Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Radio Engineering

Methods for plenoptic image data processing
Metody zpracování plenoptických obrazových dat

Master's thesis

Bc. Jan Švihálek

Master programme: Communications, Multimedia and Electronics
Branch of study: Multimedia Technology
Supervisor: Ing. Karel Fliegel, Ph.D.

Prague, January 2018

Thesis Supervisor:
Ing. Karel Fliegel, Ph.D.
Department of Radio Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
166 27 Prague 6
Czech Republic

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Švihálek Jan** Personal ID number: **392957**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Radioelectronics**

Study program: **Communications, Multimedia, Electronics**

Branch of study: **Multimedia Technology**

II. Master's thesis details

Master's thesis title in English:

Methods for plenoptic image data processing

Master's thesis title in Czech:

Metody zpracování plenoptických obrazových dat

Guidelines:

Review the methods for plenoptic image data processing and focus on the latest image compression methods. Implement the selected methods and verify their performance using suitable test image data.

Bibliography / sources:

[1] Ng, R.: Digital Light Field Photography, PhD dissertation, Stanford, 2006.
[2] Ebrahimi, T., Foessel, S., Pereira, F., Schelkens, P.: JPEG Pleno: Toward an Efficient Representation of Visual Reality, IEEE MultiMedia, vol. 23, no. 4, 2016.

Name and workplace of master's thesis supervisor:

Ing. Karel Fliegel, Ph.D., Department of Radioelectronics, FEL

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **20.09.2017** Deadline for master's thesis submission: **09.01.2018**

Assignment valid until: **15.02.2019**

Ing. Karel Fliegel, Ph.D.	Head of department's signature	prof. Ing. Pavel Ripka, CSc.
Supervisor's signature		Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Declaration

I hereby declare I have written this master thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, January 2018

.....
Bc. Jan Švihálek

Abstract

Until recent years photography was about two-dimensional image; however, recently there is a rise of technologies capturing, processing and reproducing more dimensional visual data. Light field camera is a device, which can sample light field and not only the 2D representation of the scene. This thesis describes plenoptic (light field) camera principles and mainly focuses on processing and compression of light field data.

Current state-of-the-art and ad hoc encoders for light field data compression are described. Objective and subjective quality assessment methods for light field data processing evaluation are discussed.

Implemented interface for light field data compression allows to apply and analyse compression schemes with various pre-processing steps and adjusted compression settings. Performance of implemented compression schemes is evaluated using objective metrics for image quality assessment.

There is no existing standard nor recommendation for light field data compression and quality evaluation. Existing state-of-the-art video codecs with adjusted setting and pre-processed light field data can efficiently compress such data; however, further research still needs to be done to develop standardized compression of multi-dimensional image data.

Keywords: Plenoptic data, light field, image data, compression, Lytro.

Abstrakt

Až donedávna fotografie představovala pouze dvourozměrný snímek, ale v posledních letech dochází k vzestupu technologií, které zachycují, zpracovávají a zobrazují vícerozměrná obrazová data. Plenoptická kamera je zařízení, které umožňuje vzorkovat světelné pole na rozdíl od klasické 2D reprezentace dané scény. Tato diplomová práce popisuje principy plenoptické kamery (kamery světelného pole) a zaměřuje se především na zpracování a kompresi plenoptických dat.

Práce popoisuje nejmodernější technologie a ad hoc řešení pro kompresi světelných dat. Dále jsou popsány objektivní a subjektivní metriky pro hodnocení kvality algoritmů zpracovávajících světelná data.

Implementované rozhraní pro kompresi plenoptických dat umožňuje předzpracování dat, aplikaci a analýzu kompresních algoritmů s různým nastavením. Výkon implementovaných kompresních algoritmů je zhodnocen použitím objektivních metrik pro hodnocení kvality obrazu.

V současné době neexistuje standard nebo doporučení pro kompresi a hodnocení kvality plenoptických dat. Úpravou stávajících video kodeků lze efektivně komprimovat předzpracovaná světelná data, ale aby byl vyvinut standardizovaný kompresní algoritmus vícedimenzionálních obrazových dat, je třeba aby výzkum pokračoval.

Klíčová slova: Plenoptická data, světelné pole, obrazová data, komprese, Lytro.

List of Tables

7.1 Overview of compression schemes 36

10.1 Performance of different ordering schemes 70
10.2 Comparison of different ordering schemes 71

E.1 Comparison of Lytro's cameras 89

List of Figures

2.1 Working principle of conventional camera with ray-space diagram 3
2.2 Working principle of pinhole camera 4
2.3 4D light field parametrization 5
2.4 Light field in ray-space digram 7

3.1 Single Lens Stereo 10
3.2 Examples of Single Lens Stereo Adapter 11
3.3 Stanford's multi-camera array 12
3.4 Plenoptic camera working principle 12
3.5 Spatio-angular trade-off 14
3.6 Simplified model of plenoptic camera 1.0 and 2.0 14
3.7 Plenoptic 1.0 and 2.0 micro-images comparison 15

4.1 Two generations of Lytro camera's - Lytro (F01) and Illum 18
4.2 Second prototype of Adobe's light field camera 19
4.3 Angular resolution 22
4.4 Working principle of HoloVizio display 23

6.1 Light field data - Raw lenslet structure 30
6.2 Light field data - Sub-aperture structure 30
6.3 Light field data - Epipolar images 31
6.4 Light field data structure 31
6.5 Relative difference of each view to mean view 32
6.6 Example of depth estimation representation 33
6.7 Example of perspective change 33
6.8 Example of digital refocusing 34

7.1 Generalized block diagram of JPEG2000 encoder/decoder 37
7.2 Generalized block diagram of x264 encoder 41
7.3 Sequence of light field data formats 46

9.1 Block diagram of implemented compression tool 55
9.2 Graphical User Interface - compression tool 56
9.3 Graphical User Interface - data input 57
9.4 Graphical User Interface - compression settings panel 59
9.5 Input sequence reordering possibilities (Illum) 61
9.6 Input sequence reordering possibilities (F01, lenslet blocks) 62
9.7 Graphical User Interface - objective metrics 63
9.8 Graphical User Interface - data management panel 64
9.9 Graphical User Interface - interactive subjective comparison 65
9.10 Graphical User Interface - interactive GMS Index Map analysis 65

10.1 Subset of 12 images used for performance analysis 67
10.2 MS-SSIM and GMSD for all compression schemes - Ankylosaurus 68
10.3 MS-SSIM and GMSD for all compression schemes - Friends 68
10.4 MS-SSIM and GMSD for all compression schemes - Gravel Garden 69
10.5 MS-SSIM and GMSD for all compression schemes - Railway 69
10.6 Example of JPEG2000 lenslet compression artifacts - Pillars 70
10.7 x264 - performance of different preset tunes 71
10.8 Performance of x264 encoder with different maximum number of reference frames 72
10.9 Performance of x264 encoder with different maximum number of B-frames 72
10.10 Lenslet image partitioned into blocks 73
10.11 Lenslet image partitioned into pseudo-sequence of blocks with varying size 74
10.12 Lenslet image partitioned into pseudo-sequence of blocks with varying ordering . 74

A.1 Performance of x264 encoder with different AQ strength values 79
A.2 Example of flat and detailed areas after processing with x264 with different AQ strength 80
A.3 Performance of x265 encoder with different number of I-frames in the pseudo-sequence 80
A.4 Performance of x264 encoder with default settings against adjusted settings . . . 80
A.5 Lenslet image partitioned into pseudo-sequence of slices with varying ordering . . 81

List of Acronyms

AQ Adaptive Quantization. 60, 71, 79, 80

AVC Advanced Video Coding. 40, 41, 58

bpp bits per pixel. 43, 46, 69

CB Coding Block. 41

CMOS Complementary Metal-Oxide-Semiconductor. 11, 17

CRF Constant Rate Factor. 58

CTB Coding Tree Block. 41

CTU Coding Tree Unit. 41, 68

CU Coding Unit. 41, 59

DCT Discrete Cosine Transform. 37, 38, 43

DOF Depth of Field. 13

DPB Decoded Picture Buffer. 59

DSCQS Double Stimulus Continuous Quality Scale. 49

DSLR Digital Single-Lens Reflex camera. 19, 77

DWT Discrete Wavelet Transform. 37–40

EBCOT Embedded Block Coding with Optimized Truncation. 38

EPFL École Polytechnique Fédérale De Lausanne. 29, 49, 56, 67

EZW Embedded Zero Tree Wavelet. 40

FOD Field of Depth. 21, 22

FOV Field of View. 21, 22

FR Full Reference. 52

GMS Gradient Magnitude Similarity Deviation. 63–65

GMSD Gradient Magnitude Similarity. 50, 52, 53, 62, 63, 69, 72–74, 79, 80

GOP Group of Pictures. 58, 59

GUI Graphical User Interface. 2, 55, 56, 63, 78

HDR High Dynamic Range. 39

HEVC High Efficiency Video Coding. 41, 42, 44–47, 50, 58

HVS Human Visual System. 4, 9, 38, 39, 51, 52, 70

ICIP International Conference on Image Processing. 36, 47

ICME International Conference on Multimedia and Expo. 29, 35, 36, 45, 46

IEC International Electrotechnical Commission. 37, 39, 41

IEEE Institute of Electrical and Electronics Engineers. 56

IQA Image Quality Assessment. 52

ISO International Organization for Standardization. 37, 39, 41

ITU-T International Telecommunication Union - Telecommunication Standardization Sector. 37, 39, 41

JOD Just Objectionable Difference. 50

JPEG Joint Photographic Experts Group. 26, 35, 37, 39

JSON JavaScript Object Notation. 26

MB Macro Block. 40, 41

MOS Mean Opinion Score. 50

MPEG Moving Picture Experts Group. 41

MS-SSIM Multi-Scale Structural Similarity Index. 52, 62, 63, 69–73, 80

MSE Mean Squared Error. 51, 52

MV Multi View. 47

PB Prediction Block. 41

POV Point of View. 20

PSNR Peak Signal-to-Noise Ratio. 42–46, 49–52, 62, 63, 69, 70, 72, 73, 79, 80

QP Quantization Parameter. 58

R-D Rate-Distortion. 63, 68, 70, 71

ROI Region Of Interest. 38, 39, 52

SNR Signal-to-Noise Ratio. 39

SPIHT Set Partitioning in Hierarchical Trees. 40, 42

SSIM Structural Similarity Index. 42, 43, 46, 49–52, 62, 63, 69, 86

VCEG Video Coding Experts Group. 41

VR Virtual Reality. 18

Contents

Abstract vii

Abstrakt ix

List of Tables xi

List of Figures xiii

List of Acronyms xv

1 Introduction 1

2 Photography and Light Field 3
2.1 Photography . . . 3
2.2 Light Field . . . 4
2.3 Ray-Space Diagram . . . 5

3 Light Field Acquisition 9
3.1 History of Plenoptics . . . 9
3.2 Single Lens Stereo . . . 10
3.3 Camera Array . . . 11
3.4 Plenoptic Camera . . . 11
3.4.1 Plenoptic 1.0 . . . 13
3.4.2 Plenoptic 2.0 . . . 14
3.4.3 Comparison . . . 15

