from the disparity estimation model presented in Section 3.2.1. to this one is to have a dependent factor of the final estimated PS values of the left and top adjacent micro images. To introduce this new factor, equation 12 must be replaced for a more robust equation, as presented:

$$comparison = \frac{\frac{\sum_{r=o}^{rows}\sum_{c=0}^{cols}(A_{rc}-B_{rc})^2}{r\times c}}{255} + \frac{|currentDisp. - avgNeighborDisp.|}{maxPossibleDisp.} \times \alpha \qquad (16)$$

the *currentDisp.* corresponds to the disparity value of the center of the block that is being compared with the red block and is calculated using:

$$currentDisp. = \begin{cases} |col - col_{current\_comparition}| & , horizontal\ neightbor \\ |row - row_{current\_comparition}| & , vertical\ neighbor \end{cases} \qquad (17)$$

The *avgNeightborDisp.* corresponds to the average of two neighbors for which their final disparity value has already been estimated. In this case, the neighbors from the micro images on the top and on the left are used. The value of the average neighbor disparity is given by equation:

$$avgNeighborDisp. = \frac{K_{left\_neightbour} + K_{top\_neightbour}}{2} \qquad (18)$$

The *maxPossibleDisp.* Corresponds to the maximum disparity value that a certain micro image can have and its given by:

$$maxPossibleDisp. = \min(row, col, MI.height - 1 - row, MI.width - 1 - col) \qquad (19)$$

As can be seen by the equation 16, the equation has contributions of the final estimated PS values calculated for the micro images on the left and on the top, minimizing possible errors in the estimation of the PS value for the current micro image.

Next, from Figures 38-43 the disparity maps for the following sequence of images (Figure 37) are presented, varying the parameter α.

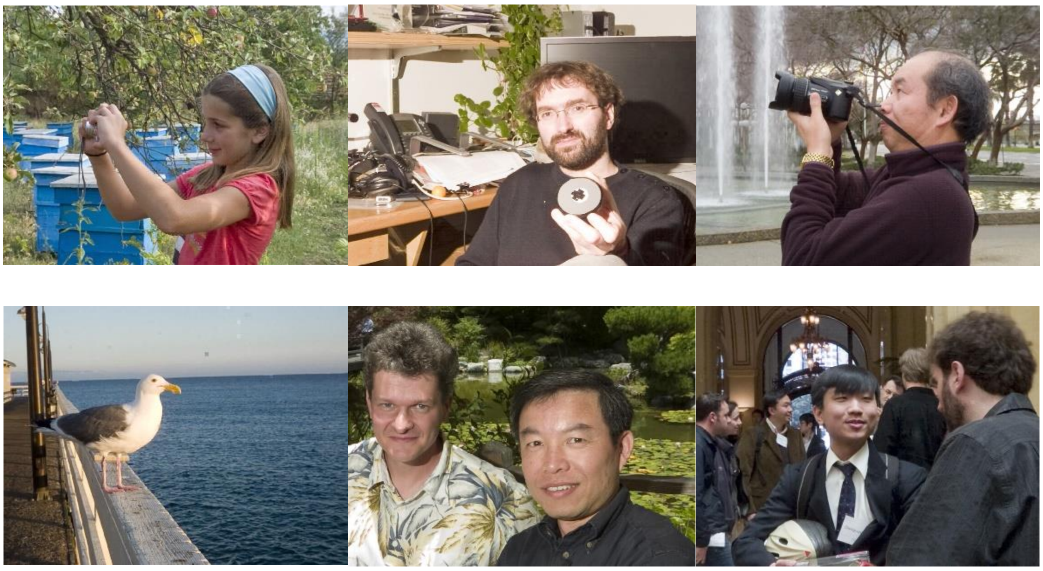


Figure 37. Image Sequence used to estimate the disparity maps with different α values.

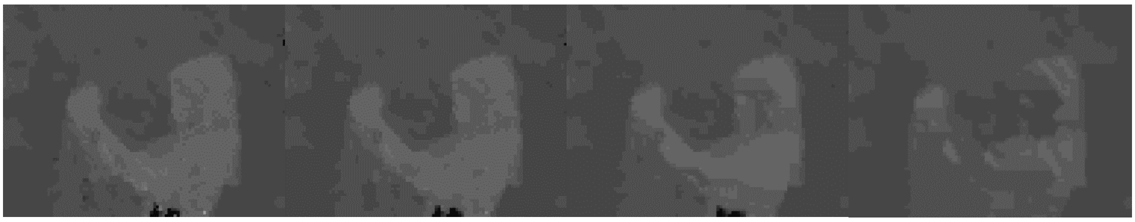


Figure 38. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Laura.

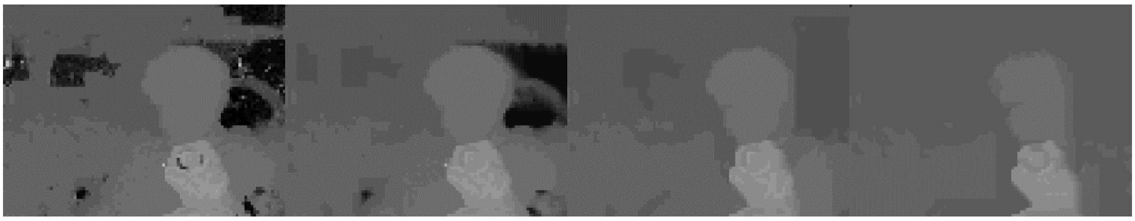


Figure 39. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Fredo.

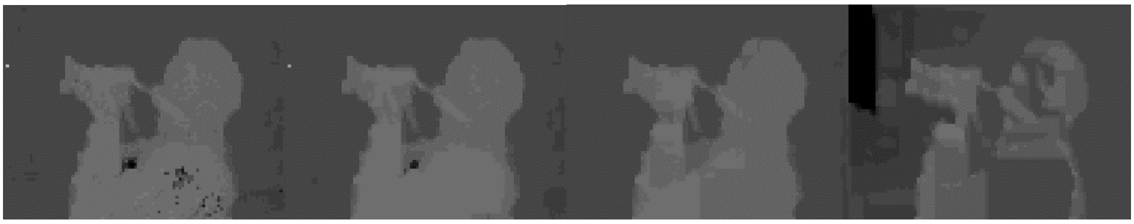


Figure 40. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Jeff.

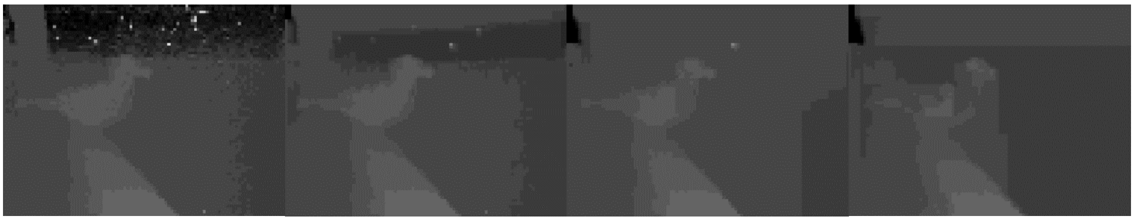


Figure 41. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Seagull.

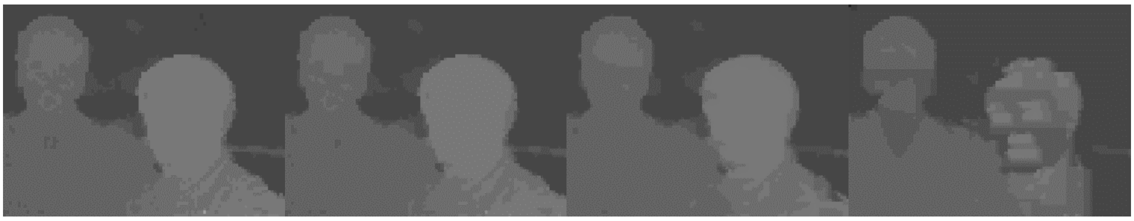


Figure 42. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Sergio.

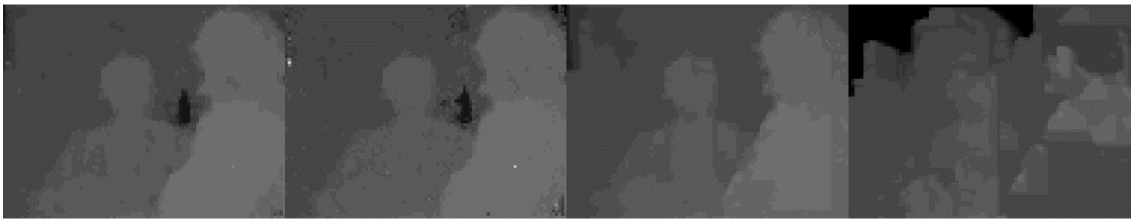


Figure 43. Disparity Maps with α = {0.0, 0.015625, 0.0625, 0.25} for the image Zhengyun1.

After analyzing all the rendered images for each image of the sequence with the following *a* values {0.0, 0.0078125, 0.015625, 0.03125, 0.0625, 0.125, 0.25, 0.50} was concluded that the rendered images with values around α = 0.015625 and α = 0.0625 have less artifacts than the ones with α = 0 (α = 0 corresponds to equation 12 which means the estimation model used is the one introduced in Section 3.2.1.), for more information consult Appendix B.

3.3.3. Disparity Map Based 2D Image Rendering (DM)

This 2D rendering algorithm was developed by Todor Georgiev and Andrew Lumsdaine [20] and can be seen as an optimization of the SSP algorithm introduced in Section 3.1.2., but instead of having a fixed patch size, it has multiple patch sizes to try to focus all the different planes of the image. The estimation of the disparity will be needed, to determine the patch size that better matches its neighbor patches. As the goal is to have an all-in-focus image, each different patch size value will allow to focus a certain focal plane, where identical patch sizes correspond to objects at a certain depth in the scene. Since the patches extracted from the different micro images have different sizes, un upscaling process will be needed to guaranty that all the extracted blocks have the same size before placing them side by side in the resulting image.

This way, an estimation of the disparity map (introduced in Section 3.4.1 or 3.4.2) is done for the light field image. Since a *row* and *col* were selected to define the position of the block used to calculate the disparity map, each micro image should get its estimated PS value and recalculate the right row and col to guaranty that all the patch sizes are centered in the same pixel, which means that a calculation to obtain the *newRow* should be performed using equation:

$$newRow = row + \frac{block\_size}{2} - \frac{patch_{size}}{2} \tag{20}$$

and the same should be done to obtain newCol using equation:

$$newCol = col + \frac{block\_size}{2} - \frac{patch_{size}}{2} \tag{21}$$

After getting the different patch sizes extracted from the different micro images, all the extracted pixel blocks must have the same size, so an upscaling to the maximum disparity value in the disparity map is necessary.

When all the extracted images have the same size, the arrangement is done side by side, as shown in the SSP algorithm in Section 3.3.2. and then the final rendered image is obtained.

Figure 44 and 45 show two results of the same light field image, the first one uses the first disparity estimation model introduced in Section 3.2.1. (α = 0) and the second image uses the second disparity estimation model with an α = 0.0625. If the artifact area inside the orange circle is analyzed, the second image has better results than the first one.

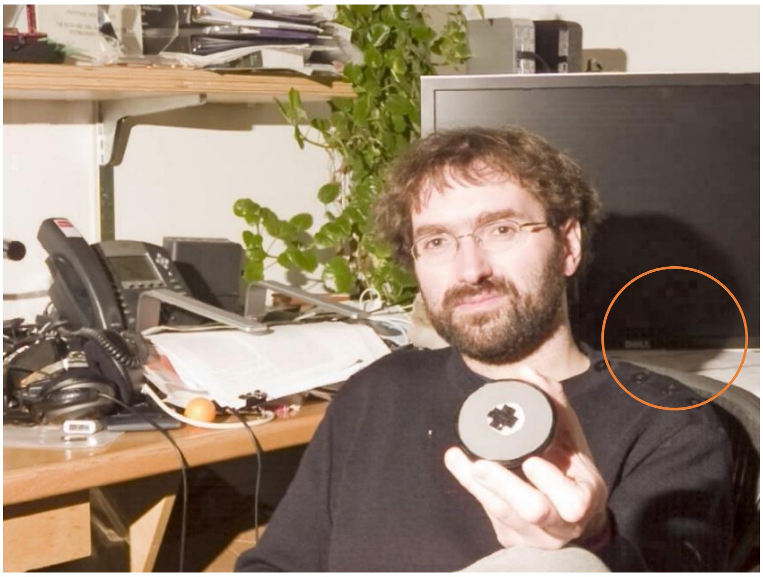


Figure 44. Fredo image using the DM algorithm with α = 0.0.

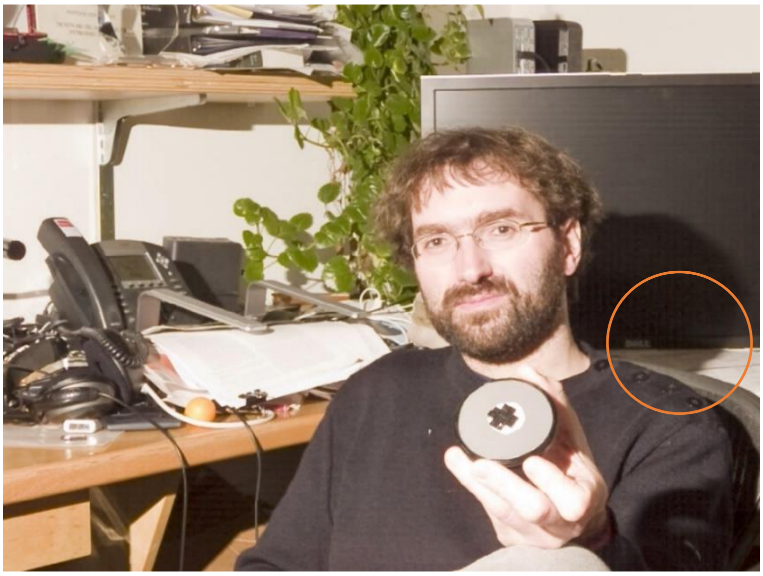


Figure 45. Fredo image using the DM algorithm with α = 0.0625.

Advantages: This is the first rendering algorithm that can display an all-in-focus 2D image.

Disadvantages: Since an upscaling process must be done to make sure all the extracted blocks have the same size, errors can be introduced in this process, depending on how much a certain block must be upscaled. Other visible disadvantage is the fact that this algorithm does not have any blending process associated so the outline of the different blocks can be seen in some areas of the image.

3.3.4. Disparity Blending Based 2D Image Rendering (DB)

The idea behind this algorithm is similar as the one explained by João Lino in [17] and is like the one presented in Section 3.2.3. except it has the blending process that has been introduced in the SSPB algorithm in Section 3.1.3. Since the disparity estimation models introduced in Section 3.2.1 and 3.2.2. are used again, different micro images might be associated with a different patch sizes, so, specifying a single window size won’t work. For that reason, a new input parameter must be introduced. This way, Window Size Percentage (WSP) corresponds to a percentage that will be multiplied for each patch size to find out what is the right window size for each micro image.

Once the window size is known for each micro image, a block of pixels with that size must be extracted from each micro image and again different blocks are presented with different sizes, so the upscaling process is necessary to make sure that all the blocks have the size of the biggest patch size, found in the disparity map, multiplied by the Window Size Percentage parameter. Just like in the SSPB algorithm all the patch sizes and window sizes in this algorithm should be odd numbers.

After executing the upscale process, the application of the blending process is necessary at the exact same way it was applied in the SSPB algorithm.

Figure 46 shows the Fredo image where a considerably better result of the artifact area inside the orange circle is seen.

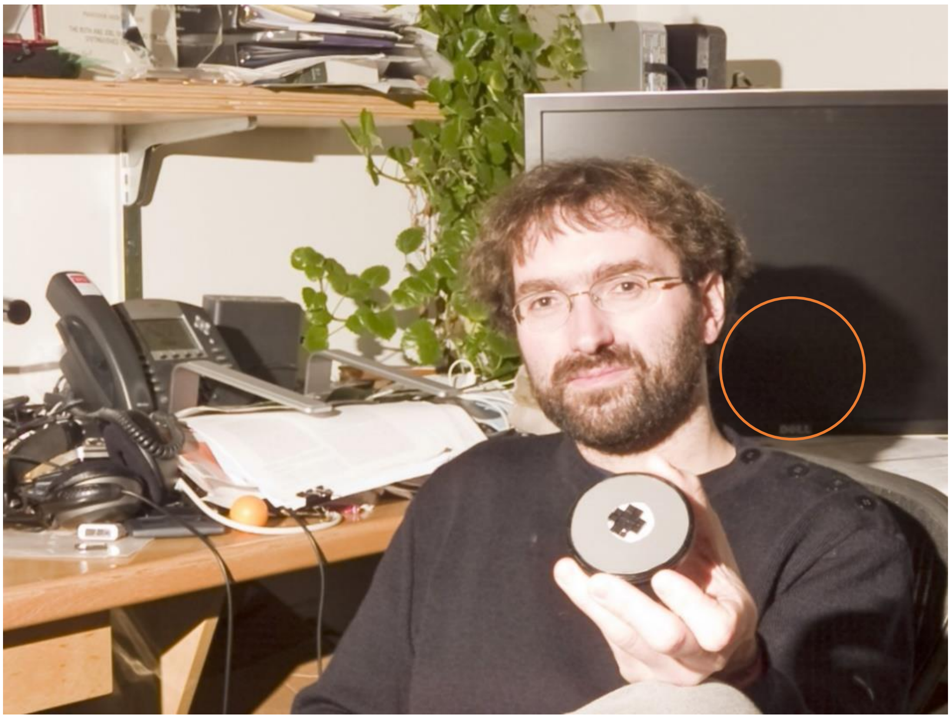


Figure 46. Fredo image using the DB algorithm with α = 0.0625 and WSP = 2.0.

Chapter 4 – Developed Rendering Application

4.1. Technologies

On the planning stage of the development process of this desktop application, a decision was made to select the programming language and libraries that were going to be used. The main goal was to create open source and independent tools that would easily work in different platforms with an easy to use graphical user interface.

Java was the selected programming language due to some of its characteristics such as:

- Automatic memory allocation;
- Garbage collection;
- Object oriented, allowing the creation of modular programs and reusable code;
- Platform-independent making it easy to run the application in different systems;
- Multithreading, having the capability for a program to perform several tasks simultaneously.

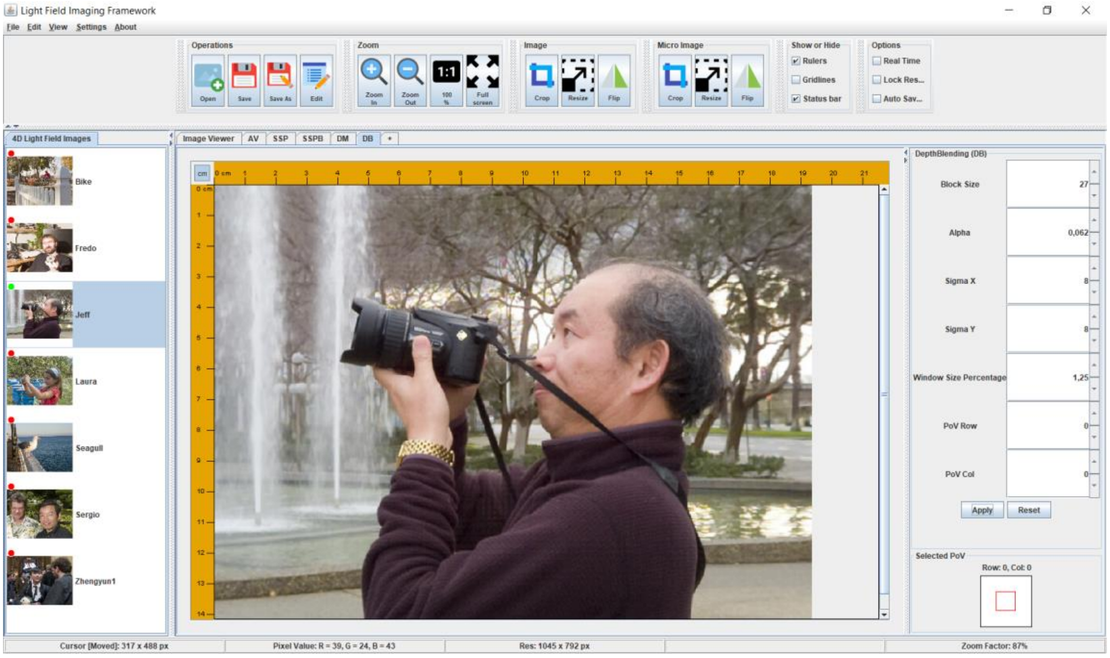
Swing is part of the Oracle's Java Foundation Classes (JFC) which is an API for providing a graphical user interface for Java programs. Since it is part of the Java Standard Edition Development Kit [23], the tools needed were already being used to be able to create a custom and easy to use graphical user interface, who would work regardless of whether the underlying user interface system is Windows, macOS or Linux.

OpenCV (Open Source Computer Vision Library) [24] is completely free for academic use, it has C++, Python and Java interfaces and supports Windows, Linux, macOS, iOS and Android. This library was designed for computational efficiency and a strong focus on real-time applications. The big community behind it and the quality of their documentation was also a big part of why this library was chosen.

Eclipse [25] was the chosen IDE due to its big affiliation to the open source and Java communities. Setting up the development environment on eclipse IDE with Java SE Development Kit1.7 and OpenCV 3.4.0 was simpler and faster when compared to some other environments such as C++ and OpenCV on Visual Studio IDE.

4.2. Graphical User Interface

The main software application is completely free and easy to use, where the user can try all the implemented algorithms, manipulate the algorithm parameters (see Figure 47), check the result in the full screen mode and save the perfect 2D image, by selecting the best focal plane and the best point of view of the light field scene.



Show or Hide
☑ Rulers
☐ Gridlines
☑ Status bar

Options
☐ Real Time
☐ Lock Res...
☐ Auto Sav...

DepthBlending (DB)

Block Size	27
Alpha	0,062
Sigma X	8
Sigma Y	8
Window Size Percentage	1,25
PoV Row	0
PoV Col	0

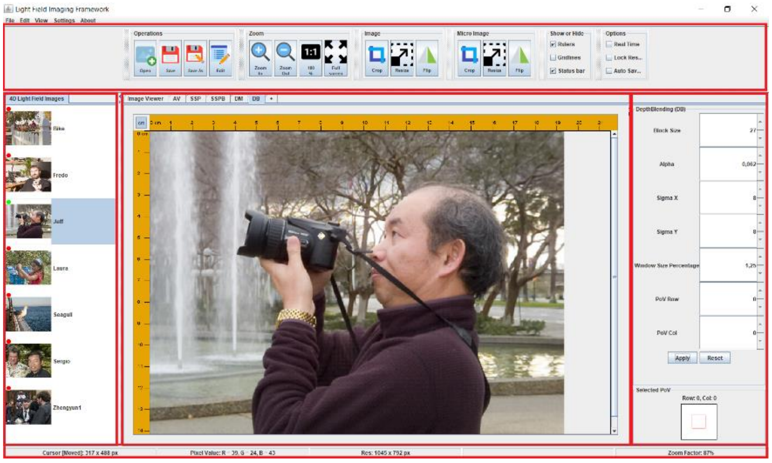
Selected PoV
Row: 0, Col: 0

Cursor [Moved]: 317 x 488 px | Pixel Value: R = 39, G = 24, B = 43 | Res: 1045 x 792 px | Zoom Factor: 87%

Figure 47. Print screen of the main desktop application.

This software might be useful for the image processing community as a content generator for objective and subjective tests. It is open source to allow contributions from other authors to make the application as robust as possible.

4.2.1. Overview



☑ Rulers
☐ Gridlines
☑ Status bar

☐ Real Time
☐ Lock Res...
☐ Auto Sav...

Block Size 27
Alpha 0,062
Sigma X 8
Sigma Y 8
Window Size Percentage 1,25
PoV Row 0
PoV Col 0

Row: 0, Col: 0

Cursor [Moved]: 317 x 488 px
Pixel Value: R - 39, G - 24, B - 43
Res: 1045 x 792 px
Zoom Factor: 87%

Figure 48. Main desktop application divided in the five most important parts.

The west part of the graphical user interface allows the user to select the light field image that he/she wants to serve as an input of all the implemented rendering algorithms. The image is loaded to memory when the user selects it for the first time, the decision to not load all the images when the program starts due to their high pixel resolution.

The central part of the graphical user interface allows the user to analyze and manipulate the chosen light field image in the tab Image Viewer, every other tab corresponds to a rendered algorithm output image tab. By switching tabs, the user can compare the results of different algorithms. All the tabs allow to zoom in or zoom out by scrolling the wheel on the mouse, to scroll up or scroll down the image by using the horizontal and vertical scrollbar. In the all-in-focus rendering algorithm tabs, the user can see its depth map estimation representation by clicking on the 2D rendered algorithm result image.

The east part of the graphical user interface only exists in the rendering algorithm tabs and corresponds to the rendering algorithm input parameters panel, where the user can specify all the desired options before starting the rendering process.

The north part of the graphical user interface contains a toolbar with tools that can be applied to the image or tools that will change the user experience using the application.

The south part of the graphical user interface contains some information such as the resolution of the displaying image, its zoom factor, the position of the cursor and the RGB values of the pixel where the cursor is at.

4.2.2. Rendering Algorithms Input Parameters

- PoV Row – This parameter must be an integer value and specifies the selected point of view row.

$$PoV_{Row} = \begin{cases} \frac{MI.height}{2} \leq PoV_{ROW} \leq \frac{MI.height}{2} & ,MI.height \,\%\, 2 == 1 \\ \frac{MI.height}{2} \leq PoV_{ROW} < \frac{MI.height}{2} & ,MI.height \,\%\, 2 == 0 \end{cases} \quad (22)$$

- PoV Col – This parameter must be an integer value and specifies the selected point of view col.

$$PoV_{COL} = \begin{cases} \frac{MI.width}{2} \leq PoV_{COL} \leq \frac{MI.width}{2} & ,MI.width \,\%\, 2 == 1 \\ \frac{MI.width}{2} \leq PoV_{COL} < \frac{MI.width}{2} & ,MI.width \,\%\, 2 == 0 \end{cases} \quad (23)$$

- Patch Size – This parameter must be an odd integer value and specifies the size of the block of pixels that will represent each micro image in the rendered 2D image. Each patch size will focus a certain image plane. (In the SSP rendering algorithm the patch size value can be also an even number, due to the simplicity of the algorithm).

$$0 \leq PS < \min(MI.width, MI.height) \quad (24)$$

- Window Size – This parameter must be an odd integer value and specifies the size of the block of pixels that will represent each micro image in the blending process of the rendering algorithm.

$$PS \leq WS < \min(MI.width, MI.height) \quad (25)$$

- Sigma X – This parameter must be a positive double value and specifies how fast the horizontal gaussian curve goes from its maximum value to its minimum value, its normally used in all the algorithms that have the blending process. The bigger the sigma, the slower curve reaches its minimum value.

$$Sigma_X > 0.0 \quad (26)$$

- Sigma Y – This parameter must be a positive double value and specifies how fast the vertical gaussian curve goes from its maximum value to its minimum value, its normally used in all the algorithms that have the blending process. The bigger the sigma, the slower curve reaches its minimum value.

$$Sigma_Y > 0.0 \tag{27}$$

- Block Size – This parameter must be a positive integer value and specifies the size of the block of pixels that will be compared with all the blocks from the adjacent micro images, to calculate the displacement from that block in the neighbor micro images.

$$0 \leq BS < \min(MI.width, MI.height) \tag{28}$$

- Alpha – This parameter must be zero or a positive double value and specifies the minimization factor of the estimation patch size errors from the disparity estimation process used in every all-in-focus algorithm.

$$\alpha \geq 0.0 \tag{29}$$

- Window Size Percentage – This parameter must be a double value equal or greater than one. This parameter only exists in the DB all-in-focus algorithm and specifies how much the patches from the depth map estimation must be multiplied by, to calculate the respective window sizes that will be used for the blending process.

$$Window\ Size\ Percentage \geq 1.0 \tag{30}$$

4.3. Block Diagram

The block diagram illustrated in Figure 49 allows to understand the different steps that must be executed for the user in order to test a specific rendering algorithm.

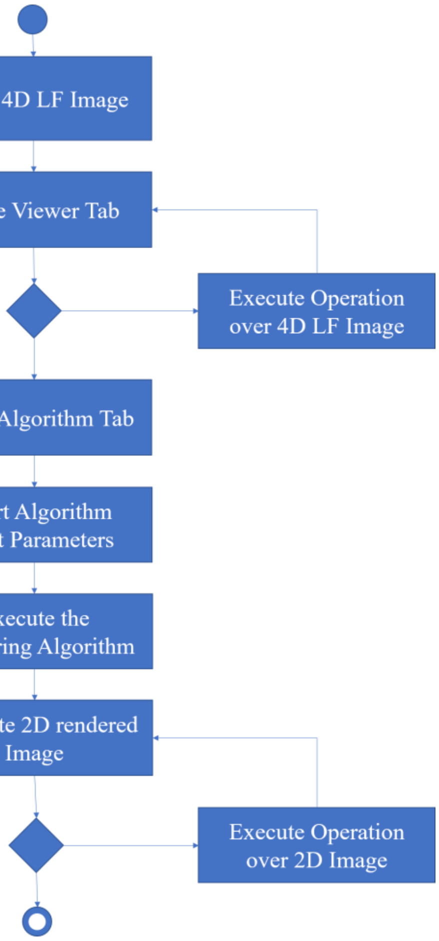

Select 4D LF Image
Image Viewer Tab
Execute Operation over 4D LF Image
Select Algorithm Tab
Insert Algorithm Input Parameters
Execute the Rendering Algorithm
Generate 2D rendered Image
Execute Operation over 2D Image

Figure 49. Block diagram of the main steps to execute a certain rendering algorithm.

The rendering process starts with the selection of the 4D light field image, the image is displayed in the Image Viewer Tab, where the user can view and execute the toolbar operations to make some adjustments into the input image.

When the input image is ready to be processed, the user must select one of the five implemented rendering algorithms, by selecting the right tab.

The user must insert the values of all the algorithm parameters press the apply button to start executing the rendering algorithm.

After a few milliseconds, depending on the chosen algorithm, the 2D output rendered image is displayed on the selected algorithm tab and all the details of the resulting image are updated in the status bar. At this point the user can save the rendered image, perform other operation or just finish the process.

Chapter 5 – Proposed Quality Assessment Tests

This chapter provides all the details about the quality assessment tests performed in the scope of this dissertation: the questions are tried to be answered, the used methodology, characterization of the participants, the application that was used to display and collect the votes, and a description of each test. The quality assessment tests performed followed recommendation from ITU-T P.910 [26]

5.1. Questions

The quality assessment tests were performed with the intent of helping us to figure it out what are the answers for the research questions previously introduced. The tests were divided into six different test cases, some of them are algorithm direct comparisons, others tend to understand the impact of the most important parameters like the PS and WS.

1. A comparison of the quality of the rendered images provided by the SSP algorithm against those resulting from the SSPB algorithm. It tries to answer the question *which one of the textured based rendering solutions provides the best rendered image, SSP or SSPB*?
2. An understanding of the impact of the patch size parameter in the SSPB algorithm. It tries to answer the question *which image would the voters choose after seeing the same content focused on different planes*?
3. An understanding of the impact of the window size parameter in the SSPB algorithm. It tries to answer the question *in a textured based rendering solution, would the test participants prefer images with more blur or less blur*?
4. A quality comparison for the rendered images provided by the DM algorithm with the images provided by the DB algorithm, for different values of the α parameter. It tries to answer the question *which one of the disparity-based rendering solutions provides the best rendered image, DM or DB*?
5. An understanding of the impact of the window size percentage parameter in the DB algorithm. It tries to answer the question *in a disparity-based rendering solution, would the test participants prefer images with more blur or less blur*?
6. A comparison was also required to answer the question *do the test participants prefer all-in-focus images or the same image with only one plane in focus*? For this test, rendered images provided by the SSPB algorithm were compared with images provided by the DB algorithm.

5.2. Methodology

Choosing the right methodology for the quality assessment tests is a key decision to obtain the best possible outcome. Common options like the single stimulus and double stimulus methods were considered as both allow to compare both images in an indirect way, but some of the questions that this dissertation is trying to answer must be compared in a direct way, which means that a pair comparison [26] [27] that would allow to display both images and rate the relation between those two would be the best option in order to obtain a better comparison result.

In order to save some time and to easily notice some of small differences between the two displayed images, the option to present both images simultaneously was selected. This way the method paired comparison – simultaneous presentation (PC-SP) [28] was chosen, allowing to display both images at the same time to obtain a higher certainty degree when comparing each pair of images.

This method implies that the test images are presented in pairs, simultaneously. Each pair consists of the same image and all pairs of images should be displayed in all the possible combinations. The human subjects evaluate which element in the pair is preferred in the context of the test scenario.

5.3. Grading Scale

The comparison is used to evaluate the relation between two images, in this case simultaneously. The participant compares the presented pair of images and assess the differences between them. A seven-level comparison grading scale was used for the quality assessment tests [28].

Value	Score
- 3	Much Worse
- 2	Worse
- 1	Slightly Worse
0	The Same
+ 1	Slightly Better
+ 2	Better
+ 3	Much Better

Table 1: Seven level comparison grading scale [28].