4 Light Field Technology 17
4.1 Capturing Technology . . . 17
4.1.1 Lytro . . . 17
4.1.2 Adobe Systems prototypes . . . 18
4.1.3 Raytrix . . . 19
4.1.4 Smartphone Solutions . . . 20
4.2 Presentation Technology . . . 20
4.2.1 Stereoscopy . . . 20
4.2.2 3D displaying . . . 21
4.2.3 Light Field Display . . . 22
4.2.4 HoloVizio . . . 22

5 Software and Tools 25
5.1 Lytro Desktop . . . 25
5.2 LFPSplitter . . . 26
5.3 Light Field Toolbox . . . 26

6 Light Field Data **29**
6.1 Formats and datasets . . . 29
6.2 Data representation . . . 32
6.2.1 Depth Estimation . . . 32
6.2.2 Change of Perspective . . . 33
6.2.3 Refocusing . . . 34

7 Compression **35**
7.1 State-Of-The-Art Compression Schemes For Conventional Image Data . . . 37
7.1.1 JPEG2000 . . . 37
7.1.2 JPEG XR . . . 39
7.1.3 SPIHT . . . 40
7.1.4 AVC/x264/H.264 . . . 40
7.1.5 HEVC/x265/H.265 . . . 41
7.2 Transform Coding . . . 42
7.3 Predictive Coding . . . 43
7.4 Pseudo-sequence Coding . . . 44
7.4.1 Sub-aperture Images Sequence Coding . . . 44
7.4.2 Data Formats for High Efficiency Coding of Lytro-Illum Light Fields . . . 44
7.4.3 High Efficiency Coding of Light Field Images Based on Tiling and Pseudo-temporal Data Arrangement . . . 45
7.4.4 Pseudo-sequence-based Light Field Image Compression . . . 46
7.4.5 Interpreting Plenoptic Images as Multi-views Sequences For Improved Compression . . . 47

8 Light Field Image Quality Evaluation **49**
8.1 Subjective assessment . . . 49
8.2 Objective assessment . . . 50
8.2.1 Peak Signal-to-Noise Ratio and Mean Squared Error . . . 51
8.2.2 Structural Similarity Index . . . 51
8.2.3 Multi-Scale Structural Similarity Index . . . 52
8.2.4 Gradient Magnitude Similarity Deviation . . . 52

9 Compression Tool **55**
9.1 Overview . . . 55
9.1.1 Input Data . . . 56
9.1.2 Compression Possibilities . . . 57
9.1.3 Objective Metrics . . . 62
9.1.4 Data Management . . . 63

10 Performance Analysis **67**
10.1 Pseudo-sequences . . . 69
10.1.1 Sub-aperture views sequence . . . 70
10.1.2 Lenslet block sequence . . . 73

11 Conclusion **77**

Appendix A Additional charts and examples archive **79**

Appendix B Structure of folders in appendix archive **83**

Appendix C Overview of functions in Matlab implementation **85**

Appendix D Implementation README file **87**

Appendix E Comparison of Lytro F01 and Illum cameras **89**

Bibliography **96**

Chapter 1

Introduction

In today's world, the imaging technology moves forward faster than ever. Imaging technology industry grows rapidly as there is an increasing demand for capturing the moments in more realistic ways. From the very beginning till now, the progress in photography was being made by filling up the missing information in captured data. Analog black and white photography was replaced by colour photography and now the possibility of taking tens of frames in one second adds the time information about the captured scene. In traditional photography, light passes through the optical lens of the camera onto the imaging sensor. Each pixel on imaging sensor represents an angular integration of incident light at this position. Three pixels with three different adjacent colour filters give the colour information at this position; however, with this one image the depth information of the scene cannot be measured. What is obtained is just a 2D representation of the 3D scene and our brain and experience are what makes us able to determine the distance between the objects in this 2D representation.

Light field camera can be seen as another step which allows to record the scene information enriched of angular dimension. One of the possible implementation is placing an array of microscopic lenses in front of the imaging sensor, light rays are demultiplexed and fall to different cells on the imaging sensor. This way certain angular information is preserved in terms of different viewpoints even though only one scene in one exposition is captured. Light field capturing technology offers new capabilities in fields such as depth estimation, post-refocus, segmentation etc. This dimension-enriched data are also more demanding on data processing, compression and representation.

This thesis first brings a quick explanation of light field and light field data acquisition in chapters 2 and 3. The main focus is on the plenoptic (light field) camera, which allows to record angular information of the scene. In the fourth chapter current market light field data technology is summarized. The fifth chapter brings an overview of available software tools for Lytro cameras. Chapter 6 analyses Lytro photography and shows different ways how light field data can be represented, used and processed and explains needed terminology for next chapters.

Compression of light field data captured by a plenoptic camera, is summarized in the chapter seven, where several state-of-the-art compression algorithms and novel ad hoc solutions are described. Next chapter tells about objective and subjective quality assessment of digital image and light field data.

Chapter 9 is dedicated to the practical implementation of this thesis, where Graphical User Interface (GUI) of implemented compression tool is explained along with all its functions and possibilities. Performance analysis of used compression schemes is described in chapter 10.

Chapter 2

Photography and Light Field

Physical properties in photography can be interpreted by simple approaches as geometrical optics (image formation), diffraction (resolution), polarization and photoelectric effect which is based on more complex theories [1]. This chapter deals with the fundamentals needed in order to understand light field data acquisition, representation and light field data processing.

2.1 Photography

Photography is an act of recording light as electromagnetic waves into an image which can be preserved possibly forever. In analog photography, this image is created by chemical reaction on light-sensitive photographic film. Now, in the digital era, photography is composed of pixel intensities measured with image sensor. This light-sensitive sensor collects all the individual light rays passing through the lens optical system. One pixel in the final image is formed by the sum of all the light rays that converges at this point. Nevertheless, with this camera design most of the information about light entering the lens of the camera is lost.

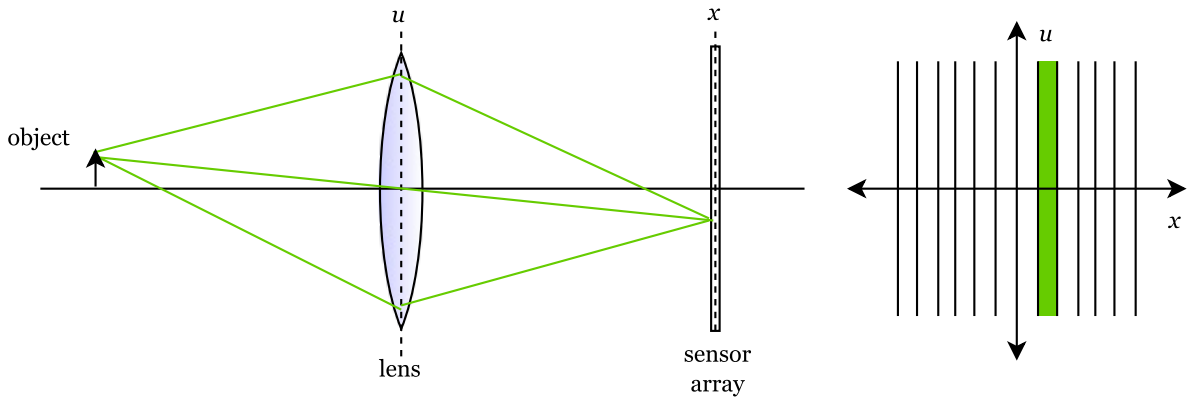

Figure 2.1: Simplified diagram depicting how conventional camera works (left) and ray-space diagram (right).

From Figure 2.1, which depicts the principle of scene recording with a single lens system, it is clear that the pixel from image sensor records only the summation of all the incident rays at

this point and therefore the information about the incident light rays angle is irretrievably lost. This is also the crucial fact behind focus problem of conventional photography. This means that only one focal plane can be in focus with respect to the rest of the photo.

2.2 Light Field

Light is an electromagnetic radiation within a narrow range of wavelengths from approximately 400 to 800 nm. These wavelength borders are determined by the atmosphere and maximum solar emission [1]. Here and in photography, light usually refers to visible light, that means the light which can be perceived by Human Visual System (HVS). Light can be described either as a wave or as a particle. In the former case, it is described as a superposition of monochromatic plane-waves characterized by their frequency, phase and direction of propagation [1]. In the latter case, it is described as a photon, of a certain frequency, which carries very low energy [1]. Particular light colour is made by one or by a mixture of several wavelengths (spectral composition) and recording the true colour, is one of the challenges in digital photography.

Described light can be emitted by any light source, for example, sun. Sunlight (directional rays) coming from the sun, our main light source, is filtered as it comes through the atmosphere [1]. Filtered sunlight, which as a mixture of the spectral components, appears as white light. Sunlight is partially absorbed and reflected by an object in the scene of our interest. These reflected rays of certain energy, frequency composition and direction is what human visual system measure. Light rays travel in every direction in space to create continuous light field [2].

Plenoptic function was formulated in [3], and is frequently used in literature to fully describe the complex information about the light filling the space area. A seven-dimensional plenoptic function can be fully described with an example of elementary pinhole camera P as can be seen in Figure 2.2.

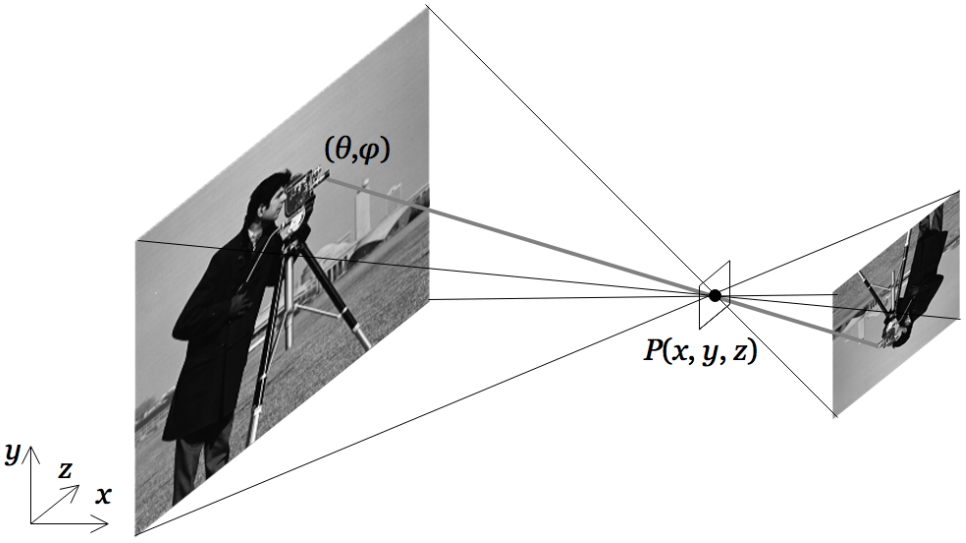

Figure 2.2: Pinhole diagram showing capturing the scene.