5.4. Test Participants

The human subjects who participated in the tests were male and female students from ISCTE-IUL, with ages between eighteen and twenty-five years old. All of them were non-experts. A total of 20 students participated in the tests, a higher number than the one recommended in [26][28].

Before the assessment, each participant was tested in terms of visual acuity and color blindness. One of them failed the visual acuity test with one of the eyes, but after an analysis of his votes, it was decided to keep his scores, since they were similar to the other participants' scores.

In order to test the visual acuity of each participant, a Snellen chart was printed in a A4 sheet. The participants were placed at a three meter of distance from the chart. And asked to enumerate a few of the letters in the normal acuity range.

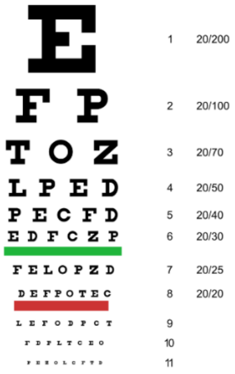


Figure 50. Snellen chart used before the assessment to test the visual acuity [28].

To test the color blindness of each participant the Ishihara plates were used. As described by Dr. Shinobu Ishihara in 1972 in the "Tests for Color-Blindness" book, a simplest version of the test can be performed by showing only six Ishihara plates from the first 15 plates were shown to the participants at seventy-five centimeters from the screen.

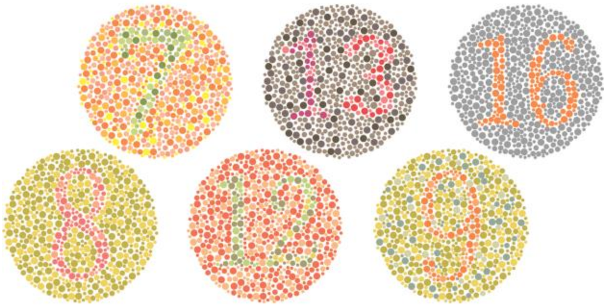


Figure 51. Ishihara plates used before the assessment to test the color blindness [28].

5.5. Developed Test Application

The software application used for the quality assessment tests was also developed using Java, OpenCV and Swing. It consisted in a simple GUI (Figure 52) that displayed the pair of images to be compared, with the comparison grading scale on the bottom and a voting button. When the user presses the vote button, the value of the grading scale is obtained and saved into a file. The screen changes to a totally gray screen for five hundred milliseconds and after that, the next pair of images to be compared is displayed and the grading scale selector is reset to the neutral position (0).

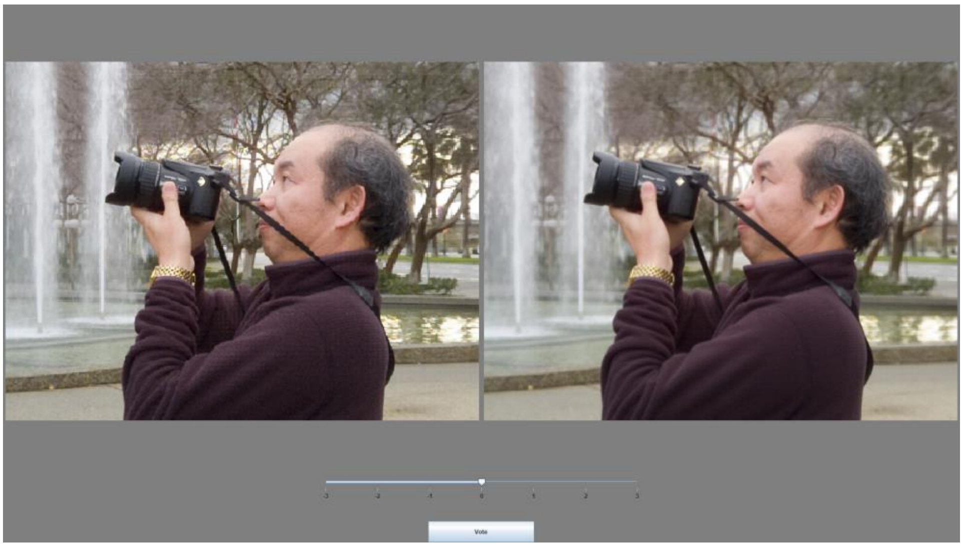


Figure 52. Developed application used for the quality assessment tests.

5.6. Test Material

Figure 53 displays all the seven images used for the quality assessment tests. All the images were provided from [29]. The images names from top to the bottom are Laura, Fredo, Jeff, Seagull, Sergio, Zhengyun1 and Bike.

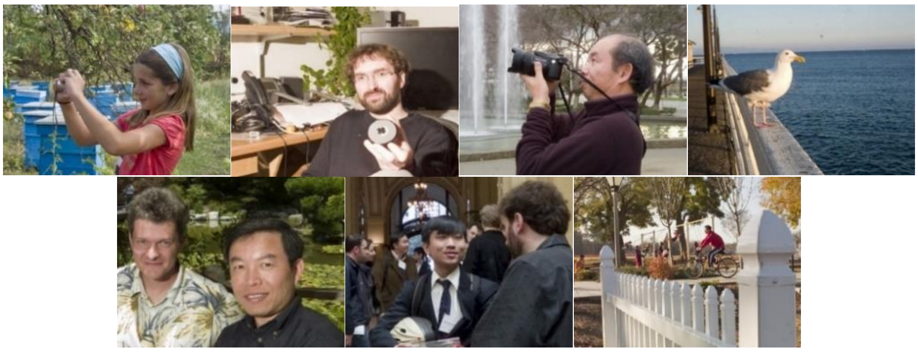


Figure 53. Test material used for the assessment tests.

All the 4D light field images used have a resolution of $w=7104$, $h=5328$ pixels, where each micro image has a resolution *of* $w=74$, $h=74$ which means that each light field image has a matrix of *72* per *96* micro images. All the rendered images displayed in the tests were centered in half of the screen and have been resized to have a height of 720 pixels and keeping the original aspect ratio.

5.7. Tests Cases

The following sections will introduce the six different tests used in the performed quality assessment.

All the images used for the tests were centered in the middle[1] PoV of each micro image.

Using the same static value for the two components of the sigma parameter, when the SSPB and DB algorithms were used, was also a decision made, the chosen value was eight so that in the blending process, all the pixels would have a significant weight contribution, Appendix A allows to have a better understanding of the relation between the sigma parameters and the window size and justifies the 1.5 factor presented in equation 31, factor that better suits for all the PS used with a sigma of eight.

To test these algorithms in different conditions, a previous study was made to find out which patch size values would affect the perception of the focal planes for each image, concluding that for some images only 2 patch sizes should be used (background focus or foreground focus) and for other images 3 patch sizes should be used (background focus or foreground focus or middle plane focus).

For all the tests that don’t intend to measure the impact of the window size parameter, was decided that the value of the WS had to be relatively proportional to the PS and for that, Equation 31 was applied, and the result was rounded to the closest equal or lower odd number.

$$WS = PS \times 1.5 \tag{31}$$

As presented in Section 5.2. the PC-SC was the selected methodology, all the tests presented a pair of images at the same time and since each pair can be displayed in two different ways (XY and YX) we used both, doubling the number of pairs displayed for each one of the six tests. This decision gave us a good understanding of how reliable and consistent each participant was.

[1] PoV_row = PoV_col = 0 when using the graphical user interface of the developed software rendering application.

5.7.1. Algorithm Comparison: SSP vs SSPB

The goal of this test was to compare two of the implemented textured based image rendering solutions, the SSP and the SSPB algorithm. The big difference between these two algorithms is the blending process in the SSPB, that introduces some image blur, helping to hide some of the existing artifacts.

For each one of the two algorithms eighteen images were generated with the parameters provided in Table 3, creating eighteen different pairs. Since each pair was showed in two different ways, thirty-six pairs were used for this test.

Due to the high number of pairs, this test was divided into two groups – A and B – with eighteen pairs each.

		SSP	SSPB	
Num.	Image	PS	PS	WS
1	Bike	7	7	9
2		9	9	13
3		11	11	15
4	Fredo	9	9	13
5		11	11	15
6		13	13	19
7	Jeff	7	7	9
8		9	9	13
9		11	11	15
10	Laura	7	7	9
11		9	9	13
12	Seagull	7	7	9
13		9	9	13
14	Sergio	7	7	9
15		9	9	13
16		11	11	15
17	Zhengyun1	7	7	9
18		9	9	13

Table 2: Parameters of the generated images used for the SSP vs SSPB test.

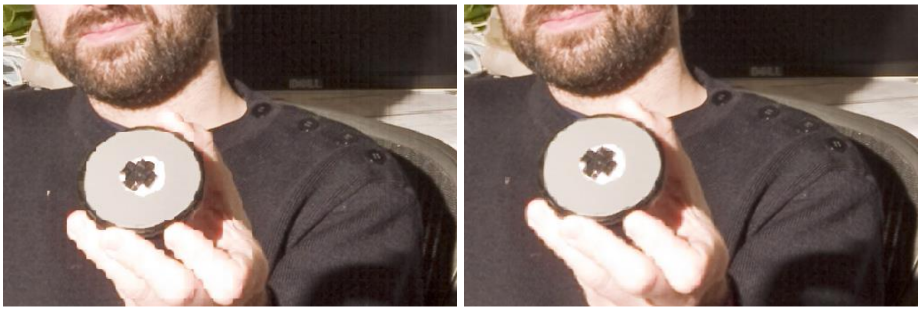


Figure 54. Fredo image rendered by SSP and SSPB, PS=11, WS=15.

5.7.2. SSPB: The Influence of the Patch Size Parameter

The objective of this test was to understand the influence of the patch size parameter in the textured based rendering solution SSPB, since different patch sizes correspond to different image planes on focus. As explained in the introduction of Chapter 5, the patch sizes used were a result of a previous study where the impact of each patch size in every image was analyzed, this way some images will be tested with just two patch sizes while others will be tested for three different patch sizes.

The results of this test show which focal plane was preferred, e.g., a focused background, a focused foreground or focused in the middle of the two. The analysis of this test must be different from the previous one, since it is not so straightforward, since in this case the image content has a strong impact on the results.

Eighteen different images were generated for this test using the parameters presented in Table 4. Since all combinations of the generated images from the same light field image were compared, fifteen different pairs were created. Since each pair is displayed in two ways, the value was doubled to thirty pairs.

This test was divided into two groups – A and B – with fifteen pairs each.

		SSPB		
Num.	Image	PS	WS	Comb.
1	Bike	7	9	3
2		9	13	
3		11	15	
4	Fredo	9	13	3
5		11	15	
6		13	19	
7	Jeff	7	9	3
8		9	13	
9		11	15	
10	Laura	7	9	1
11		9	13	
12	Seagull	7	9	1
13		9	13	
14	Sergio	7	9	3
15		9	13	
16		11	15	
17	Zhengyun1	7	9	1
18		9	13	

Table 3: Parameters of the generated images used for the SSPB: influence of the PS test.

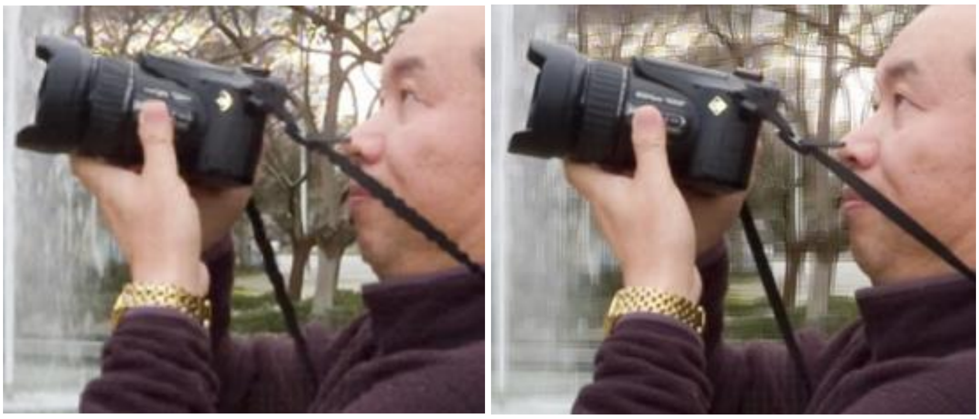


Figure 55. Jeff image rendered by SSPB, PS=7, WS=9 and PS=11, WS=15.

5.7.3. SSPB: The Influence of the Window Size Parameter

The idea of this test was to understand the influence of the window size parameter in the textured based rendering solution SSPB, since the window size corresponds to the intensity of the introduced blur in the whole image. The results of this test will provide which ratio Patch Size / WS leads to the most acceptable blur for each image. In this test, image content is also expected to have a strong impact on the results.

The selected PS was the one that would have a focused foreground for each different image, this means the background will have artifacts since the selected patch size does not match the patch size that would focus the background. The right WS is key to hide/soft the existing artifacts.

Twenty-four images were generated for this test using the parameters specified in Table 5. Since all combinations of the generated images from the same light field image were compared, thirty-six different pairs were created. That value was doubled to seventy-two pairs as each pair is displayed in two different ways.

This test was divided into four groups – A, B, C and D – with eighteen pairs each.

		SSPB		
Num.	Image	PS	WS	Comb.
1	Fredo	11	11	6
2		11	15	
3		11	19	
4		11	23	
5	Jeff	9	9	6
6		9	13	
7		9	17	
8		9	21	
9	Laura	9	8	6
10		9	13	
11		9	17	
12		9	21	

		SSPB		
Num.	Image	PS	WS	Comb.
13	Seagull	9	9	6
14		9	13	
15		9	17	
16		9	21	
17	Sergio	11	11	6
18		11	15	
19		11	19	
20		11	23	
21	Zhengyun1	9	9	6
22		9	13	
23		9	17	
24		9	21	

Table 4: Parameters of the generated images used for the SSPB: Influence of the WS test.

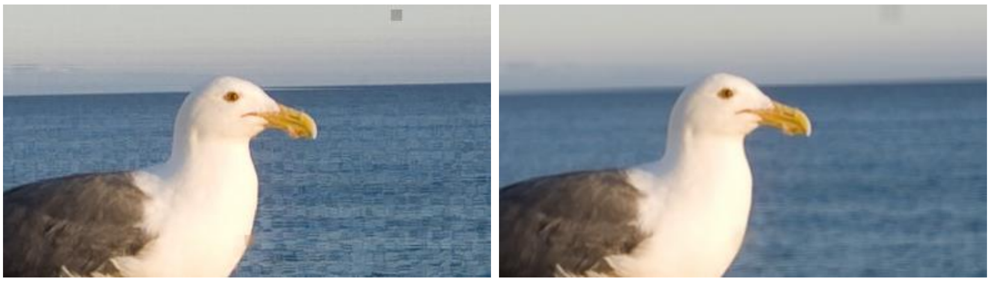


Figure 56. Seagull image rendered by SSPB, PS=9, WS=9 and PS=9, WS=21.

5.7.4. Algorithm Comparison: DM vs DB

The goal of this test was to compare two of the implemented disparity-based image rendering solutions, the DM and the DB algorithm. The big difference between these two algorithms is the blending process in the DB, that introduces some image blur, helping to soft the borders between adjacent micro images contributions.

A big difference with the SSP and SSPB algorithms, the ones used in the first test, is that multiple patch size values will be estimated and used to try to focus all the different focal planes from the scene, creating an all-in-focus image and consequently reducing drastically the existing artifacts.

For each one of the two algorithms twenty-four images were generated with the parameters presented in Table 6, creating twenty-four different pairs. Since each pair was showed in two different ways, forty-eight pairs were used for this test.

This test was divided into two groups – A and B – with twenty-four pairs each.

		DM	DB
Num.	Image	α	α
1	Fredo	0.0	0.0
2		0.15625	0.15625
3		0.0625	0.0625
4		0.25	0.25
5	Jeff	0.0	0.0
6		0.15625	0.15625
7		0.0625	0.0625
8		0.25	0.25
9	Laura	0.0	0.0
10		0.15625	0.15625
11		0.0625	0.0625
12		0.25	0.25

		DM	DB
Num.	Image	α	α
13	Seagull	0.0	0.0
14		0.15625	0.15625
15		0.0625	0.0625
16		0.25	0.25
17	Sergio	0.0	0.0
18		0.15625	0.15625
19		0.0625	0.0625
20		0.25	0.25
21	Zhengyun1	0.0	0.0
22		0.15625	0.15625
23		0.0625	0.0625
24		0.25	0.25

Table 5: Parameters of the generated images used for the DM vs DB test.