The intensity distribution of light coming through the pinhole can be described by spherical coordinates as direction (θ, ϕ) and its colour can be described by wavelength λ [4]. Now let's imagine that the pinhole camera is moved to every position possible in 3D space to coordinates x, y, z, to get every possible view of the scene. The seventh dimension is a time t considering that the scene is dynamic. Situation depicted in Figure 2.2 simplifies plenoptic function into 5D plenoptic function, because wavelength λ and time t are missing. Seven-dimensional plenoptic function $L(\theta, \phi, \lambda, x, y, z, t)$ fully characterizes the scene and this characterization is in practice significantly sampled, quantized and more especially reduced of dimensions. For example, by taking a photo with the traditional camera 7D, plenoptic function is reduced into a flat 2D array of intensities. Another light describing the function, called lumigraph, was introduced later in [5]. The four-dimensional lumigraph function is a subset of more complex plenoptic function. In the field of photography and computer graphics lumigraph function is usually mentioned under the term light field, which in computer graphics is explained as a set of all light rays in space [2]. Levoy *et al.* [6] explained that any ray from the light field can be described by intersections of two 2D planes as two points (u, v) and (x, y) because light rays remain constant in free transparent space. Such simplification of plenoptic function into 4D light field is shown in Figure 2.3

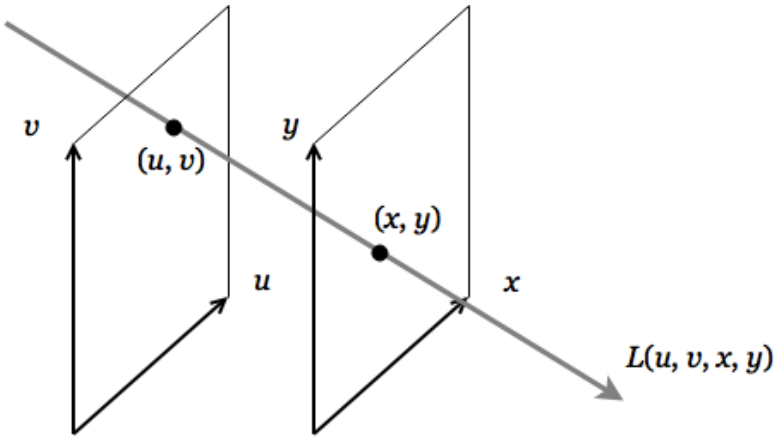

Figure 2.3: Light field parametrized by two points as intersection of two 2D planes.

Later Ren Ng in [7] used light field function $L(u, v, x, y)$ to describe the working principle of his light field camera. First 2D plane (u, v) is at camera plane and the second plane (x, y) is at focal plane [6]. Technology and methods are capable to obtain the additional information about the angle of incident light rays are described in 3.

2.3 Ray-Space Diagram

The term ray-space diagram (or Cartesian ray-space diagram) used in [3], needs to be introduced for further explanation and easier visualization of light field data acquisition. Ray-space diagram will be probably better explained first on conventional camera as shown in Figure 2.1 (right). In this figure, two previously mentioned planes are reduced in dimension, therefore rays are

now described as $L(u, x)$, instead of light field previously described as $L(u, v, x, y)$. Let's use the notation of light field used in [2] and let's call u as a directional axis because it determines the direction from which rays fall on the image sensor. And let's x be noted as a spatial axis, because it holds spatial information. Now any ray (left part of Figure 2.1) can be represented as a point with coordinates (u, x) in ray-space diagram (right part of Figure 2.1).
As was mentioned before, the classic camera does not record any information about the direction of incoming rays as it sums energy from all rays falling onto sensor cells. In Figure 2.1 it can be seen that all green rays are increments in summation that takes place at one sensor cell of image sensor array. In the ray-space diagram, this is depicted as one (green) vertical bar, because rays coming from all possible directions u (position on the lens) share the same convergence point x (sensitive cell in image sensor array). Each bar in the ray-space diagram can be seen as one sensitive cell in image sensor array which integrates all incident rays at this point. Previously mentioned only apply if x plane lies in image sensor plane (or convergence point). Vertical bar with zero slope can change into non zero positive or negative slope if convergence point of the light rays moves in front or behind x plane respectively [2]. Note that in this case parametrization plane x stays in position and only convergence point is moving. Moving with convergence point of rays is what is called focusing (changing the distance between an image sensor and camera lens). By moving the convergence point further from the camera main lens, the object which is closer to the camera is getting into focus (focal plane moves closer to the camera). Figure 2.4 shows conversion to ray-space diagram for two different settings - in top image parametrization plane x is in image sensor plane which corresponds to vertical bar in ray-space. In the bottom part of Figure 2.4 sensor plane is moved further away from the camera main lens, therefore the positive slope of vertical bar in the ray-space diagram [2].

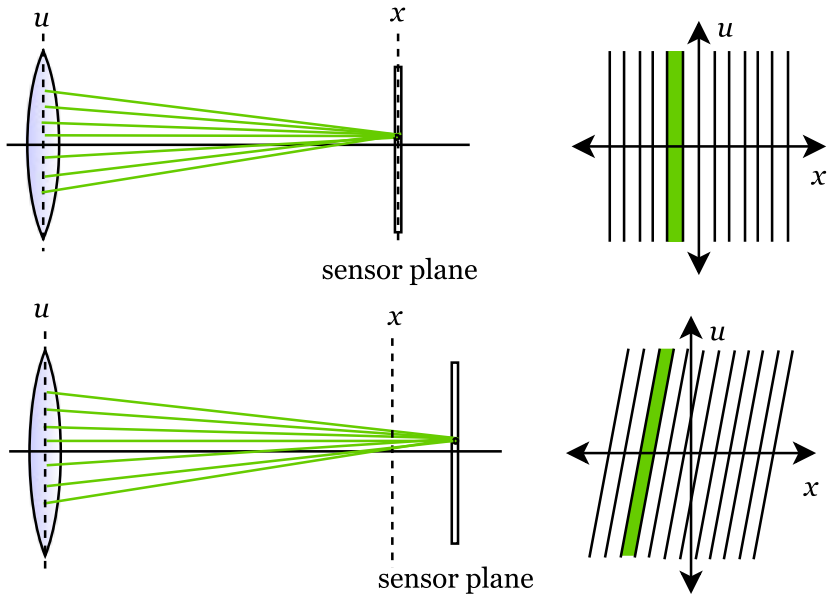

Figure 2.4: Light field depicted in ray-space diagram for two different situations. Figure shows how slopes in the ray-space depend on distance between convergence point and parametrization plane x.

Chapter 3

Light Field Acquisition

HVS allows us to distinguish the depth and distance of objects in the observed scene, because of the brain processing the stereo information coming from our eyes (represents two different viewpoints) [1]. This side-by-side offset gives us a stereoscopic view of the scene and our brain is capable to recognize how far or how near the object is. One eye captures the light which comes through the lens and falls on a retina. Depth information is created by combining sensations from both eyes. The traditional camera can be very roughly compared to one eye of HVS as it records the scene from only one viewpoint. The classic camera captures light intensity and its colour, however, angular information is lost as the light-sensitive cell integrates all the incidental rays. Brain (experience with real-life scenes) is what enables us to recognize the depth and estimate distance in a 2D image [1]. In order to sample angular information, rays from different directions need to be mapped to corresponding sensitive positions/areas (different cameras/light-sensitive cells on the sensor). First and fundamental method to capture the light field is to create an array of synchronized cameras, where each camera will record the scene from the slightly different angle and thus will partially contribute to capturing the light field. Synchronization, precise measurements and settings are crucial for this system to work properly.

3.1 History of Plenoptics

Barrier method can be considered as one of the first auto-stereoscopic methods (auto, in this case, means, that observer does not need any additional optical equipment). The first application was proposed and demonstrated in 1692 by French painter G.A. Bois-Clair, who used the so-called parallax-barrier technique to create a 3D sensation to viewer [8]. This method uses at least two images, sliced into stripes, which are aligned behind opaque bars in the same frequency. Back in 1692, paintings changed as a viewer was walking around them. This method was later applied in photography by Frederic Eugene Ives, who in 1903 patented parallax stereogram [9]. His patent used vertical plates to control which part of the image can be viewed by which eye to create a stereoscopic sensation. Ives came up with the technique which was and still is widely exploited in various applications as 3D postcards, trade cards, etc.

First principles of capturing light field in photography are dating back to the year 1908, when Gabriel M. Lippmann introduced the spatially multiplexed technique of light field capturing [10] by using an array of lenses (called integral photography). The idea was to use an array of small spherical lenses on top of the picture instead of vertical stripes. Proposed technique can be used both for recording or displaying the image. This modification allows creating stereo sensation not only in horizontal direction. Many others developed his idea further during following decades - Sokolov (1911), Coffey (1935), Ivanov (1948) and Chutjian (1968) with first digital light field recording device [2]. Some methods of light field data acquisition are described in following sections of this chapter.

3.2 Single Lens Stereo

First and the most straightforward method of capturing spatial information is by capturing the scene from two nearby viewpoints, therefore, acquiring two images which are shifted horizontally or vertically from each other [4]. For simplification let's assume the camera with the main lens and an eccentric aperture as shown in Figure 3.1. If the subject to be captured is in focus plane, its image is focused, but it is composed of only half of the all possible light rays [4]. Then if the object is out of focus plane, its image is still on image sensor plane, but now its shifted, because the aperture transmits only the rays coming through the right part of the main lens as shown in Figure 3.1.

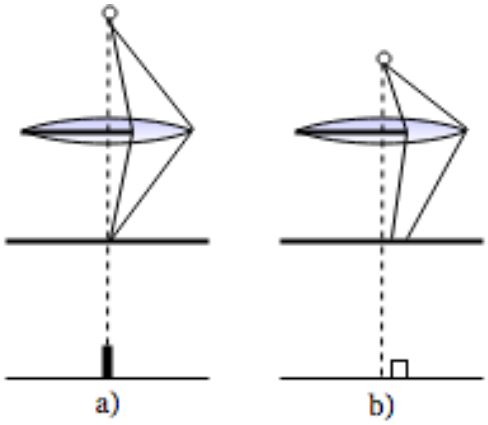

Figure 3.1: Single lens stereo - a) object is located in focus plane and its image is focused, but it is composed by only half of the light rays, b) object is out of focus plane and its image is displaced [4].

If the eccentric aperture is positioned in the left or right part, the image projected onto sensor plane is located on the right or left respectively. This rule obeys only if the captured object is located between the plane of focus and camera aperture plane. If the captured object is located behind focus plane, the image projected on the sensor plane is on the left or right if the location of the eccentric aperture is on the left or right respectively [4]. The rate and direction of image shift from the centre of sensor plane allow one to estimate the distance of the object [4].

Figure 3.2: Pentax's (left) and Samsung's (right) stereo adapter[1].

In practice, so-called stereo adapter can be encountered, which is placed before normal lens as shown in Figure 3.2. Lens adapter consists of mirrors, which are used to direct the rays coming through two separate holes on to the light-sensitive image sensor. This method creates side-by-side images, which are typical for stereo image content and thus can be later easily displayed using stereo displays.

3.3 Camera Array

Another technique, which is also principally simple and widely used to capture light field, is to use an array of conventional cameras. One of the most mentioned applications of the camera array in literature is Stanford's large camera array [11]. The idea exploits the fact that image sensors are getting cheaper and also the possibility to perform computational photography will be cheaper and easier in future. With that in mind, an array of 100 cameras with Complementary Metal-Oxide-Semiconductor (CMOS) sensors was constructed in order to perform various imaging tasks [11]. The output of this array can be seen as an array of different images taken from slightly different positions, which are moved in horizontal and/or vertical parallax from each other. All the image data can be visualized as light field representation in 2D images after some data processing (computational photography).

3.4 Plenoptic Camera

More elegant way to capture light field with only one exposure at a time is by using a plenoptic camera. Word plenoptic comes from the composition of two words. First part "plen" - comes from the Latin plenus and it means full and the second part, "optic" (optics) is self-explanatory. Plenoptic camera (or another term used in literature is light field camera) is a device that captures part of the optical structure of the light by measuring how does the scene look from all possible perspectives at the position of the camera's main lens [4]. There are slightly different

[1]Source: Ars Technica - The old school tech Samsung used to achieve single lens 3D,
URL: `https://arstechnica.com/gadgets/2013/01/the-old-school-tech-samsung-used-to-achieve-single-lens-3d/`.
Used 22/03/2017.

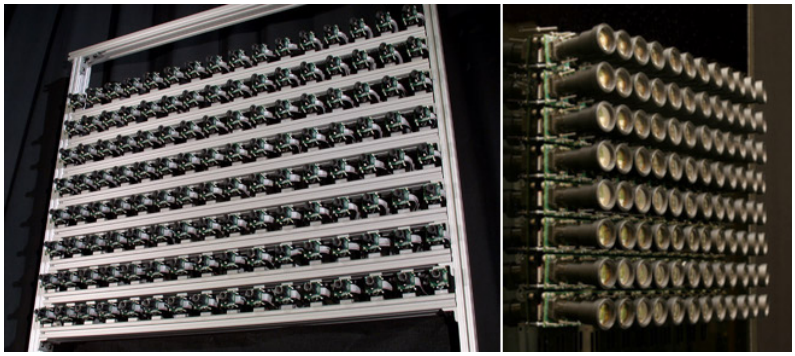

Figure 3.3: Stanford's multi-camera array [11].

designs of plenoptic cameras for which usually the terms plenoptic 1.0 and 2.0 are being used in literature and their differences will be described later in this section.
Design of light field camera is similar to a conventional camera as the main difference is made by placing an array of micro-lenses between the main camera lens and the light-sensitive image sensor. An inserted array of micro-lenses is the key part, which enables the camera to get the angular information of incident light rays [2]. This camera design provides required information on how the captured scene looks from a certain area of potential angles defined by the camera main lens [4]. Figure 3.4 depicts how micro-lens array is used to preserve information about angle thanks to the additional separation of converging light rays. Each micro-lens in the lens array covers a small array of light-sensitive image points (cells) from the whole image sensor. Area of sensitive cells under each micro-lens records focused image of the main lens aperture [7].

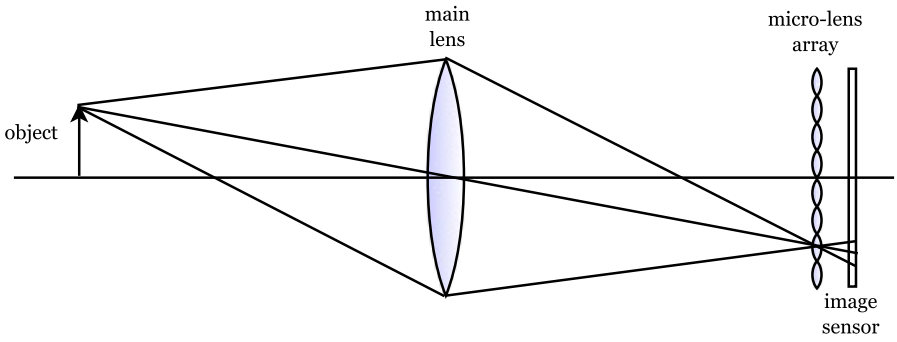

Figure 3.4: Simplified diagram showing working principle of plenoptic camera.

The image sensor thus preserves the sub-images of the main lenses aperture captured by each micro-lens. This fact, together with additional image data processing, allows revealing the depth or distance of objects in the scene [2]. With that in mind, it is clear that light field camera of this design needs to use an image sensor with resolution as high as possible for the purpose of dense light rays sampling (angular resolution) while achieving high resolution of the final image (one viewpoint) [2]. Light field camera provides the flexibility to produce photographs focused at different focal depths. This is made by suitable choice/processing of the sub-images thanks to the design of the camera. Ray-tracing technology is used in order to obtain the final image of the recorded light field [2]. Basically, it is about creating the synthetic camera, which

is configured accordingly to the user needs and then monitoring light rays to the image sensor plane. The desired image is created by adding up all the light rays in the imaginary image plane [2]. This technique allows to remove undesired blur (only the blur caused by moving from focal plane) through manipulation with the convergence of light rays. Light field camera thus allows users to first capture the scene and then focus, which is a significant difference compared to conventional cameras. It is clear that the light field camera would not be immune to images blurred thanks to the movement of the object in the scene or to the blur caused by movement of the camera itself. The described technique also solves the problem of traditional cameras, where the Depth of Field (DOF) is determined by the size of the aperture since each pixel is focused independently by the synthetic composition [2]. Another effect of this construction is also the possibility to create a number of images, seen from slightly different positions of the observer, in a single exposure taken from one position. With a conventional camera, one would have to take a series of pictures with a step that would correspond to a shift in image sensor plane directions. Among other useful utilization of this design is the fact that single exposure is enough to measure the horizontal and vertical parallax corresponding to the imaginary shifts (movements) and thus get estimates for depth measurements of the objects in the captured scene [4].

3.4.1 Plenoptic 1.0

Plenoptic camera 1.0 is based on Lippmann's approach of integral photography [10] which was later developed in [4]. In this design, the micro-lens array with lenses focused at infinity is placed in the focal plane of the main camera lens and exactly one focal length from the light-sensitive sensor [12]. Each micro-lens, instead of integrating all the incident rays, split the rays and directs them onto sensor area under the particular micro-lens. The plenoptic camera 1.0, as implemented by Ren Ng in [2], samples a set of light rays at a single point in space. Image data from the light-sensitive sensor is represented in a 2D array of 2D arrays with sampled radiance, where the position is sampled by micro-lenses and direction is sampled by cells.

Each micro-lens image is defocused with respect to the image created by the main lens and only one pixel from each micro-lens is used to create one final image. This brings up the main drawback of plenoptic 1.0, which is the low resolution of the final image. For angular sampling information in 30×30 array, 900 pixels needs to be reserved at image sensor (from which only 1 pixel is used for the final image composition)[13]. It is clear, that relatively large number of sensor cells must be assigned to each microlens in order to achieve sufficient angular resolution, which results in poor spatial resolution of the final image (resolution of the final image is equal to the dimensions of microlens array). This trade-off between spatial and angular resolution is shown in Figure 3.5.

It can be seen that spatial-angular trade-off is defined by the number of lenses in the microlens array. A large number of pixels under micro-leas means large angular resolution, but with a fixed

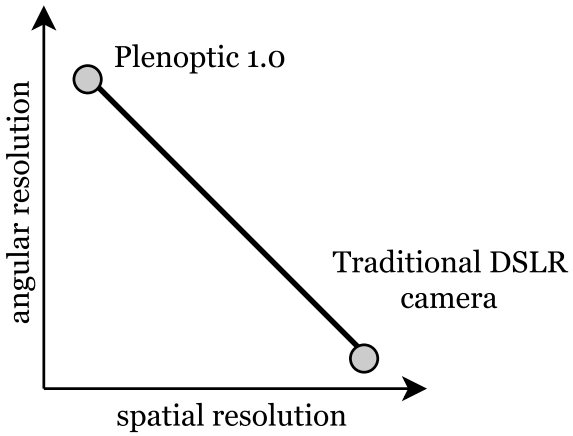

Figure 3.5: Spatio-angular trade-off - constant resolution of image sensor results in inverse relation between spatial and angular resolution which can be achieved.

number of sensor sensitive cells, it also means low spatial resolution. A high number of small micro-lenses is needed in order to achieve high spatial resolution. By increasing the number of micro-lenses thus decreasing the number of pixels under each micro-lens image is getting to the limit, where noisy results thanks to edge effects of each lens are produced [12].

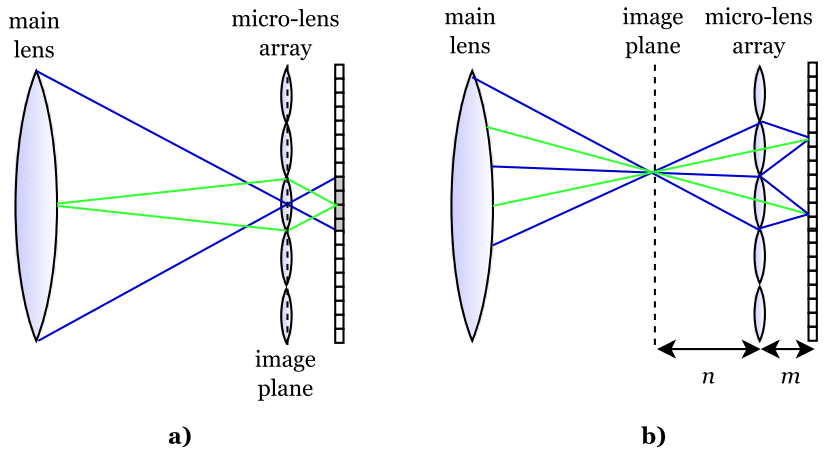

Figure 3.6: Simplified model of plenoptic camera shows how micro-lens array directs rays onto image sensor. a) depicts plenoptic camera 1.0 with micro-lens array placed in image plane and b) depicts plenoptic camera 2.0 with micro-lens array placed in distance n behind image plane.