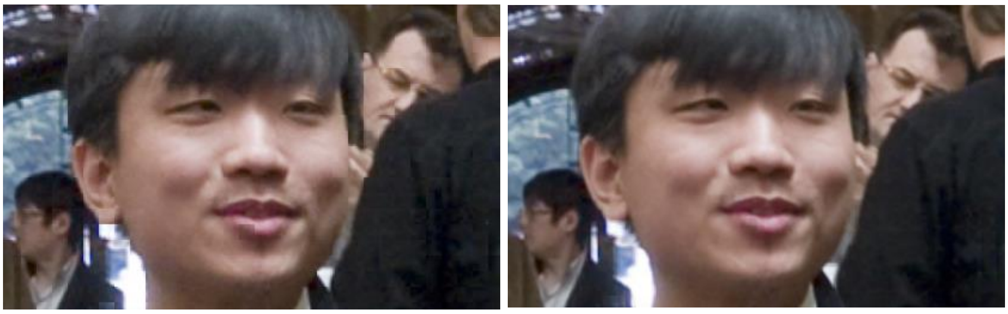


Figure 57. Zhengyun1 image rendered by DM and DB, α=0.015625, WSP=1.5.

5.7.5. DB: The Influence of the Window Size Percentage Parameter

The idea of this test was to understand the influence of the window size percentage parameter in the disparity-based rendering solution DB, since window size percentage correspond to the intensity of the introduced blur in the whole image. The results of this test will provide which ratio Patch Size / WS leads to the most acceptable blur for each image. In this test, image content is also expected to have a strong impact on the results.

The difference between this test and the third test, where the influence of the window size parameter for the SSPB was tested, is that there won't be control of the selected patch size as a disparity estimation was used, which means that multiple patch sizes were used and consequently multiple window sizes.

Twenty-four images were generated for this test using the parameters specified in Table 7. Since all combinations of the generated images from the same light field image were compared, thirty-six different pairs were created. That value was doubled to seventy-two pairs as each pair is displayed in two different ways.

This test was divided into four groups – A, B, C and D – with eighteen pairs each.

		DB	
Num.	Image	WSP	Comb.
1	Fredo	1.25	6
2		1.50	
3		1.75	
4		2.00	
5	Jeff	1.25	6
6		1.50	
7		1.75	
8		2.00	
9	Laura	1.25	6
10		1.50	
11		1.75	
12		2.00	

		DB	
Num.	Image	WSP	Comb.
13	Seagull	1.25	6
14		1.50	
15		1.75	
16		2.00	
17	Sergio	1.25	6
18		1.50	
19		1.75	
20		2.00	
21	Zhengyun1	1.25	6
22		1.50	
23		1.75	
24		2.00	

Table 6: Parameters of the generated images used for the DB: Influence of WSP test.

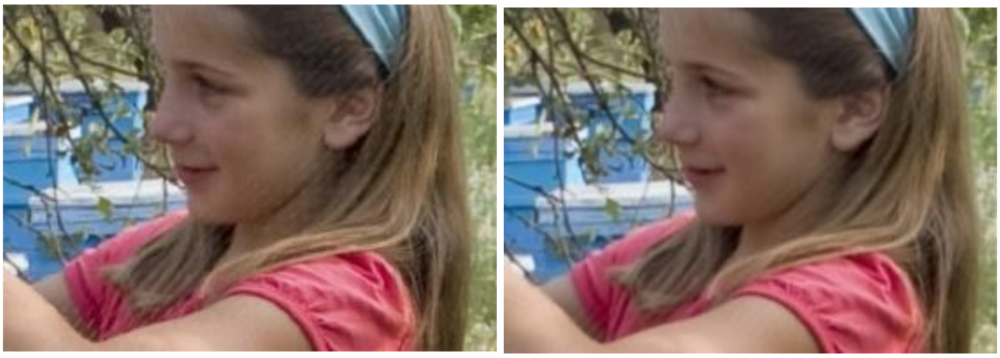


Figure 58. Laura image rendered by DB, α=0.0625, WSP=1.25 and α=0.0625, WSP=2.00.

5.7.6. Algorithm Comparison: SSPB vs DB

The objective of this test was to compare an implemented textured based image rendering solution SSPB with an implemented disparity-based image rendering solution DB, two totally different solutions. Both algorithms use the blending process, helping to soft/hide some of the existing artifacts, however the textured based solution has only one plane on focus, while the disparity-based solution leads to an all-in-focus image. For each one of the two algorithms six images were generated with the parameters provided in Table 8, creating six different pairs. Since each pair was showed in two different ways, twelve pairs were used for this test. This test wasn't divided in groups.

Table 7 displays the parameter values used for the SSPB, only images where the main subject is on focus have been used (different PS value based on the image content. In the DB algorithm the *a* used its optimal value of 0.0625 and both algorithms use a WSP factor similar to 2.0.

		SSPB		DB	
Num.	Image	PS	WS	α	WSP
1	Fredo	11	23	0.0625	2.00
2	Jeff	9	19	0.0625	2.00
3	Laura	9	19	0.0625	2.00
4	Seagull	9	19	0.0625	2.00
5	Sergio	11	23	0.0625	2.00
6	Zhengyun1	9	19	0.0625	2.00

Table 7: Parameters of the generated images used for the SSPB vs DB test.

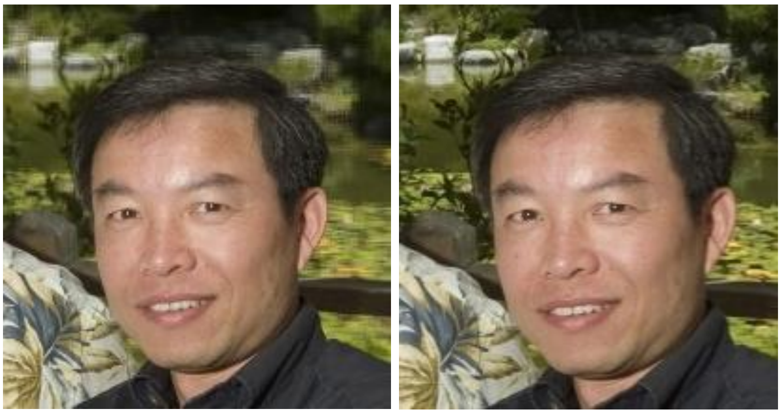


Figure 59. Sergio image rendered by SSPB (PS=11, WS=23) and DB (α=0.0625, WSP=2.00).

5.7.7. Test Versions

As explaining throughout the introduction of each one of the six tests, some of them were divided into groups. Joining all the groups from each test, four different versions of the test were created as illustrated in Table 9.

	Test Versions			
Test	Test Version 1	Test Version 2	Test Version 3	Test Version 4
SSP vs SSPB	A (01-18)	B (19-36)	A (01-18)	B (19-36)
SSPB: PS	A (16-30)	B (01-15)	A (16-30)	B (01-15)
SSPB: WS	A (01-18)	B (37-54)	C (19-36)	D (55-72)
DM vs DB	A (25-48)	B (01-24)	A (25-48)	B (01-24)
DB: WS	A (01-18)	B (37-54)	C (19-36)	D (55-72)
SSPB vs DB	A (01-12)	A (01-12)	A (01-12)	A (01-12)

Table 8: Table with all the images existing in each one of the four test versions.

Chapter 6 – Quality Assessment Results and Analysis

In this chapter the results from the quality assessment tests are displayed and analyzed, based on ideas from [30][31]. The results of the tests came in a matrix of integers, where each line *i* correspond to a test with a certain pair of images and each column *j* correspond to each participant vote for that specific test. All the odd rows contain a test where the images were showed in format XY and the next row has the exact same test but inverted as YX. In order to compare the results from both, the first thing to do is to invert all the votes from the even rows.

The valid values for *i* and *j* are:

$$\begin{gathered} 1 \leq i \leq M \\ 1 \leq j \leq N \end{gathered} \tag{32}$$

where M and N represent the number of individual pairs and the number of participants, respectively.

The score given by a participant for a specific pair of images is represented by x_{ij}, that can assume the following integer values:

$$-3 \leq x_{ij} \leq 3 \tag{33}$$

If x_{ij} corresponds to:

- a negative number means that the left image (-) of the pair is better;
- a positive number means that the right image (+) of the pair is better;
- a zero means that both images are equal.

Throughout the following sections where the results from each test will be presented, four types of analysis have been used:

- Individual Analysis

Individual analysis allows to obtain a group of detail about a certain pair, like the individual confidence value $IC(i)$, the individual score without outliers $IS(i)$ and the individual winner image of the pair $IW(i)$.

To obtain the score of the pair *i* without outliers, the average score value $\bar{u}(i)$ must be calculated:

$$\bar{u}(i) = \frac{1}{N}\sum_{j=1}^{N} x_{ij} \tag{34}$$

After calculating the average score value $\bar{u}(i)$, the standard deviation $\sigma(i)$ must be calculated:

$$\sigma(i) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left[x_{ij} - \bar{u}(i)\right]^2} \tag{35}$$

When the $\bar{u}(i)$ and the $\sigma(i)$ are known the pair individual score without outliers $IS(i)$ can be calculated using:

$$IS(i) = \frac{1}{N}\sum_{j=1}^{N} x_{ij}\,, \qquad \bar{u}(i) - 2 \times \sigma(i) \leq x_{ij} \leq \bar{u}(i) + 2 \times \sigma(i) \tag{36}$$

To obtain the confidence value, the individual negative confidence value $INC(i)$ must be determined:

$$INC(i) = \frac{100}{N}\sum_{j=1}^{N}\left|x_{ij}\right|, \quad x_{ij} < 0 \tag{37}$$

Also, the individual positive confidence value $IPC(i)$ must be determined:

$$IPC(i) = \frac{100}{N}\sum_{j=1}^{N} x_{ij}, \quad x_{ij} > 0 \tag{38}$$

When both $INC(i)$ and $IPC(i)$ are known, the individual confidence value of the pair $IC(i)$ is given by:

$$IC(i) = \begin{cases} INC(i), & INC(i) \geq IPC(i) \\ IPC(i), & INC(i) < IPC(i) \end{cases} \tag{39}$$

The individual winner of the pair is also calculated using the values of $INC(i)$ and $IPC(i)$ using equation:

$$IW(i) = \begin{cases} Image\ (-) & ,INC(i) > IPC(i) \\ Image\ (+) & ,INC(i) < IPC(i) \\ Draw & ,INC(i) = IPC(i) \end{cases} \tag{40}$$

- Group Analysis:

Group analysis allows to extract details about the two pairs of the same images, that were displayed as XY and YX, and obtain the average group confidence value $\overline{GC}(k)$, the average group score without outliers $\overline{GS}(k)$ and the group winner image $GW(k)$, where k corresponds to the group number and is given by:

$$1 \leq k \leq \frac{M}{2} \tag{41}$$

Since each group is formed by two pairs, the maximum value for *k* corresponds to half of the existing pairs *M*.

The average group confidence value $\overline{GC}(k)$ that allows to understand the average percentage of participants that voted for the same image in both pairs, otherwise this result will be inconclusive. This metric can be calculated using:

$$\overline{GC}(k) = \begin{cases} \dfrac{IC(2k) + IC(2k+1)}{2}, & IW(2k) = IW(2k+1) \\ Inconclusive, & IW(2k) \neq IW(2k+1) \end{cases} \quad (42)$$

The average group score value $\overline{GS}(k)$ that allows to understand the average score given by the participants to the winning image when the same image was preferred in both pairs, otherwise this result will be inconclusive. This metric is given by:

$$\overline{GS}(k) = \begin{cases} \dfrac{IS(2k) + IS(2k+1)}{2}, & IW(2k) = IW(2k+1) \\ Inconclusive, & IW(2k) \neq IW(2k+1) \end{cases} \quad (43)$$

The winner image from the group winner $GW(k)$ exists if the same image was preferred in both pairs by the test participants, otherwise this metric will be inconclusive. The function that allows to calculate which is the preferred image is given by:

$$GW(k) = \begin{cases} IW(2k), & IW(2k) = IW(2k+1) \\ Inconclusive, & IW(2k) \neq IW(2k+1) \end{cases} \quad (44)$$

- Algorithm Analysis:

  Algorithm analysis is used to calculate the average algorithm confidence $\overline{AC}(alg)$ of all the individual confidence values $IC(i)$ where the individual winner image of a specific pair $IW(i)$ was an image generated from the given algorithm *alg* and can be calculated using:

$$\overline{AC}(alg) = \frac{1}{M}\sum_{i=1}^{M} IC(i)\,, \quad IW(i) = alg \quad (45)$$

- Overall Analysis:

  Overall analysis is used to calculate the average score $\overline{OS}()$ of all the individual score values $IS(i)$ and can be calculated using:

$$\overline{OS}() = \frac{1}{M}\sum_{i=1}^{M} IS(i) \quad (46)$$

6.1. Algorithm Comparison: SSP vs SSPB

As explained in Section 5.7.1. the goal of this test was to find out which of the two algorithms is better, so the number of votes of the participants will be the focus of this analysis. This test had 18 original pairs of images, but since each pair was showed in two different ways XY and YX the participants voted for 36 pairs.

Winner	Winning Pairs [0, 36]	Winning Pairs Percentage [0, 100 %]	Avg. Algorithm Confidence [0, 100 %]	Avg. Overall Score [-3, 3]
SSP (+)	2	5.56 %	50.00 %	-1.31
SSPB (-)	34	94.44 %	81.47 %	
Draw	0	0.00 %	Inconclusive	
Total	36	100.00 %		

Table 9: Winning individual analysis with avg. algorithm confidence value and avg. overall score.

Table 9 shows the results of the analysis made for the individual pairs. The SSPB algorithm was the chosen one in 94.44 % of the cases, got an average confidence of 81.47 % of the participant votes when was considered the best algorithm and got an overall average score of 1.31 being considered somewhere between the Slightly Better (1) and Better (2) scores when compared with the SSP algorithm.

After analyzing all the individual pairs, a group analysis was performed to understand if the results were consistent.

Winner	Winning Groups [0, 18]	Winning Groups Percentage [0, 100 %]
SSP	0	0.00 %
SSPB	16	88.89 %
Inconclusive	2	11.11 %
Total	18	100.00 %

Table 10: Winning group analysis for SSP vs SSPB.

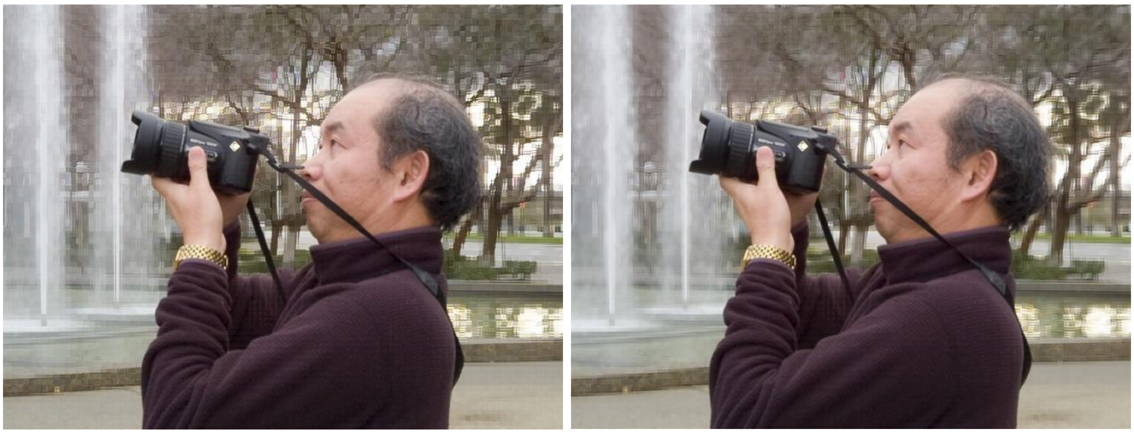


Figure 60. Jeff image rendered by SSP (PS=11) and SSPB (PS=11, WS=15).

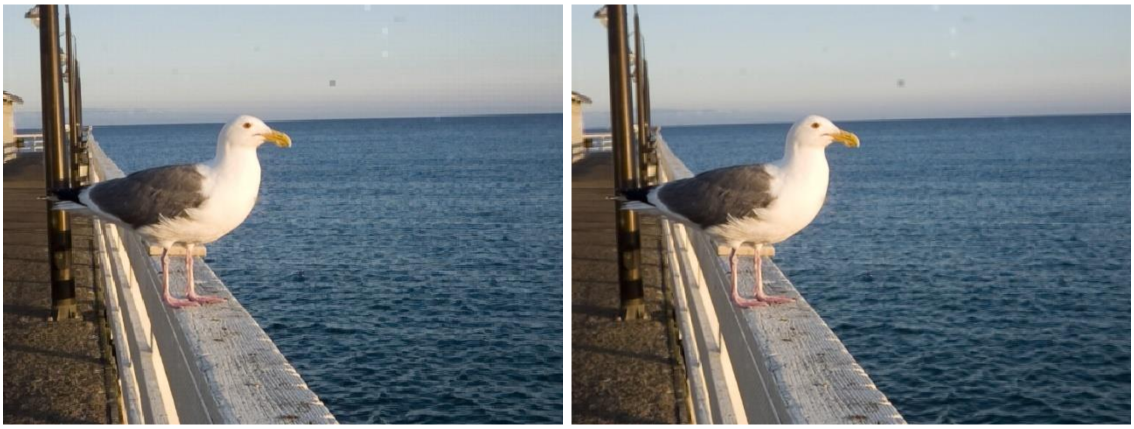


Figure 61. Seagull image rendered by SSP (PS=9) and SSPB (PS=9, WS=13).

Figure 60 and 61 present the 2 grouped pairs that were inconclusive, meaning that two different winners were selected the two times this pair was showed, both have the object in the foreground on focus, and the WS is not that big when compared with the PS in both cases, making hard to decide which one of the images is better.

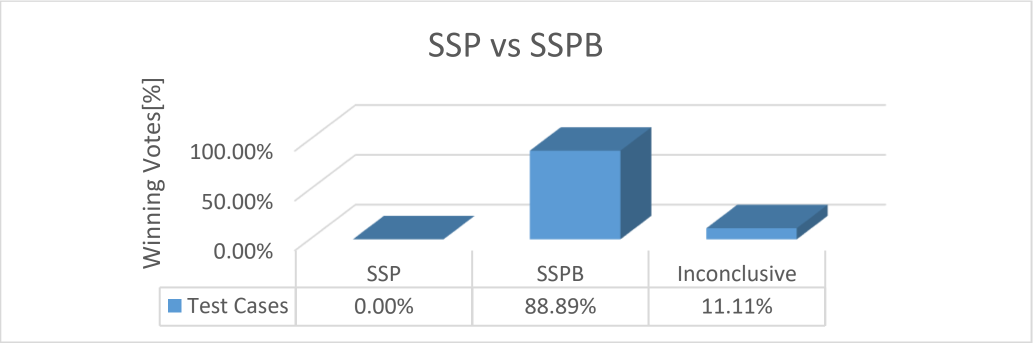


Figure 62. This chart illustrates the winning grouped pairs percentage of this test case.

Table 10 and Figure 62 display the results of the group analysis and once again the SSPB algorithm displays a lot of better results when compared with the SSP algorithm, making it easy to conclude that the SSPB is the best algorithm.

6.2. SSPB: The Influence of the Patch Size Parameter

As explained in Section 5.7.2. the goal of this test was to understand the impact of the patch size parameter in the SSPB algorithm, for each image, multiple images were generated with different patch sizes, since each different patch size focus a different plane of the image. All the combinations of those generated images were tested so the number of votes of the participants will be the focus of this analysis to understand for each image content what was the participants preference. This test had 18 original images, since we want to test all the combinations possible, 15 pairs were created but since each pair was showed in two different ways XY and YX the participants voted for 30 pairs. An individual analysis was made for each image. Due to the size of the following images the zones on focus were marked with an orange circle.

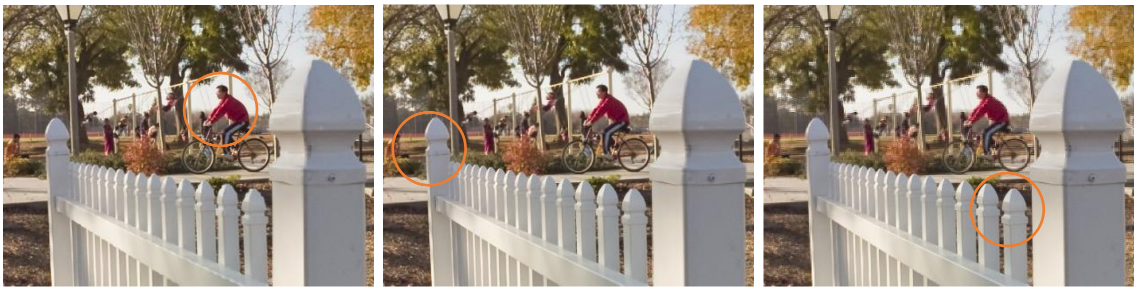


Figure 63. Bike image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Middle	0	2	80.00 %	+1.33
Middle	Foreground	2	0	95.00 %	-2.67
Background	Foreground	2	0	100.00 %	-2.75
Total		6			

Table 11: Group analysis combinations results for Bike.

Image	Winning Pairs [0, 6]	Winning Pairs Percentage [0, 100 %]
Background	2	33.33 %
Middle	4	66.66 %
Foreground	0	0.00 %
Total	6	100.00 %

Table 12: Winning individual analysis results for Bike.

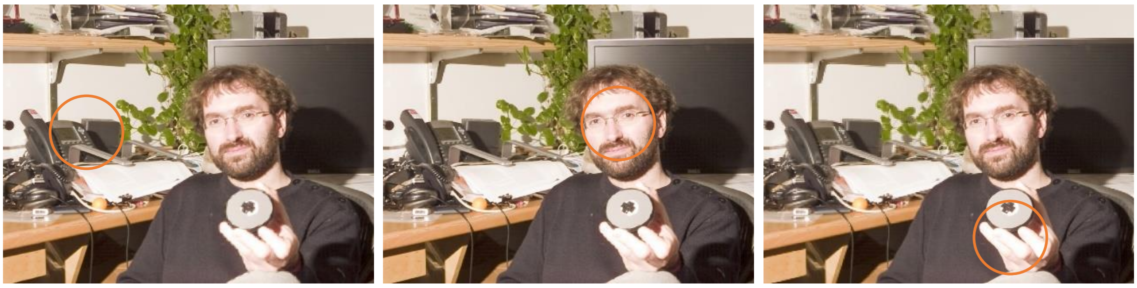


Figure 64. Fredo image with 3 images with different focal planes: Background (PS=9, WS=13), Middle (PS=11, WS=15), Foreground (PS=13, WS=19).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Middle	0	2	85.00 %	2.11
Middle	Foreground	2	0	85.00 %	-1.58
Background	Foreground	0	2	80.00 %	1.67
Total		6			

Table 13: Group analysis combinations results for Fredo.

Image	Winning Pairs [0, 6]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Middle	4	66.66 %
Foreground	2	33.33 %
Total	6	100.00 %

Table 14: Winning individual analysis results for Fredo.

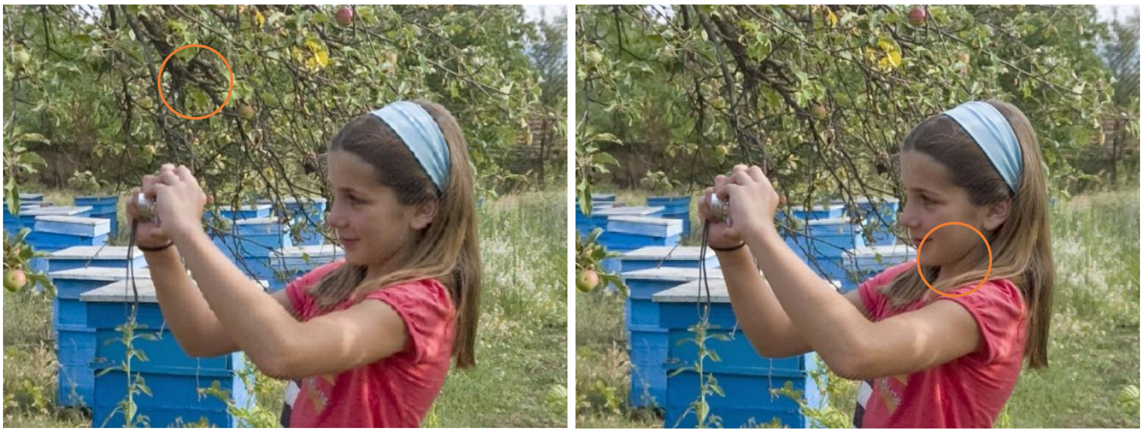


Figure 65. Laura image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Foreground	0	2	100.00 %	2.10
Total		2			

Table 15: Group analysis combinations results for Laura*.*

Image	Winning Pairs [0, 2]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Foreground	2	100.00 %
Total	2	100.00 %

Table 16: Winning individual analysis results for Laura*.*

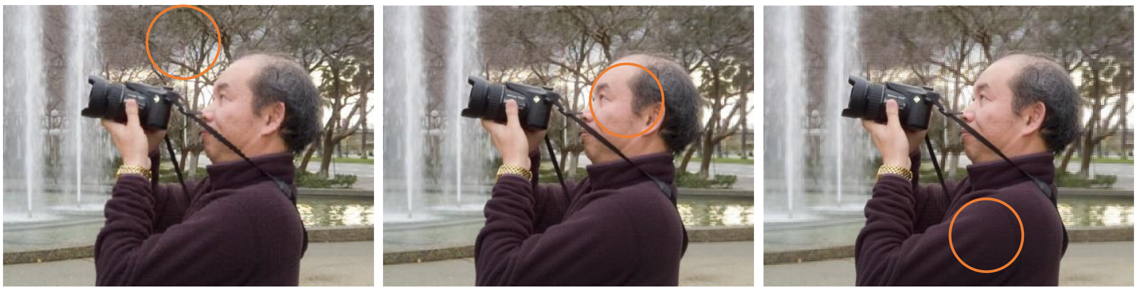


Figure 66. Jeff image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Middle	0	2	100.00 %	2.38
Middle	Foreground	1	1	50.00 %	0.05
Background	Foreground	0	2	75.00 %	1.22
Total		6			

Table 17: Group analysis combinations results for Jeff*.*

Image	Winning Pairs [0, 6]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Middle	3	50.00 %
Foreground	3	50.00 %
Total	6	100.00 %

Table 18: Winning individual analysis results for Jeff*.*

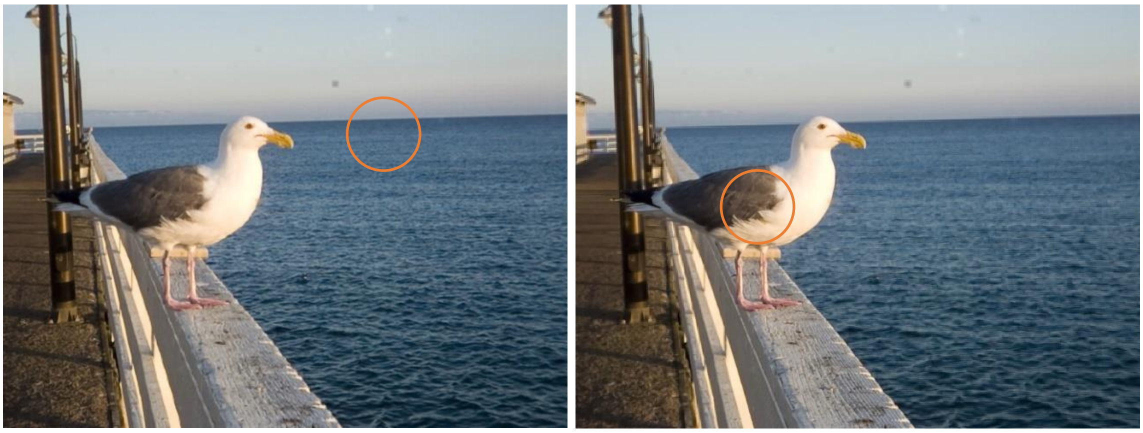


Figure 67. Seagull image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Foreground	0	2	60.00 %	0.85
Total		2			

Table 19: Group analysis combinations results for Seagull.

Image	Winning Pairs [0, 2]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Foreground	2	100.00 %
Total	2	100.00 %

Table 20: Winning individual analysis results for Seagull.

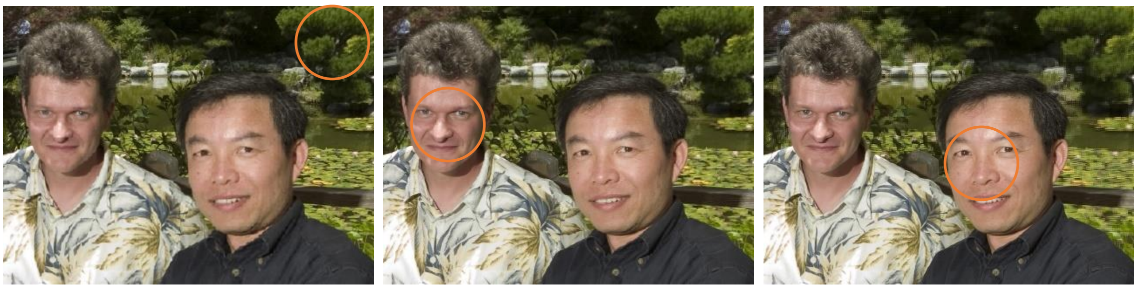


Figure 68. Sergio image with 3 images with different focal planes: Background (PS=7, WS=9), Middle (PS=9, WS=13), Foreground (PS=11, WS=15).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Middle	0	2	100.00 %	2.72
Middle	Foreground	0	2	60.00 %	0.45
Background	Foreground	0	2	95.00 %	2.21
Total		6			

Table 21: Group analysis combinations results for Sergio.

Image	Winning Pairs [0, 6]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Middle	2	33.33 %
Foreground	4	66.66 %
Total	6	100.00 %

Table 22: Winning individual analysis results for Sergio.

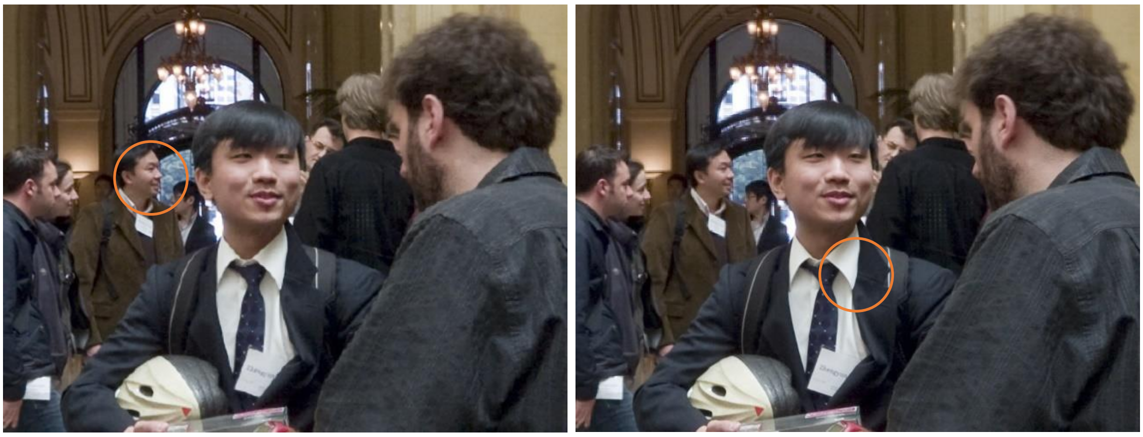


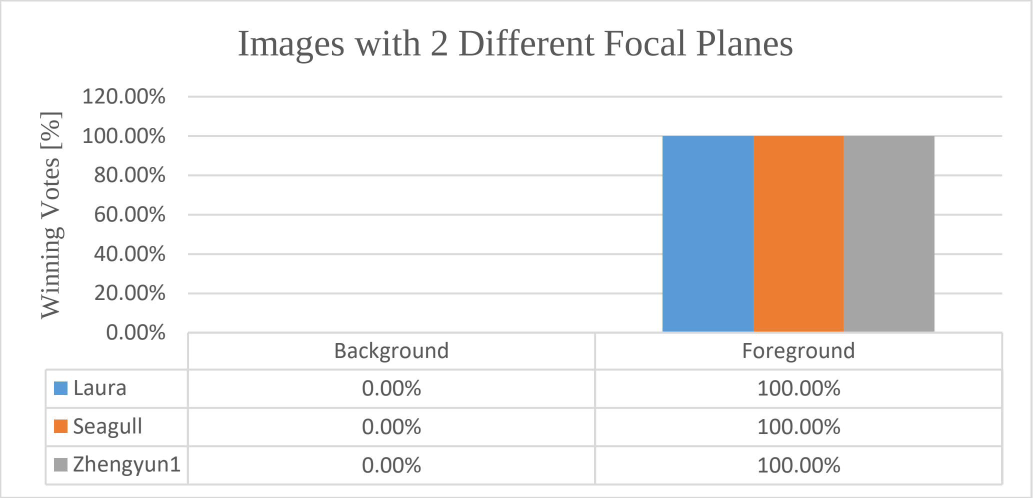
Figure 69. Zhengyun1 image with 2 images with different focal planes: Background (PS=7, WS=9), Foreground (PS=9, WS=13).

Image A (-)	Image B (+)	Image A Winning Pairs [0, 2]	Image B Winning Pairs [0, 2]	Avg. Group Confidence [0, 100 %]	Avg. Group Score [-3, 3]
Background	Foreground	0	2	80.00 %	1.55
Total		2			

Table 23: Group analysis combinations results for Zhengyun1.