3.4.2 Plenoptic 2.0

Plenoptic 2.0 design, in literature also often called "Focused Plenoptic" design, is another approach to sampling light field. Lumsdaine and Georgiev in [12] solved the major issue of plenoptic 1.0 by reducing the angular resolution in order to gain more spatial resolution. The principal main difference is in the relative position of the micro-lens array to the main lens and sensor. In this design, the micro-lens array is no longer located at the focal plane of the main lens, but now it is positioned at distance m from the image sensor. In this distance, micro-lenses are focused

on the image plane of the main lens [12]. Now each micro-lens acts as an individual pinhole camera, which sees only a fraction of a virtual image in the camera. Function of plenoptic 2.0 is shown in Figure 3.6. As can be seen, plenoptic 2.0 is different from plenoptic 1.0 by the fact that micro-lens is placed at distance m from the sensor and it is focused on the image plane of the main lens at distance n [14]. Relay system with main lens and designed distances n and m with focal length f follows the thin lens equation [12]

$$\frac{1}{f} = \frac{1}{n} + \frac{1}{m}. \tag{3.1}$$

Spatial resolution can be modified by moving the micro-lens array with respect to the image sensor. Modifying m/n ratio gives the option to choose a position of the trade-off point between spatial and angular resolution. Resolution is now decoupled from the number of microlenses as with this approach final image is composed by multiple pixels per micro-lens instead of one per micro-lens as in plenoptic 1.0 design [12]. Even though this approach reduces angular resolution, the fact of increased spatial resolution could satisfy some of the modern-day photographers.

3.4.3 Comparison

In plenoptic camera 1.0 micro-lens array is placed at one focal length from the image sensor while being focused at infinity (defocused with respect to the main lens). On the other hand in plenoptic 2.0, the micro-lens array is placed in distance m from the image sensor and distance n from the image plane of main lens image while being focused on the image created by the main lens [13]. In different words, a small portion of the image projected by the main lens is transferred onto pixel array under micro-lens with much higher spatial resolution. This means that images under micro-lenses of plenoptic 2.0 are sharp and inverted. And that images under micro-lenses of plenoptic 1.0 design appear blurry as they only show one point viewed from different angles. Difference between those two approaches can be seen better in Figure 3.7.

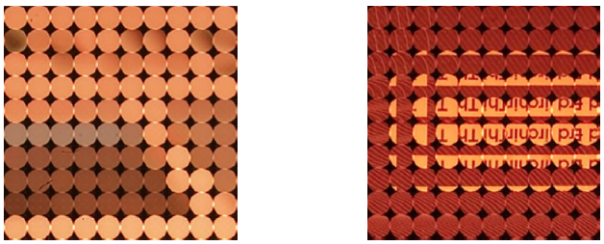

Figure 3.7: Comparison of micro-images in plenoptic 1.0 (left) and in plenoptic 2.0 (right)[2].

Raw sensor data of plenoptic 2.0 approach can be imagined as an array of sharp images of main lens image plane. While raw sensor data of plenoptic 1.0 can be seen as an array which consists of small arrays (defined by micro-lens) containing angular information for one point of the scene [15]. With that in mind it is clear that with same fixed dimensions of imaging sensor (and

[2] Source: The Focused Plenoptic Camera - slides of Todor Georgiev,
URL: `http://www.tgeorgiev.net/EG10/Focused.pdf`. Used 02/04/2017.

no additional interpolating algorithms), plenoptic 1.0 would achieve higher angular resolution while plenoptic 2.0 would achieve higher spatial resolution. The result difference between the two methods is in the image resolution of the rendered image because plenoptic 1.0 use one pixel per micro-lens while plenoptic 2.0 approach use multiple pixels per micro-lens.

Chapter 4

Light Field Technology

4.1 Capturing Technology

Following subsections briefly summarize current market with plenoptic cameras. Focus is on the main manufacturers and description of their products or prototypes. At the end of this section several smartphone implementations are mentioned.

4.1.1 Lytro

Lytro, Inc., probably the most publicly known manufacturer of consumer-oriented light field cameras, was founded in 2006 by scientist Ren Ng. Founder and former Lytro CEO wrote his dissertation thesis at Standford university with the topic "Digital Light Field Photography" [2], which won Standford University's prize for best thesis in computer science [16]. Ren Ng's dissertation thesis is used as a main source of information for this thesis. In 2015 Lytro expanded in the field of cinema, virtual reality, scientific and industrial applications as they successfully extended usage of the light field across various industries. Lytro had to face a tough challenge by competing against an established industry with much larger companies, where camera requirements and parameters are already set to certain level and the regular consumer is also used to certain requirements/specifications.

In 2012 Lytro, Inc. introduced the very first commercially available light field capturing device of the same name - Lytro [16]. First generation Lytro camera (also referred to as F01) does not resemble a regular camera at first glance thanks to its unusual shape shown in Figure 4.1. Inside it consists of regular parts as a set of lenses, CMOS image sensor, processor (in this case called "Light Field Engine") and also the key part, which is the array of micro-lenses placed in front of the image sensor.

Second generation camera, called Illum, was released two years after the first generation with significantly enhanced parameters. The outside look changed more to regular camera-like appearance. Parameters of both Lytro cameras are listed in Table E.1 in appendix E. One of the main commercially offered advantages is the fact that user can take a photo instantly (camera tuning on and taking photo should take up less than one-second [18]) and take care about fo-

Figure 4.1: Left - Lytro Illum (2*nd* generation) released in 2014, right - Lytro Lytro (1*st* generation), also called F01, released in 2014. Pictures taken from [17] and [18] respectively.

cusing later. Another claim by Lytro is that their camera can take better pictures in low-light situations without using flash as it records entire light field in its range of view [18]. It's capability to reproduce 3D image by taking a single shot with single lens is also another appealing feature to the consumers.

In 2015 Lytro presented Lytro Immerge, claiming that it is the first solution for light field cinematic Virtual Reality (VR). Lytro Immerge allows highly configurable and seamless capturing thanks to camera array in spherical design[1]. Year later Lytro Cinema was introduced, which was the first professional light field capturing system for film and TV production [19]. Lytro Cinema brings a breakthrough for filmmakers with taking the controls and some of the decisions from the scene to post-production and therefore allows to create a number of various shots [19]. Lytro claims in their press release, that Lytro Cinema has the highest resolution video sensor ever designed, with 755 RAW megapixels at up to 300 FPS [19]. Lytro Cinema can shoot up to 16 stop of dynamic range and has wide colour gamut.

In 2016 Lytro's exited from the consumer light field camera business and started to fully focus on developing the light field VR platform[2]. During the time this thesis was being written, Lytro stopped hosting the website where images taken with their cameras were shared[3]. Their website allowed users to upload and share their photos within the fully functional interface which allowed refocusing, 3D depth representation etc.

4.1.2 Adobe Systems prototypes

Adobe Systems Inc. is among other companies which are exploring the possibilities of light field data and is developing light field capturing devices. Adobe Systems is most widely known as a specialist for multimedia manipulation and processing software and for several years there have been few papers about light field camera and their prototypes.

[1]Source: Lytro - Press Release - Lytro Immerge,
URL: `https://goo.gl/osN7aB`. Cited 03/04/2017.

[2]Source: Digital Photography Review - Lytro CEO confirms exit from consumer photography business, focus on VR,
URL: `https://goo.gl/GVhDff`. Cited 02/01/2018.

[3]Source: The Verge - article
URL: `https://goo.gl/TdqC5t`. Cited 02/01/2018.

Between years 2004 and 2006 Adobe developed their first prototype of light field camera, which was publicly presented in 2007 [20]. The prototype used 100 megapixel sensor and hexagonal lens array made of 19 small lenses corresponding to 19 different focal points [21]. That means each sub-image was formed by approximately 5.2 megapixels. Added value to their presentation was the fact of cooperation between their prototype and software tool Adobe Lightroom, where they showed the possibility of so-called "focus-brush" or "defocus-brush". This tool would allow a user to easily focus or defocus certain area of a taken image in post-processing with using Adobe-like brush tool [20].

Second prototype exploits two ordinary lenses (positive) and rectangular array of 20 negative lenses (4×5 array) [20] - shown in Figure 4.2. This array design, mounted on top of the standard lens, showed improvement in terms of lost pixels [21].

The third generation of Adobe's prototype camera was presented at NVidia's GPU Conference 2010 [20]. The prototype consisted of Contax 645 camera and micro-lens array which was placed between main lens and image sensor [20]. With the third prototype, Adobe also showed an improvement with real-time software allowing for re-focusing in software. Adobe, which is going in a different direction than Lytro, is developing their light field lens which will be compatible with traditional Digital Single-Lens Reflex camera (DSLR) cameras and along with that, they are working on software to process this light field data [20].

Figure 4.2: Second prototype of Adobe's light field camera (left) and set of 20 images obtained by this camera [21].

4.1.3 Raytrix

Raytrix[4], based and founded in 2008 in Germany, company which also provides light field cameras with specialization in professional industrial and research applications. Their cameras are for example used in observations and control of fluid mechanism, volumetric velocimetry, optical inspection, plant analysis, microscopy, robotics etc. Raytrix's solutions are not aimed at a regular customer as they are highly specialized on individual industrial applications.

[4]Raytrix - URL: `http://www.raytrix.de`.

4.1.4 Smartphone Solutions

Thanks to the constantly increasing popularity and sophisticated optics of smartphone cameras, some manufacturers are making the first steps of integrating light field technology into the mobile phones. There is already a list of companies which are pioneering with light field technology among the smartphones.
Pelican Imaging (acquired by Tessera in 2016[5]), has been researching since 2006 and in 2013 presented their low-cost miniature (3 mm height) camera array[6]. Their device is capturing 16 images (4×4 array) of 16.7 megapixels, which is then processed by their developed software into one 8 megapixel image in JPEG file format. Also, some other companies like Toshiba Semiconductors or LinX Imaging (acquired by Apple[7]) have presented their light field technology solutions for smartphones. However, lately there was not so many implementations of light field technologies into smartphones as this industry is massive and it will take some time for this "new" technology to settle down.