Image	Winning Pairs [0, 2]	Winning Pairs Percentage [0, 100 %]
Background	0	0.00 %
Foreground	2	100.00 %
Total	2	100.00 %

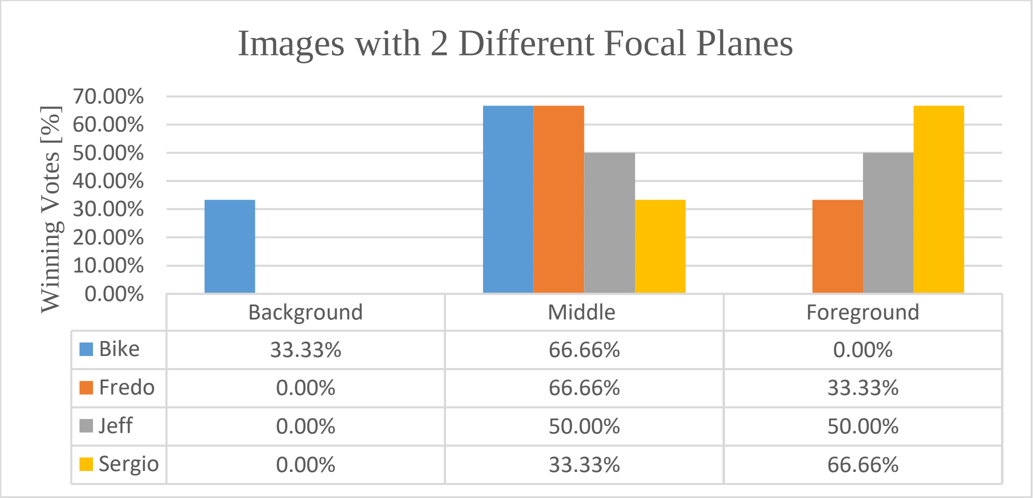
Table 24: Winning individual analysis results for Zhengyun1.



	Background	Foreground
🟦 Laura	0.00%	100.00%
🟧 Seagull	0.00%	100.00%
■ Zhengyun1	0.00%	100.00%

Figure 70. Images with 2 focal planes comparison.

For the images that only had two images with different focal planes the participant votes are clear, an image that has the foreground in focus is better (2) than the image that has the background in focus.



	Background	Middle	Foreground
🟦 Bike	33.33%	66.66%	0.00%
🟧 Fredo	0.00%	66.66%	33.33%
■ Jeff	0.00%	50.00%	50.00%
🟨 Sergio	0.00%	33.33%	66.66%

Figure 71. Images with 3 focal planes comparison.

For the images that have three images with different focal planes, Figure 71, the participants votes aren’t as clear as in Figure 70. The image content is a super important factor here, lets analyze every image:

- For Bike, the image that has the focus in the middle plane won with 66.66% of the votes. After viewing 3 images in detail, the middle image was the one selected because when the foreground is on focus a mix of blur and artifacts are placed in the background. When the background is on focus, the same happens to the foreground. The image where the middle plane is on focus does not create as much blur/artifacts in the other planes since it is close from the other two. The second image with more votes was the image where the background is on focus, since the content of this image the background occupies a big area of the image, it's better to have the foreground with artifacts than the background. Another thing to notice here is that this is the only image that it is not a portrait photograph, this might be a factor too, since in portraits the face of the subject is regularly one of the most interesting zones of the image for the users.

- For Fredo, the image with more votes is the middle one, where the guy's face is on focus. The image where the focus is on the foreground, the guy's hand is on focus and his face is not that blurry, being the second plane with the most votes.

- For Jeff, in the middle and foreground images his face is on focus in both images and that explains the fact that both images tie with the highest number of votes.

- For Sergio, there are 2 persons in this image, in the middle image the guy on the left has his face on focus, in the foreground image, both have their face on focus. Once again, the fact that this is a portrait, explains the reason why the foreground image was the one with more votes and the middle image the second with more votes as one of the faces was on focus.

The patch size is an extremely important parameter for all the textured based rendering algorithms, since as showed here, different patch sizes are associates with different qualities. When selecting the right patch size for a specific image, the image content should be taken into consideration! One conclusion we can easily take from Figure 71 is that if the image is a portrait, all the faces should be on focus.

6.3. SSPB: The Influence of the Window Size Parameter

As explained in Section 5.7.3. the goal of this test was to understand the impact of the window size parameter in the SSPB algorithm, for each image, multiple images were generated with different window sizes, which will make images with more or less blur, for the same PS, the bigger the WS, the more blur the image have. All the combinations of those generated images were tested so the number of votes of the participants will be the focus of this analysis to understand for each image content what was the participants preference. This test had 24 original images, since we want to test all the combinations possible, 36 pairs were created but since each pair was showed in two different ways XY and YX the participants voted for 72 pairs. An individual analysis was made for each image.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=11, WS=11)	0	0.00 %
1.50 (PS=11, WS=15)	4.5	37.50 %
1.75 (PS=11, WS=19)	5.5	45.83 %
2.00 (PS=11, WS=23)	2	16.66 %
Total	12	100.00 %

Table 25: Individual analysis results for Fredo.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=9, WS=9)	0	0.00 %
1.50 (PS=9, WS=13)	4	33.33 %
1.75 (PS=9, WS=17)	5	41.66 %
2.00 (PS=9, WS=21)	4	25.00 %
Total	12	100.00 %

Table 26: Individual analysis results for Jeff.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=9, WS=9)	0.5	4.16 %
1.50 (PS=9, WS=13)	4	33.33 %
1.75 (PS=9, WS=17)	3.5	29.16 %
2.00 (PS=9, WS=21)	4	33.33 %
Total	12	100.00 %

Table 27: Individual analysis results for Laura.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=9, WS=9)	0	0.00 %
1.50 (PS=9, WS=13)	5	41.66 %
1.75 (PS=9, WS=17)	3	25.00 %
2.00 (PS=9, WS=21)	4	33.33 %
Total	12	100.00 %

Table 28: Individual analysis results for Seagull.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=11, WS=11)	1.5	12.50 %
1.50 (PS=11, WS=15)	3.5	29.16 %
1.75 (PS=11, WS=19)	5	41.66 %
2.00 (PS=11, WS=23)	2	16.66 %
Total	12	100.00 %

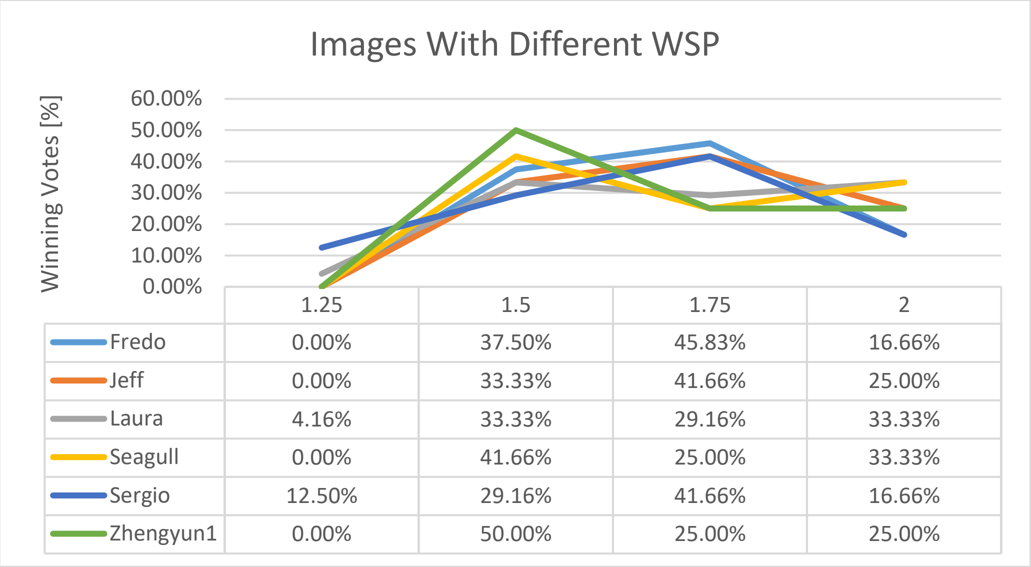
Table 29: Individual analysis results for Sergio.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25 (PS=9, WS=9)	0	0.00 %
1.50 (PS=9, WS=13)	6	50.00 %
1.75 (PS=9, WS=17)	3	25.00 %
2.00 (PS=9, WS=21)	3	25.00 %
Total	12	100.00 %

Table 30: Individual analysis results for Zhengyun1.

WSP	Avg. Winning Pairs Percentage [0, 100 %]
1.25	2.78 %
1.50	37.50 %
1.75	34.72 %
2.00	25.00 %
Total	100.00 %

Table 31: Average individual analysis results.



	1.25	1.5	1.75	2
Fredo	0.00%	37.50%	45.83%	16.66%
Jeff	0.00%	33.33%	41.66%	25.00%
Laura	4.16%	33.33%	29.16%	33.33%
Seagull	0.00%	41.66%	25.00%	33.33%
Sergio	12.50%	29.16%	41.66%	16.66%
Zhengyun1	0.00%	50.00%	25.00%	25.00%

Figure 72. Images with different blur intensities comparison.

Figure 72 show us that the test participants like blurry images but not too blurry. The images who got more votes have a WSP of 1.5 or 1.75, there is not a pattern or an optimal value which means that the WSP might not be as important as some other parameters like patch size.

6.4. Algorithm Comparison: DM vs DB

As explained in Section 5.7.4. the goal of this test was to find out which of the two algorithms is better, so the number of votes of the participants will be the focus of this analysis. This test had 24 original pairs of images, but since each pair was showed in two different ways XY and YX the participants voted for 48 pairs.

Winner	Winning Pairs [0, 48]	Winning Pairs Percentage [0, 100 %]	Avg. Algorithm Confidence [0, 100 %]	Avg. Overall Score [-3, 3]
DM (+)	4	8.33 %	65.00 %	- 0.89
DB (-)	43	89.58 %	71.16 %	
Draw	1	2.08 %	40.00 %	
Total	48	100.00 %		

Table 32: Winning individual analysis with avg. algorithm confidence value and avg. overall score.

Table 32 shows the results of the analysis made for the individual pairs. The DB algorithm was the chosen one in 89.58 % of the cases, got an average of 71.16 % of the participant votes when was considered the best algorithm and got an average score of 0.89 being considered somewhere between The Same (0) and Slightly Better (1) scores when compared with the DM algorithm.

After analyzing all the individual pairs, a group analysis was performed in order to understand if the results were consistent.

Winner	Winning Groups [0, 24]	Winning Groups Percentage [0, 100 %]
DM	1	4.16 %
DB	20	83.33 %
Inconclusive	3	12.50 %
Total	24	100.00 %

Table 33: Winning group analysis for DM vs DB.

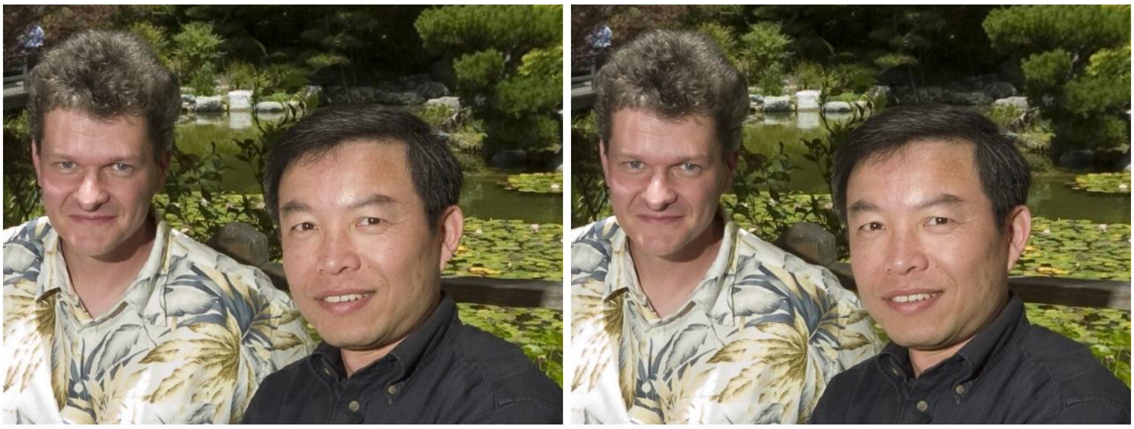


Figure 73. Sergio image rendered by DM (α=0.0625) and DB (α=0.0625, WSP=1.5).

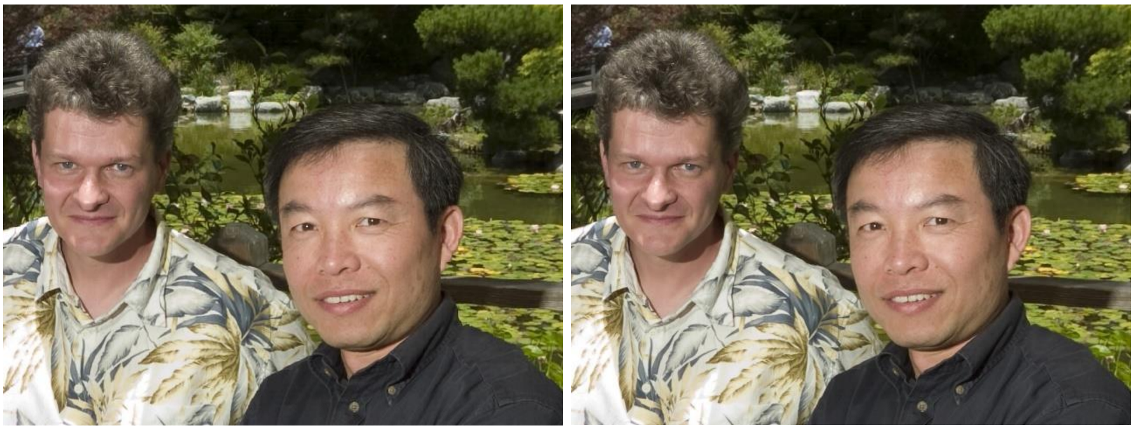


Figure 74. Sergio image rendered by DM (α=0.0) and DB (α=0.0, WSP=1.5).

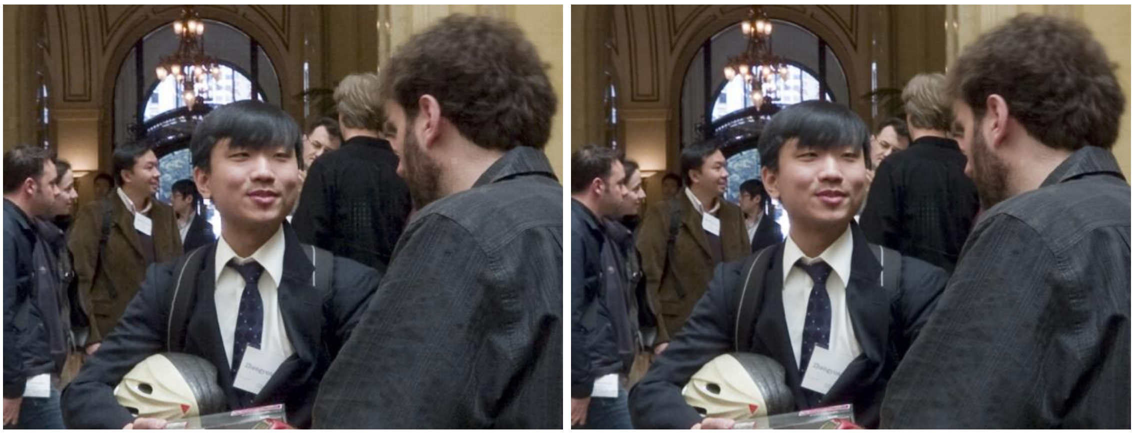


Figure 75. Zhengyun1 image rendered by DM (α=0.0625) and DB (α=0.0625, WSP=1.5).

Figure 73, 74 and 75 present the 3 grouped pairs that were inconclusive, meaning that two different winners were selected the two times this pair was showed, both algorithms are all-in-focus and the quality of both is extremely good. The DB algorithm has extra blending process, creating some blur, but since both images are all-in-focus the number of artifacts is a lot smaller when compared with the comparison of the SSP and SSPB algorithms that have a lot of different planes out of focus, which makes a lot harder to decide which one of these two algorithms is better.

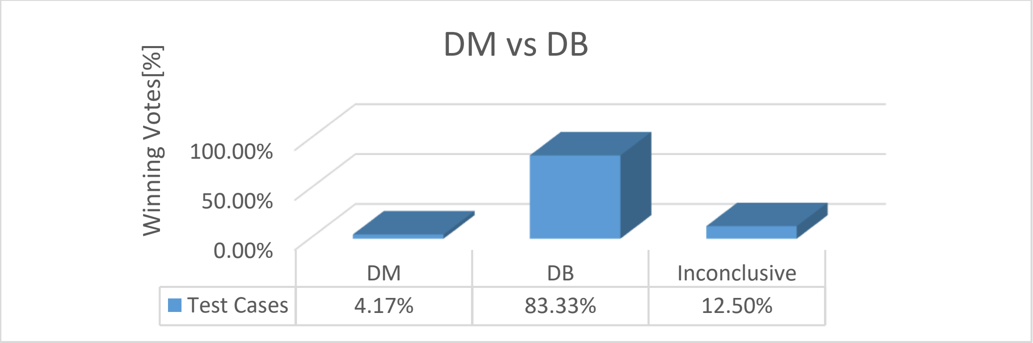


Figure 76. This chart illustrates the winning grouped pairs percentage of this test case.