4.2 Presentation Technology

Once light field is captured and processed it needs to be presented as well. The goal of light field presentation device is to provide a faithful representation of real or synthetic scenes i.e. life-like view. The true light field displaying would have to be the real reconstruction of light field as was seen from a natural view. Working principle of light field displays is based on a reversed principle of light field cameras. In cameras, the light field is described with respect to a surface-image sensor (described by intersection with the sensor and angle). Light field display operates on direction selective light emission, which means that light emitting surface enables to emit different light beams from a point in the desired manner [22]. Light field visualization technologies need to work without the use of stereoscopic headsets/glasses or some head tracking devices to provide a full experience of light field visualization for multiple viewers. Most of the commercially used devices, as 2D screens with special headset/glasses or screens with lenticular lenses, are basically using the brain for the calculations as these devices provide very limited number of Point of View (POV). Another problem is the fact that some of these technologies produce nausea, eye tiredness etc.

4.2.1 Stereoscopy

Probably the most known and exploited kind of system for displaying 3D content is based on stereoscopy. There is a variety of implementations which all use the same clues to create a 3D sensation to viewer. Stereoscopy is using the lateral distance between our eyes and different

[5]Source: BusinessWire - Tessera Technologies Acquires Technology Assets From Pelican Imaging Corporation, URL: `http://goo.gl/WxohGZ`. Cited 03/04/2017.
[6]Source: LightField Forum - Pelican Imaging Array Camera, URL: `http://goo.gl/1qCzNN`. Cited 03/04/2017.
[7]Source: TechCrunch - Apple Buys LinX, A Camera Module Maker Promising DSLR-Like Mobile Performance, URL: `https://goo.gl/7ADZ9x`. Cited 03/04/2017.

images are shown for the left and for the right eye of the viewer. The brain recognizes these different images and creates the depth/distance perception [23]. Technologies such as passive polarizers, passive anaglyph (colour filters) or active shutters are being used in stereoscopic displaying together with some sort of glasses (eyewear) to prevent cross-talk between left and right eye. Each of those technologies has its advantages and disadvantages over the other, but in overall stereoscopy has many advantages and is highly exploited especially in the entertainment industry. The technology is easy to implement, less expensive and usually effective (for some applications)[23]. However, there are some major setbacks and needs which are calling for more effective, realistic and less irritating systems to be developed. One of the problems with these systems is the constant need of eye-wear, which may be still acceptable during 2-3 hour film, but probably not for other, longer or more frequent applications. Another drawback is the inability to cope with motion parallax, because as the viewer moves the viewpoint does not change. This is again acceptable in cinemas where viewers are stationary, but may not be acceptable in other applications where the viewer would want to look at objects from different perspective. Motion parallax, could be approached by some head mounted tracking system, but this is only applicable to one viewer only. The major setback still lays in the fact that stereoscopy is not very comfortable thanks to the conflict in the brain caused by showing different image for the different eye. Eye accommodation and vergence conflict is the fact that causes a headache, nausea and motion sickness. More on about why and how are these problems caused while using stereoscopic systems is explained in [23]. Stereoscopic systems are acceptable for short and less frequent usage, but for some other applications there is a need for better technologies to reproduce the spatial content.

4.2.2 3D displaying

Displaying 3D content is one dimension richer when compared to classical 2D displays and therefore 3D displays can be exploited in many practical applications. There are several properties/parameters which are characteristic for spatial reconstruction or 3D displays. Field of View (FOV) is probably the most important one and the ultimate goal is to have the same FOV as to which the viewer is used in 2D displays. The angle which determines the angle of the FOV cone is called emission range. Another important term is independent beam and number of independent beams, which determines angular resolution/Field of Depth (FOD). Some of the mentioned parameters are depicted in the Figure 4.3. Usually, the number of independent beams (angular resolution) is limited in the vertical parallax as the horizontal parallax is more important for a viewer and there are systems which handle horizontal and vertical parallax separately thus having different horizontal and vertical resolution [24]. Combination of those parameters determines the quality of the reproduced scene.

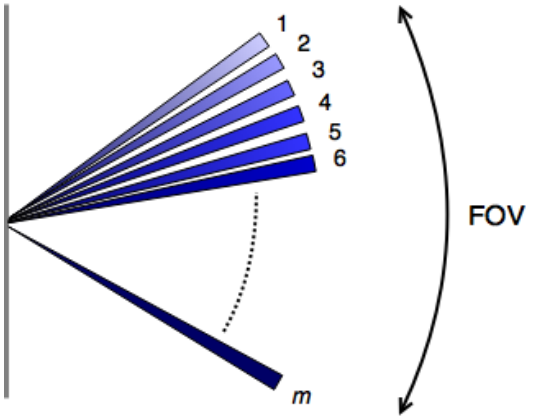

Figure 4.3: Angular resolution = FOV/Independent beams.

4.2.3 Light Field Display

Toru Iwane designed simple 3D light field display [25], by creating a reverse version of light field camera, which simply reconstructs ("decode") 3D volume image near to the lens array from displayed 2D light field data. The technology consist of lenslet array plate, flat display (smartphone-sized display) and simple data processing method to reverse perspective of synthetic or captured scene. Three-dimensional information which is encoded into 2D light field data (capturing part) is then displayed on a flat screen. This 2D data is symmetrically inverted for each microlens and decoded by microlens array of the display. View angle of such display is determined by the parameters of used microlenses [26]. Displaying system presented in [26] gives observer natural perception of reproduced images and without visual contradictions. The presented system does not require any adjustment of the lens array because attaching the lens array to the display panel is enough which was not usually the case with former light field displays.

4.2.4 HoloVizio

Another system displaying multi-dimensional content, called HoloVizio, was presented in [27] and patented by Hungarian company called Holografika. This system produces light beams in optical modules, various light beams hit the points of the screen under various angles of incidence. The position of given point of the screen with respect to optical modules and geometry determines exit angle. Light beams are composed into the continuous view by the holographic screen [24]. Working principle of such display can be shown in Figure 4.4.

High FOV of this system can be achieved by modification of arrangement and angles of optical modules and FOD can be modified by the distance from screen [24]. HoloVizio is one of the truly 3D displaying systems and Holografika is one of the leaders in 3D display development as the company already produced a number of successful systems.

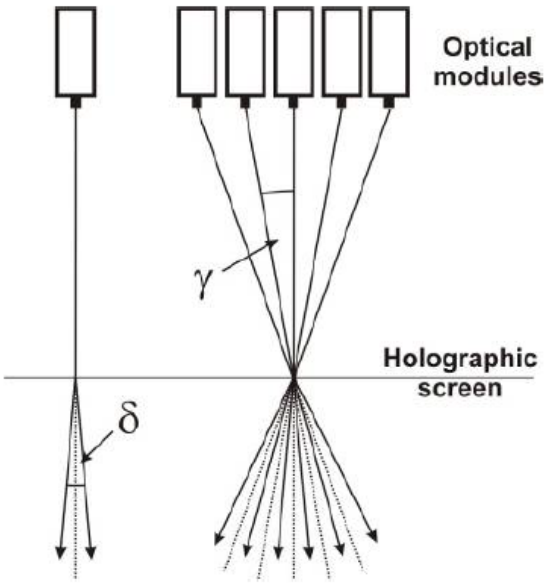

Figure 4.4: Working principle of HoloVizio display [24].

Chapter 5

Software and Tools

Following sections briefly describe tools, which are used for processing and management of light field data captured by Lytro cameras. Currently, there is one official software tool developed by Lytro, and several other open-source tools developed by the community from which two are described here.

5.1 Lytro Desktop

Lytro Inc. provides software called Lytro Desktop to access the files obtained with Lytro cameras. Lytro Desktop enables the user to interactively refocus, shift the perspective and export LFP in early versions and LFR files in latest versions. First generation Lytro camera (F01) produces following files for one image: $IMG\_xxxx.lfp$ and $IMG\_xxxx - stk.lfp$ file (where each x is a digit place-holder). Second generation camera (Illum) produces one more file $IMG\_xxxx - dm.lfp$ (where dm stands for depth map). There are two types of LFP files, one has usually size around 16 MB (for F01 camera) and contains raw Bayer data of $m \times n$ pixel sensor with some metadata containing additional information about captured image and camera itself [28]. The other type of LFP file (usually size of 1-2 MB) is a web-oriented file (obtained from Lytro Desktop software after importing the pictures from the camera) which is used to reduce file size and rendering time for display. This file is basically a set of JPEG files, from which each uniquely represents part of captured light field. The set of files is composed of visually interesting JPEG files, each showing the scene with different focal depth [28].
As Lytro is a consumer product, information about its file format are not available to the public and regular user has to settle down with their software which is available for MacOS and Windows. In overall, Lytro Desktop does not provide much control over light field data (except viewing, refocusing and exporting anaglyph) and other tools are being developed so the light field data can be exploited differently.

5.2 LFPSplitter

Nirav Patel [28] was first to create an open source tool, called LFPSplitter, which was developed to work with LFP and LFR raw Lytro file formats. LFPSplitter is command line controlled tool which can be used to extract plain text metadata, plain text listing of depth look-up table and component Joint Photographic Experts Group (JPEG) files [28].Using LFPSplitter one can obtain a greyscale raw data from the sensor, frame metadata, private metadata (contains camera and sensor serial numbers) and table file (contains array information). Three latter mentioned files are saved in JavaScript Object Notation (JSON) text format. Frame metadata contains information about:

- image resolution and orientation
- pixel format - value of black and white
- pixel packing (endian, bits), mosaic array and upper left pixel
- colour transform array, gamma value, white balance gain, ISO, exposition
- time of image acquisition
- shutter, lens parameters (focal length, f-number ...)
- lens, chip temperature
- micro-lens array parameters (tiling, rotation, scale, ...)
- x, y, z from accelerometer
- firmware, camera type, modes of image acquisition

By extracting data from $IMG\_xxxx - stk.LFP$ (where stk stands for stack) one can obtain stack of rendered JPEG files, each focused on different focal plane, depth look-up table in text format and table file in JSON file format. Depth look-up table contains series of flattened $m \times n$ (20×20 array in first version of Lytro Desktop and 330×330 for later versions) double values of depth at which the image should be refocused if viewer wants to focus corresponding area (first depth value correspond to the top left part and then the values continue in rows). JSON metadata table file contains metadata version, the reference to the look-up table and references to all rendered images with its corresponding depth. By using LFPSplitter one can obtain various information about the data and use obtained files for further processing of light field photography.

5.3 Light Field Toolbox

Another open source tool to not only extract but also to process light field data is Light Field Toolbox in Matlab [29] by Donald Dansereau. The first version (v0.1) purely focused on Lytro imagery and was limited to functions used for loading, decoding, colour correction and visualization of light field data. However, during the time this thesis was written, newer version (v0.4) was released, which contains more than 35 functions. Functions in this toolbox can be divided into several groups: decoding/input, filtering, image adjustment, visualization, calibration and

utility. Decoding and input type contain functions for decoding LFP or raw file and functions to batch and recursively process light field images. Light Field Toolbox can be used to process other light field formats than LFP, for example, the gantry-style light fields from [11]. It also creates grid model of lens array using raw white images extracted from the camera. The second group of functions is focused on filtering as it contains fully functional and even demo functions to create and apply 2D and 4D filters for linear depth/focus and de-noising. The toolbox also contains functions for image adjustment such as colour balance, gamma correction and histogram equalization. There are several functions which can be used for light field visualization (not only .LFP format visualization) with a user having the possibility of controlling the content or with a predefined path of showing the 2D slices of the light field. Light Field toolbox is very convenient tool to start experimenting with light field data as it contains variety of functions for multiple applications and as it is still being supported and developed. Author also created a community on social media Google+, where questions/answers, practical tips and applications are shared[1].

[1]Light Field Toolbox community,
URL: `https://plus.google.com/communities/114934462920613225440`. Cited 03/12/2017.

Chapter 6

Light Field Data

This chapter brings an overview of possible data formats of Lytro camera images and its possible representations/applications. Terminology described in this chapter is further used in the practical section. Focus is on the light field data gathered from Lytro cameras, because this type of light field data is widely used in papers for compression and processing. Another, not negligible, fast is that Lytro Illum datasets are the most frequent in the community.

6.1 Formats and datasets

In this section, several possible data formats are shown in order to better understand the performance of each separate compression scheme. One of the most frequently used datasets in latest papers is Lytro Illum dataset from École Polytechnique Fédérale De Lausanne (EPFL) [30]. A dataset of approximate size 55 GB contains 118 light field images captured with Lytro Illum camera in uncompressed raw format (each LFR file of size around 50-55 MB). Furthermore the dataset contains files which are extracted using Lytro Desktop software, like depth map, the relative depth of field coordinates, calibration data and image thumbnails [30]. The dataset also contains 4D light field images, which are obtained by Light Field Matlab Toolbox [29][31] (toolbox has been already described in chapter 5). Images are divided into 10 different groups based on its content (ISO and Colour Charts, Buildings, Nature, Grids, People etc.). Dataset can be used for benchmarking of novel algorithms for light field data image processing, compression and quality evaluation. Subset of 12 images from this dataset was already used in International Conference on Multimedia and Expo (ICME) challenge, where it was used to evaluate performance of submitted papers on light field data compression [32].

Each of the files from dataset has been processed by Light Field Toolbox which resulted in MAT-file for each image, containing light field data in 5D representation - $LF(u, v, x, y, ch)$. Where dimensions $x = 434, y = 625$ (for Lytro Illum camera) represent number of micro-lenses in horizontal and vertical direction respectively. And $u = v = 15$ (again for Lytro Illum) represent number of pixels under each lenslet. From what has been said it is clear that spatial resolution of Lytro Illum camera is 625×434 and directional resolution is 15×15.

When all 15×15 pixels (image data from sensor under each lenslet) are taken and formed

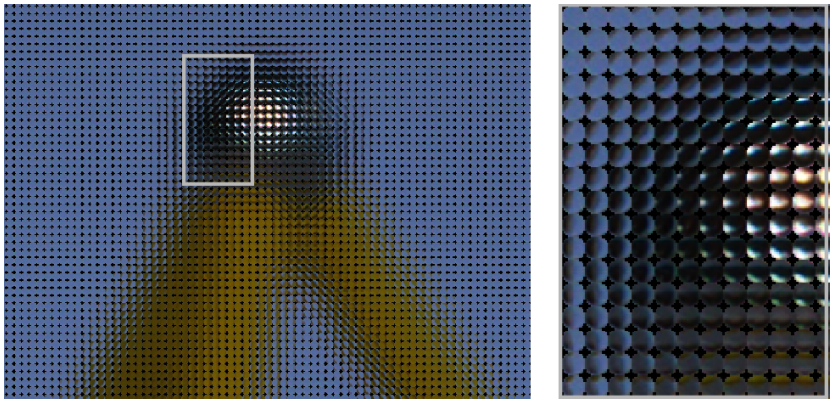

Figure 6.1: Light Field image data shown in raw lenslet structure.

into 625×434 "lenslet" array, where the raw sensor lenslet information is aligned as can be seen in Figure 6.1. The lenslet image was cropped so the lenslet structure can be seen in print as well. It can be seen how the image under each microlens is reversed. Note the image shape under each lenslet, where corner pixels have no information. However, some methods exist for filling in these regions [31] to extrapolate for outside views which has not been even recorded.

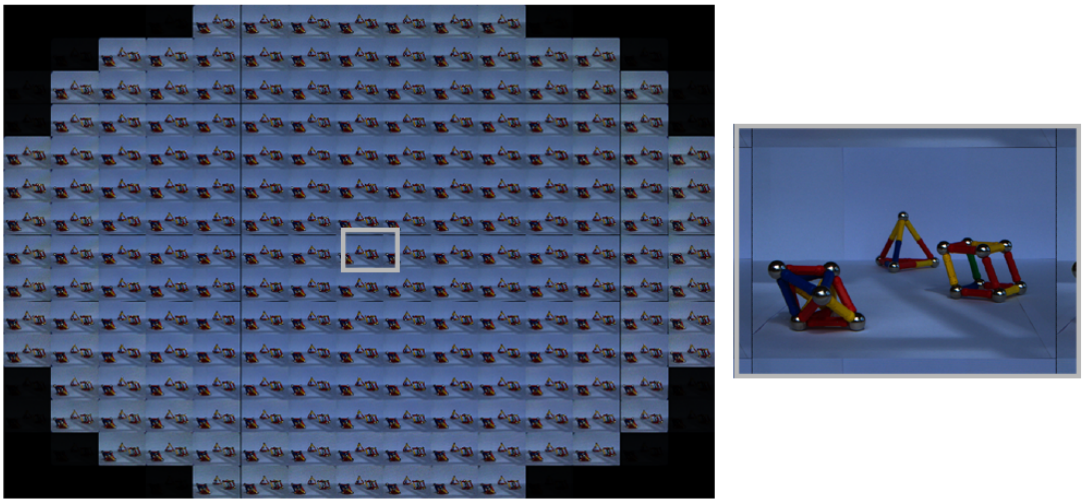

Figure 6.2: Light Field image data shown in array of sub-aperture images.

If one pixel value is taken at the same position under each lenslet one sub-aperture image is obtained. Therefore 15×15 individual sub-aperture images can be obtained. All sub-aperture images are shown in Figure 6.2 also with close-up to one sub-aperture image. The array contains image data in circular shape, that is because the micro-lenses are also circular.

Another way to represent light field data can be through epipolar images [2]. In this case v and y stays fixed and u and x varies (in $LF(u, v, x, y, ch)$). Set of several epipolar images is shown in Figure 6.3 together with one sub-aperture view for reference. In each epipolar image, u varies horizontally and x vertically with spatial resolution 434 pixels and directional resolution 15 pixels. Depth of objects within the scene can be estimated based on the slopes of lines in epipolar images. The greater the slope is the further distance from world focal plane [2]. Note

that there is three-sided pyramid in background and cube in foreground (on the right). As can be seen, the blue line of the pyramid has negative slope in each epipolar image (as well as two red lines). Lines of the cubes have almost zero slope (vertical lines). This corresponds to the fact that pyramid is in further distance.

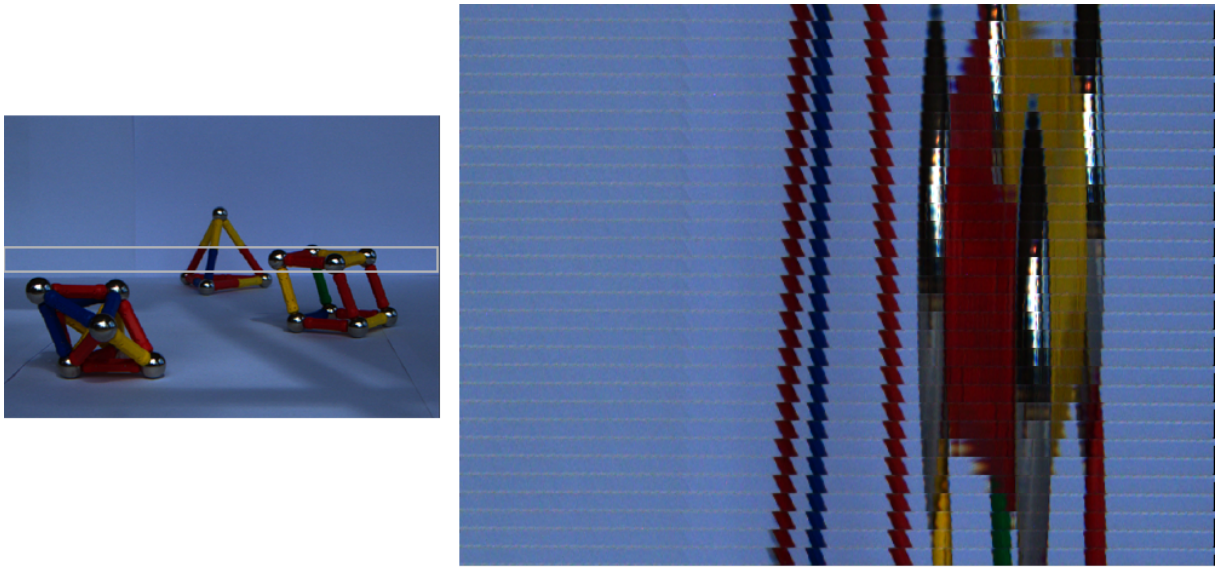

Figure 6.3: Light Field image data shown as epipolar images.

Fifth dimension *ch* has four components, where first three are RGB coordinates and fourth is pixel confidence weight channel [31]. The confidence channel represents confidence related to each pixel. Weight values are highest in the middle of each lens and equal to zero in dark corners where no information is recorded. Confidence weight channel can be used for example in filtering or histogram equalization applications [31]. Typical light field data structure which is obtained from a 5D LF array can be reorganized and imaged as example in Figure 6.4.