Table 33 and Figure 76 display the results of the grouped pair analysis and once again the DB algorithm displays a lot of better results when compared with the DM algorithm, making it easy to conclude that the DB is the best algorithm.

6.5. DB: The Influence of the Window Size Percentage Parameter

As explained in Section 5.7.5. the goal of this test was to understand the impact of the window size percentage parameter in the DB algorithm, for each image, multiple images were generated with different window size percentages, which generates images with different amounts of blur. All the combinations of those generated images were tested so the number of votes of the participants will be the focus of this analysis to understand for each image content what was the participants preference. This test had 24 original images, since we want to test all the combinations possible, 36 pairs were created but since each pair was showed in two different ways XY and YX the participants voted for 72 pairs. An individual analysis was made for each image.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	0	0.00 %
1.50	2.5	20.83 %
1.75	4.5	37.50 %
2.00	5	41.67 %
Total	12	100.00 %

Table 34: Individual analysis results for Fredo.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	3	25.00 %
1.50	4.5	37.50 %
1.75	2.5	20.83 %
2.00	2	16.67 %
Total	12	100.00 %

Table 35: Individual analysis results for Jeff.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	0	0.00 %
1.50	2.5	20.83 %
1.75	4	33.33 %
2.00	5.5	45.83 %
Total	12	100.00 %

Table 36: Individual analysis results for Laura.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	0	0.00 %
1.50	3	25.00 %
1.75	3	25.00 %
2.00	6	50.00 %
Total	12	100.00 %

Table 37: Individual analysis results for Seagull.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	3	25.00 %
1.50	4.5	37.50 %
1.75	2.5	20.83 %
2.00	2	16.67 %
Total	12	100.00 %

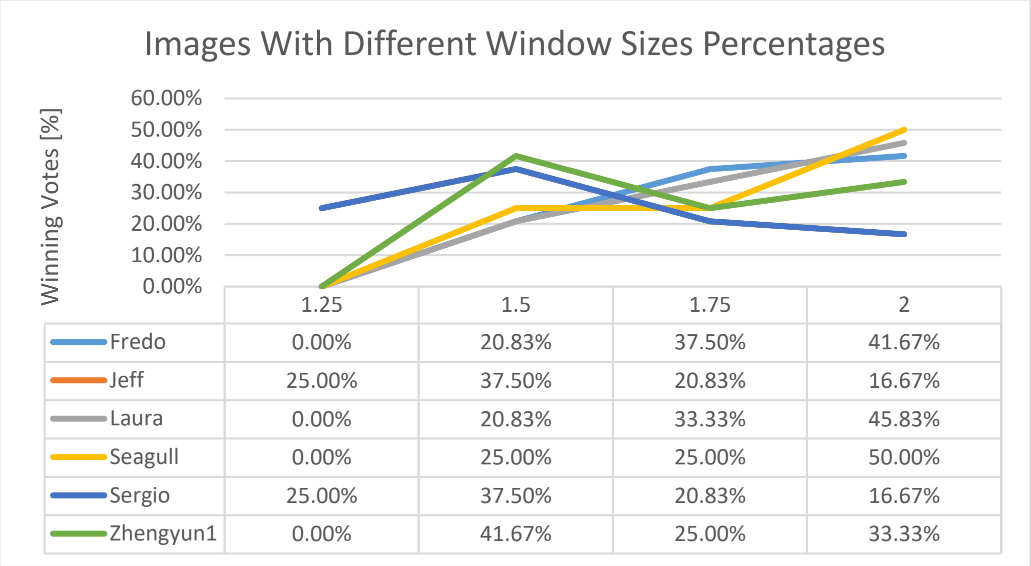
Table 38: Individual analysis results for Sergio.

WSP	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]
1.25	0	0.00 %
1.50	5	41.67 %
1.75	3	25.00 %
2.00	4	33.33 %
Total	12	100.00 %

Table 39: Individual analysis results for Zhengyun1.

WSP	Avg. Winning Pairs Percentage [0, 100 %]
1.25	8.33 %
1.50	30.56 %
1.75	27.08 %
2.00	34.03 %
Total	100.00 %

Table 40: Average individual analysis results.



	1.25	1.5	1.75	2
Fredo	0.00%	20.83%	37.50%	41.67%
Jeff	25.00%	37.50%	20.83%	16.67%
Laura	0.00%	20.83%	33.33%	45.83%
Seagull	0.00%	25.00%	25.00%	50.00%
Sergio	25.00%	37.50%	20.83%	16.67%
Zhengyun1	0.00%	41.67%	25.00%	33.33%

Figure 77. Images with different blur intensities comparison.

Figure 77 show us that the test participants like blurry images. The images who got more votes have a WSP of 1.5 or 2.0, there is not a pattern or an optimal value which means that the WSP might not be as important as some other parameters like patch size.

6.6. Algorithm Comparison: SSPB vs DB

As explained in Section 5.7.6. the goal of this test was to find out which of the two algorithms is better, so the number of votes of the participants will be the focus of this analysis. This test had 6 original pairs of images, but since each pair was showed in two different ways XY and YX the participants voted for 12 pairs.

Winner	Winning Pairs [0, 12]	Winning Pairs Percentage [0, 100 %]	Avg. Algorithm Confidence [0, 100 %]	Avg. Overall Score [-3, 3]
SSP (+)	0	0.00 %	Inconclusive	- 1.35
DB (-)	12	100.00 %	74.17 %	
Draw	0	0.00 %	Inconclusive	
Total	12	100.00 %		

Table 41: Winning individual analysis with avg. algorithm confidence value and avg. overall score.

Table 41 shows the results of the analysis made for the individual pairs. The DB algorithm was the chosen one in 100 % of the cases, got an average of 74.17 % of the participant votes when was considered the best algorithm and got an overall average score of 1.35 being considered somewhere between the Slightly Better (1) and Better (2) scores when compared with the SSPB algorithm.

After analyzing all the individual pairs, we grouped the scores of the two versions of each pair and of course all indicated that the DB algorithm was the winner of every group pair as it got 100 % of the votes on the individual pairs.

Winner	Winning Groups [0, 6]	Winning Groups Percentage [0, 100 %]
SSP	0	0.00 %
DB	6	100.00 %
Inconclusive	0	0.00 %
Total	6	100.00 %

Table 42: Winning group analysis for SSPB vs DB.

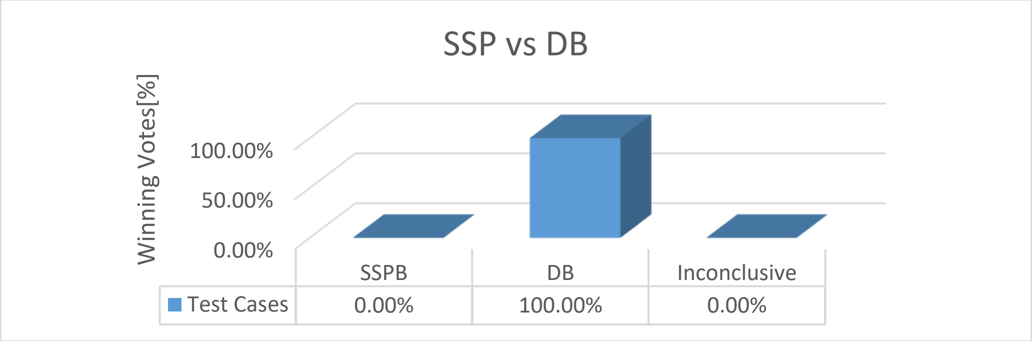


Figure 78. This chart illustrates the winning grouped pairs percentage of this test case.

Table 42 and Figure 78 display the results of the grouped pair analysis and once again the DB is without a doubt the best algorithm between the two.

Chapter 7 – Conclusions and Future Work

7.1. Conclusions

In this dissertation the problem of the current display technology not being ready to work with 4D content was addressed. Different 2D lenslet rendering algorithms for plenoptic cameras 2.0 were implemented and evaluated. These algorithms were referenced and described for a better understanding of their properties and to comprehend the decisions made to implement each one of those algorithms from scratch.

Two different types of solutions were presented, textured based rendering solutions that only allow to have one plane on focus and disparity-based rendering solutions that after estimating the disparity map can render an all-in-focus image of the scene. For this second type of solutions an improved disparity estimation model, that is influenced by previous estimations of the top and left neighbor estimations and contains the *a* parameter so the user can have some control on the desired estimation process, was suggested.

The major contributions that this dissertation made for the image processing community were an easy to use and easy to scale light field software application that includes different light field rendering algorithms, a software application that allows to perform subjective tests using the pair comparison simultaneous presentation methodology and the definition of subjective tests to evaluate the quality of the rendering algorithms.

Quality assessment tests were performed based on recommendation ITU-T P910, with the intent to compare the performance of different rendering solutions and the impact of some algorithm input parameters.

The quality assessment test results were analyzed using individual, group, algorithm and overall analysis in order to get key and concrete conclusions about the participant's preferences.

After analyzing the data prevenient from the different analysis executed for each one of the six tests, all the research questions were answered based on the obtained conclusions:

- The textured based rendering solution that got better results in the quality assessment tests was the SSPB algorithm winning 88.89 % of winning groups percentage and considered between Slightly Better (1) and Better (2) when compared with the other tested textured base algorithm, the SSP.

- The disparity-based rendering solution that got better results in the quality assessment tests was the DB algorithm with 83.33 % of the winning groups percentage and considered between The Same (0) and Slightly Better (1) when compared with the other tested disparity-base algorithm, the DM. Both disparity-based solutions present all-in-focus rendered images with great image quality, which make harder to conclude which algorithm is better.

- From the two types of rendering solutions, the disparity-based rendering algorithm DB got better results in the quality assessment tests with 100.00 % of the winning groups percentage and considered between Slightly Better (1) and Better (2) when compared with the best textured base algorithm, the SSPB. The rendered image from the DB algorithm is an all-in-focus image, while the rendered image from the SSPB algorithm has only one plane on focus, the other planes will have a mix of blur and artifacts since the patch does not make those planes on focus. This way we can conclude that the test participants prefer unrealistic all-in-focus images and that all the in all the algorithm comparisons tested, the most sophisticated and complex solutions performed better when compared with the simplest ones.

- With the analysis performed to determine the impact of the patch size (PS), was concluded that it has a huge impact in the quality of the textured based algorithm SSPB and that is extremely dependent of the image content as can be analyzed in detail in the previous chapter. Almost every image used for the tests was a portrait and was concluded that the preferred patch size would always correspond to the focal plane that would have the face of the main subject on focus.

- With the analysis performed to determine the impact of the window size (WS) in the textured based solution SSPB and the impact of the window size percentage (WSP) in the disparity-based solution DB, as can be seen with more detail in the previous chapter, none of the tests showed big preferences by the participants, which permitted to conclude that these parameters aren’t that important in order to obtain a better image quality.

7.2. Future Work

This dissertation has been focused on the implementation and evaluation of rendering algorithms and rendering approaches for lenslet light field imaging. This section presents some of work items directly related to the topics considered in this dissertation, which are worthwhile to be pursued in the future, namely:

- It would be interesting to compare the SSPB algorithm with the DB not in terms of image quality but in terms of user preference as in this context an image with worst quality might be more appealing to the user, as the subject of the photo stands out due the blurry background.

- It would be interesting to analyze the performance of these algorithms using a non-portrait image set, specially the impact of the patch size (PS) parameter, as the preferred PS for the participants corresponded always to the plane where the main subject of the photo was on focus, as analyzed in the previous chapter.

- The rendering software tool was developed for desktop and performance wasn’t a priority. Depending on the input parameters of the all-in-focus solutions, the estimation of the disparity map can be time consuming. This way, it would be interesting to study ways to improve the performance of these algorithms using the GPU instead of the CPU. There is already some work in this field presented by A. Lumsdaine, G. Chunev and T. Georgiev in [32].

- Light Field Imaging standardizations such as MPEG-I and JPEG Pleno will have to deal with high amounts of data, requiring extremely efficient coding [18][19] and efficient compression where quality assessment tests would be ideal to evaluate those compression results.

Appendices

Appendix A – The relation between Sigma and Window Size

The following chart allows to compare gaussian curves using different sigma values.

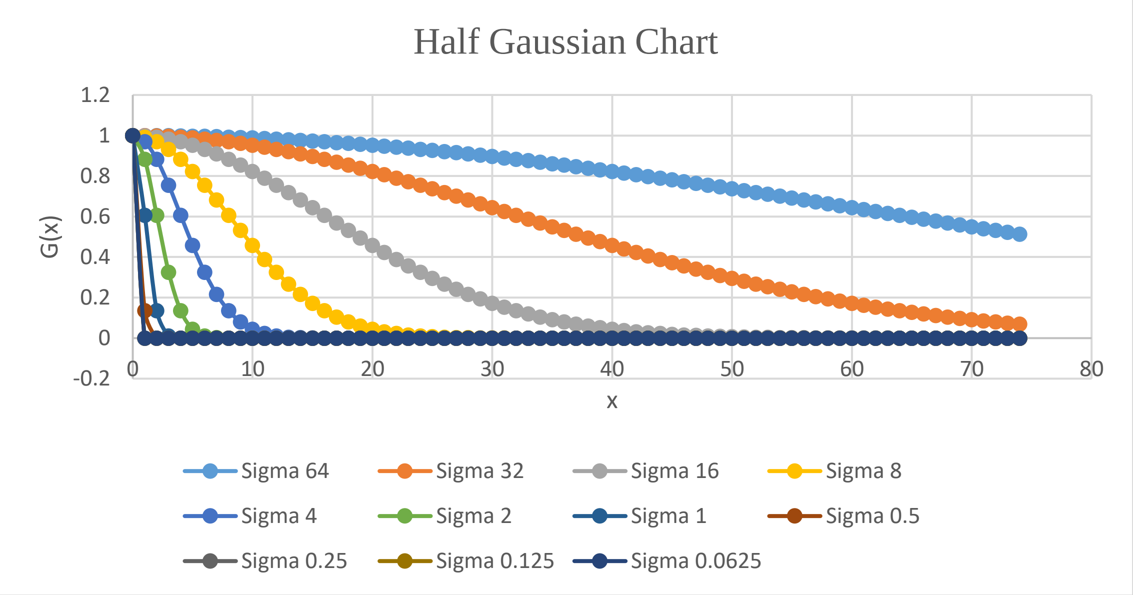


Figure 79. This chart illustrates half of the Gaussian function to understand the relation between the pixel weight G(x) and half of the window size (x).

In the quality assessment tests, the sigma value used was eight and the reason was that after analyzing the data presented in the Appendix B, was concluded that the maximum estimated patch size, for the six tested images, was twenty-two. Since the highest WSP that was used was two, the maximum WS value was forty-four.

With a small sigma the gaussian curve would go from one to zero extremely fast and with a big sigma the curve would go from one to zero extremely slow. This way a curve that would reach zero at a distance of less twenty-two or positive twenty-two, forty-four in total, would be the ideal as all the values would have a significant value.

Appendix B – The impact of the α parameter in the Disparity Estimation

The following tables display metrics extracted from different disparity estimation processes of the same images, varying the α parameter.

	Estimated Patch Size values for Zhengyun1					
α	Min	Max	Mean	Median	Mode	Std
0	0	20	8.53494152	9	7	1.706182813
0.0078125	1	14	8.526900585	9	7	1.67519966
0.015625	2	13	8.507894737	9	7	1.670065556
0.03125	3	11	8.477923977	8	7	1.649059032
0.0625	3	11	8.371783626	8	7	1.567187028
0.125	0	11	7.936988304	8	7	1.600248608
0.25	0	11	6.837280702	7	7	2.17541733
0.5	0	10	3.757163743	5	0	3.289306815

Table 43: Metrics extracted from the disparity estimation processes of the image Zhengyun1 with different values of α.

	Estimated Patch Size values for Jeff					
α	Min	Max	Mean	Median	Mode	Std
0	0	19	8.294590643	7	7	1.767039015
0.0078125	1	16	8.317105263	7	7	1.756241946
0.015625	2	16	8.319298246	7	7	1.740078502
0.03125	6	11	8.319444444	7	7	1.698942477
0.0625	6	11	8.214473684	7	7	1.565451521
0.125	3	11	7.899415205	7	7	1.574040024
0.25	0	11	7.03245614	7	7	2.045205302
0.5	0	11	2.793567251	0	0	3.359122104

Table 44: Metrics extracted from the disparity estimation processes of the image Jeff with different values of α.

	Estimated Patch Size values for Fredo					
α	Min	Max	Mean	Median	Mode	Std
0	0	19	9.030671296	9	9	2.967013325
0.0078125	0	19	9.084201389	9	9	2.810184098
0.015625	1	17	9.310329861	9	9	2.460078901
0.03125	3	15	9.654079861	9	9	1.599691853
0.0625	8	15	9.729311343	9	9	1.421485957
0.125	8	15	9.689670139	9	9	1.370855625
0.25	4	15	9.695457176	9	9	1.230280648
0.5	0	15	9.298755787	9	9	1.206540174

Table 45: Metrics extracted from the disparity estimation processes of the image Fredo with different values of α.

	Estimated Patch Size values for Laura					
α	Min	Max	Mean	Median	Mode	Std
0	0	16	8.056278935	8	8	1.152638096
0.0078125	1	16	8.047019676	8	8	1.145120903
0.015625	1	13	8.038049769	8	8	1.125706627
0.03125	1	11	8.021701389	8	8	1.108073852
0.0625	1	11	7.978443287	8	8	1.041651948
0.125	6	10	7.900318287	8	8	0.890919932
0.25	7	10	7.759403935	8	8	0.758715634
0.5	0	10	7.111400463	7	7	1.425027495

Table 46: Metrics extracted from the disparity estimation processes of the image Laura with different values of α.

	Estimated Patch Size values for Seagull					
α	Min	Max	Mean	Median	Mode	Std
0	0	22	6.36328125	7	7	2.114242952
0.0078125	0	22	6.446469907	7	7	1.829944543
0.015625	0	13	6.888454861	7	7	1.156891007
0.03125	0	14	7.054398148	7	7	0.966417166
0.0625	0	13	7.134982639	7	7	0.976063433
0.125	0	10	6.872974537	7	7	1.067761458
0.25	0	10	6.718460648	7	6	1.054375557
0.5	0	10	6.406394676	6	6	1.033260341

Table 47: Metrics extracted from the disparity estimation processes of the image Seagull with different values of α.

	Estimated Patch Size values for Sergio					
α	Min	Max	Mean	Median	Mode	Std
0	6	16	9.305266204	10	7	1.951965427
0.0078125	7	14	9.294849537	10	7	1.945477032
0.015625	7	14	9.28587963	10	7	1.937877123
0.03125	7	13	9.262876157	10	7	1.921165991
0.0625	7	12	9.223958333	10	7	1.893019604
0.125	7	12	9.123408565	10	7	1.855911028
0.25	3	12	8.732783565	9	7	1.597408711
0.5	0	12	4.216579861	4	0	3.956117434

Table 48: Metrics extracted from the disparity estimation processes of the image Sergio with different values of α.

The following charts allow to compare the extracted metrics of the different estimation processes for the six images used.

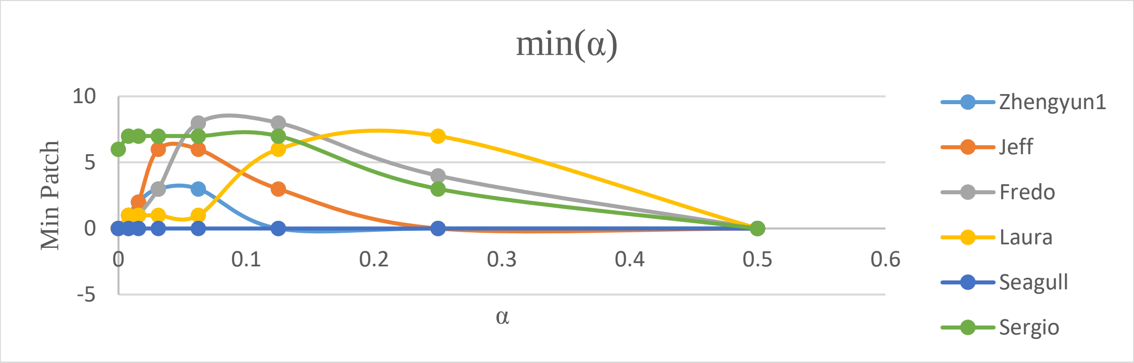


Figure 80. This chart illustrates the minimum estimated patch size for different disparity estimation processes, using different values of α.

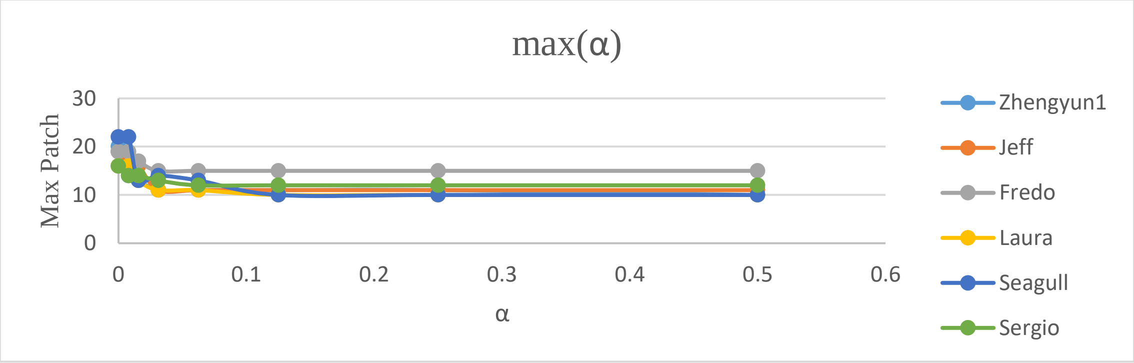


Figure 81. This chart illustrates the maximum estimated patch size for different disparity estimation processes, using different values of α.

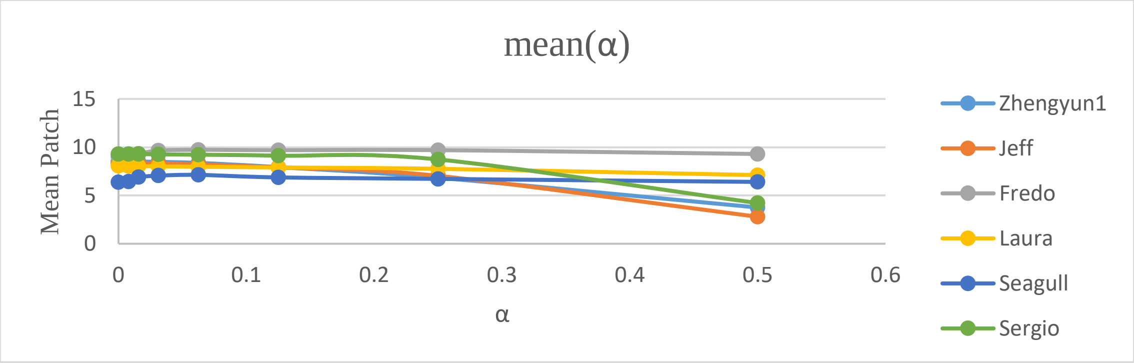


Figure 82. This chart illustrates the average estimated patch size for different disparity estimation processes, using different values of α.

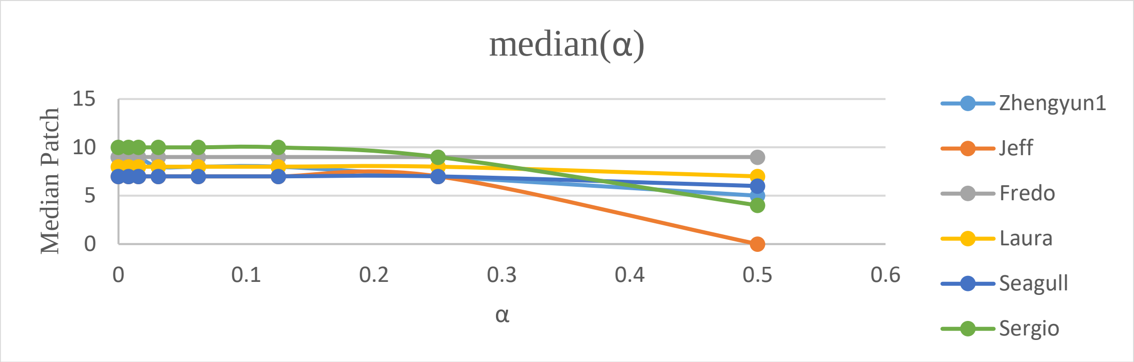


Figure 83. This chart illustrates the median of the estimated patch size for different disparity estimation processes, using different values of α.

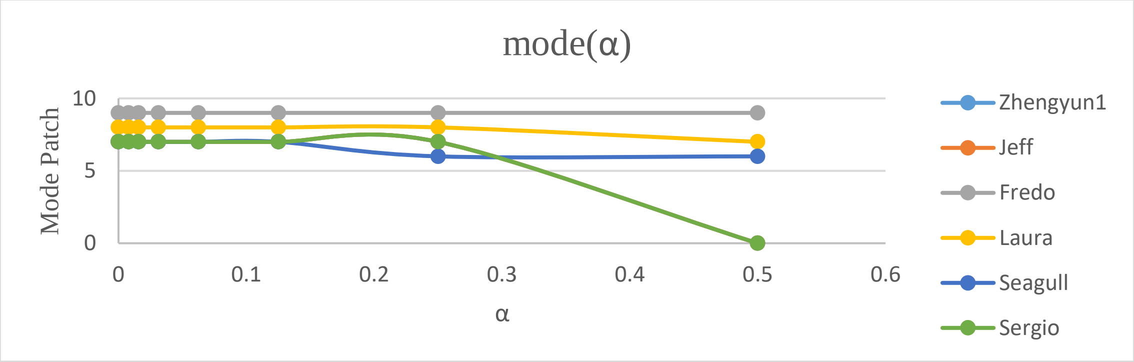


Figure 84. This chart illustrates the mode of the estimated patch size for different disparity estimation processes, using different values of α.

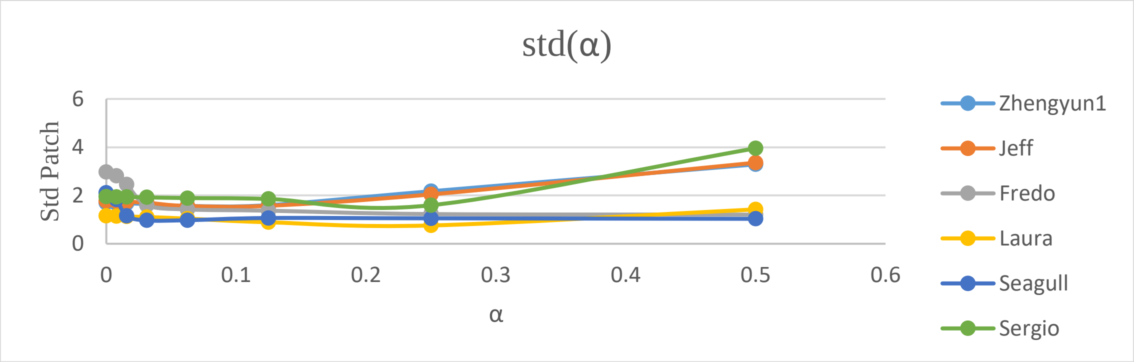


Figure 85. This chart illustrates the standard deviation of the estimated patch size for different disparity estimation processes, using different values of α.

Bibliography

[1] M. Faraday, "Throughts on Ray Vibrations," *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science,* vol. 28, pp. 345-350, 1846.

[2] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision In Computation Models of Visual Processing," in MIT Press, Cambridge, 1991.

[3] C. DeCusatis, “APPENDIX I The SI System and SI Units for Radiometry and Photometry,” em Handbook of Applied Photometry, American Institute of Physics, 1997, p. 484.

[4] J. E. Greivenkamp, “Physical Optics,” Journal of the Optical Society of America A, vol. 28, nº 2, pp. 2-44, 2011.

[5] D. Taylor, T. Hallett, P. Lowe and P. Sanders, "Prime vs. Zoom," in Fotografia Digital, Lisboa, DK Publishing, 2016, pp. 124-125.

[6] "Nikon - DSLR Camera Basics," Nikon Corporation, [Online]. Available: http://imaging.nikon.com/lineup/dslr/basics/19/01.htm. [Accessed 4 January 2016].

[7] D. Taylor, T. Hallett, P. Lowe and P. Sanders, "Controlar a exposição," in Fotografia Digital, Lisboa, DK Publishing, 2016, pp. 76-77.

[8] "Nikon USA - Aperture," Nikon Inc., [Online]. Available: http://www.nikonusa.com/en/learn-and-explore/article/g3cu6o1r/understanding-maximum-aperture.html. [Accessed 4 January 2016].

[9] D. Taylor, T. Hallett, P. Lowe e P. Sanders, “Profundidade de Campo,” em Fotografia Digital, Lisboa, DK Publishing, 2016, pp. 108-109.

[10] "Cambridge in Colour - Depth of Field," Cambridge in Colour, [Online]. Available: http://www.cambridgeincolour.com/tutorials/depth-of-field.htm. [Accessed 6 January 2016].

[11] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," Stanford University Computer Science Tech Report CSTR 2005-02, California, 2005.

[12] G. Lippmann, "Epreuves Reversibles Donnant la Sensation du Relief," *Journal de Physique Théorique et Appliquée,* vol. 7, no. 1, p. 821–825, 1908.

[13] A. Lumsdaine and T. Georgiev, "The Focused Plenoptic Camera," *IEEE International Conference on Computational Photography (ICCP),* vol. 1, pp. 1-8, 2009.

[14] "Lytro Website," Lytro, Inc., [Online]. Available: https://www.lytro.com/. [Accessed 7 January 2016].

[15] "Raytrix Website," Raytrix GmbH, [Online]. Available: http://www.raytrix.de/. [Accessed 2016 January 7].

[16] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz and M. Levoy, "High Performance Imaging Using Large Camera Arrays," ACM Transactions on Graphics, vol. 24, no. 3, pp. 765-776, 2005.

[17] J. Lino, "2D Image Rendering for 3D Holoscopic Content using Disparity-Assisted Patch Blending," (Master’s thesis, Instituto Superior Técnico, Lisbon, Portugal). Retrieved from

https://fenix.tecnico.ulisboa.pt/downloadFile/395145926409/Disserta%C3%A7%C3%A3o.pdf, 2013.

[18] C. Conti, L. D. Soares and P. Nunes, "Light Field Coding With Field-of-View Scalability and Exemplar-Based Interlayer Prediction," IEEE TRANSACTIONS ON MULTIMEDIA, vol. 20, no. 11, pp. 2905 - 2920, 2018.

[19] C. Conti, L. D. Soares and P. Nunes, "Light field image coding with jointly estimated self-similarity bi-prediction," Signal Processing: Image Communication, vol. 60, pp. 144-159, 2018.

[20] T. G. Georgiev and A. Lumsdaine, "Focused Plenoptic Camera and Rendering," *Journal Of Electronic Imaging,* vol. 19, no. 2, pp. 1-11, 2010.

[21] T. Georgiev and A. Lumsdaine, "Superresolution with Plenoptic 2.0 Cameras," in *Frontiers in Optics 2009/Laser Science XXV/Fall 2009 OSA Optics & Photonics Technical Digest, OSA Technical Digest (CD) (Optical Society of America, 2009), paper STuA6,* San Jose, 2009.

[22] T. Georgiev, A. Lumsdaine, J. Yu and Z. Yu, "An analysis of color demosaicing in plenoptic cameras," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Washington, 2012.

[23] "Java SE Development Kit," Oracle, [Online]. Available: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. [Accessed 28 September 2015].

[24] “OpenCV,” OpenCV Team, [Online]. Available: https://opencv.org/. [Accessed 28 September 2015].

[25] “eclipse IDE,” Eclipse Foundation, [Online]. Available: https://www.eclipse.org/ide/. [Acedido em 28 September 2015].

[26] "Recommendation ITU-T P.910: Subjective video quality assessment methods for multimedia applications," 2008.

[27] "Recommendation ITU-R BT.500: Methodology for the subjective assessment of the quality of television pictures," 2012.

[28] S. Opozdaa and A. Sochana, "The survey of subjective and objective methods for quality assessment of 2D and 3D images," Theoretical and Applied Informatics, vol. 26, no. 1-2, pp. 39-67, 2014.

[29] "Todor Georgiev - Adobe," [Online]. Available: http://tgeorgiev.net/. [Accessed 28 September 2015].

[30] J.-S. Lee, F. D. Simone and T. Ebrahimi, "Subjective Quality Evaluation via Paired Comparison: Application to Scalable Video Coding," IEEE Transactions on Multimedia, vol. 13, no. 5, pp. 882 - 893, 2011.

[31] P. Hanhart, L. Krasula, P. L. Callet and T. Ebrahimi, "How to benchmark objective quality metrics from paired comparison data?," in 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, 2016.

[32] A. Lumsdaine, G. Chunev and T. Georgiev, "Plenoptic Rendering with Interactive Performance Using GPU," in SPIE Electronic Imaging, January 2012.