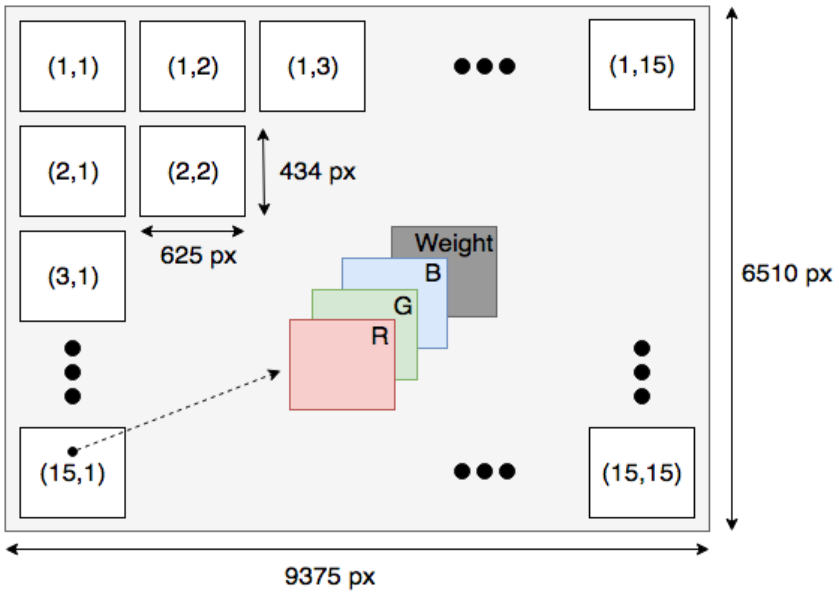

Figure 6.4: Light Field data structure of sub-aperture images stacked into 2D array.

Sub-aperture images (in literature sometimes called all views) are easily obtained from LF (in Matlab convention $LF(1:15, 1:15, :, :, 1:3)$) and then can be stacked in 15×15 matrix as shown in Figure 6.2. From the image and it's closeup it is once again clear that there is significant spatial redundancy between neighbouring views/images.

Difference of each view and mean average image of all views is depicted in Figure 6.5. The first difference of all views with a mean image is computed, then a sum of all pixel values in residual image is calculated. The relative difference (blue = 0, yellow = 1) is depicted Figure 6.5. It can be seen that the most views close to the mean are located in the centre of all views structure. Around the edges images are more different from the mean image. This is caused by the position of the view itself, but also the fact that boundary sub-aperture images contain artefacts and are colour-distorted.

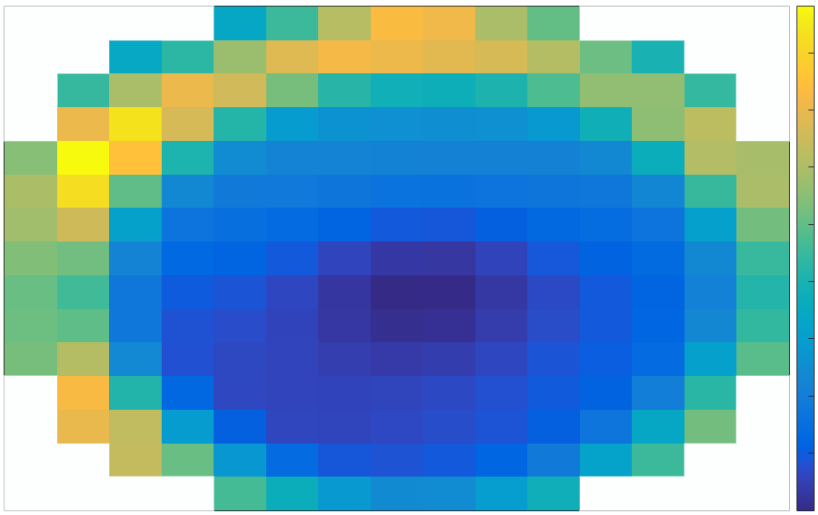

Figure 6.5: Relative difference of each view when compared to mean view - blue = 0 (no difference), yellow = 1 (maximum difference).

6.2 Data representation

Light field data can be represented in various ways as is also briefly mentioned in chapters 5 and 9. Based on the whole idea of light field data, this type of data can be used and approached in different manners.

6.2.1 Depth Estimation

Depth estimation in light field data is more convenient, robust and accurate (thanks to its multiple views) in contrast to stereo capturing systems, which needs to be calibrated [22]. Depth estimation can be based on the slope of lines in epipolar images [2] which was described previously in this chapter. Built-in algorithm for depth estimation is also available in Lytro Desktop software which comes together with Lytro cameras. Also several other improved algorithms are

developed and are mentioned in comprehensive overview [22]. In Figure 6.6, examples of two depth estimates are shown (images used are from [33], but the algorithm itself is not relevant for this thesis).

Figure 6.6: Example of depth estimation representation of Lytro Illum data. Left - input image, Middle - algorithm from [33], Right - Lytro Desktop depth estimation. Source [33].

6.2.2 Change of Perspective

Change of perspective is the underlying idea behind the angular-information recording device such as light field cameras. As was shown and explained in previous sections, sub-aperture images can be obtained from raw plenoptic data when the lenslet structure is known beforehand. After

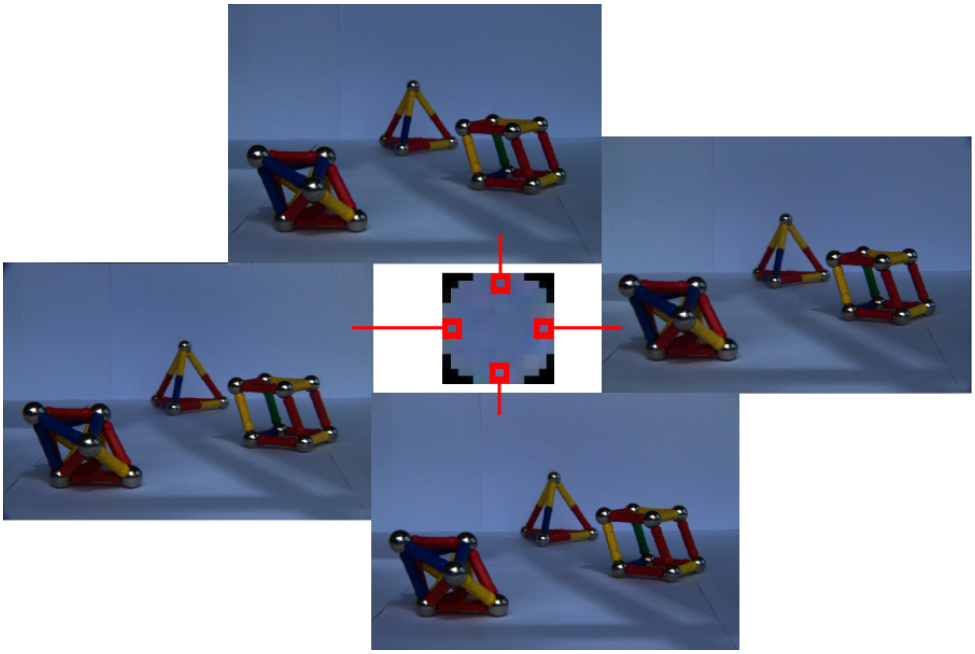

Figure 6.7: Example of perspective change - four sub-aperture images were generated (two furthermost in vertical parallax and two in horizontal) by the method described in chapter 6.1.

that individual sub-aperture images can be reorganized into required structure so it can be used for interactive user navigation within light field data. This perspective navigation is also included in Lytro Desktop and also one simple implementation can be found in Light Field Toolbox [29]. The core of this same implementation was used in the compression tool implemented within scope of this thesis and modified accordingly to meet the requirements.

6.2.3 Refocusing

Light field data allows to refocus photographs after exposure therefore to overcome the so called focus problem [2]. In Figure 6.8 it can be seen that one-exposure raw light field image can be refocused to new arbitrary individual planes in the captured scene.

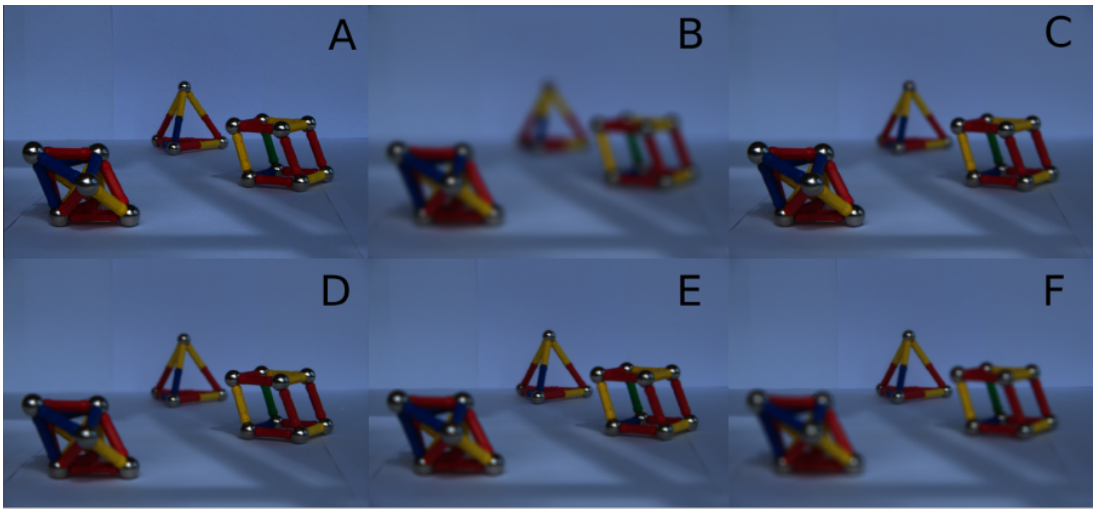

Figure 6.8: Example of digital refocusing - A - all in focus center sub-aperture view, B to F - focus plane is moving from camera to background of the scene.

First demonstration of light field digital refocusing was proposed in light field rendering paper [6]. After that several implementation papers were published since year 2000 until now about digital refocusing, also called synthetic aperture photography. Extensive overview of refocusing algorithms can be found in Ren Ng's dissertation thesis [2]. Digital refocusing can be imagined as a ray-tracing method, where recorded light rays are traced to the point where they would have finished in virtual and simulating imaging sensor, which sums the light at each point in virtual plane [2]. It is possible to utilize the recorded angular information to define new image focused on focal plane at some distance.

Digital refocusing can be implemented as basically summation of dilated and shifted copies of sub-aperture images over the whole aperture as described in [2]. This same principal (among others) is used in Light Field Toolbox and was used for generation of images in Figure 6.8. More sophisticated algorithms are described in [2], [34] (with Fourier Slice Theorem), [7], [35] and also in the Light Field Toolbox [29].

Chapter 7

Compression

In this chapter, lossy and lossless compression techniques for light field data are described. The number of different captured views during light field data acquisition is high (varies on the type of capturing technique) and therefore the amount of light field data to be processed is immense. For example in Lytro cameras, where the acquisition type is based on spatial multiplexing, it is crucial to have a huge number of sensitive cells and therefore the amount of data has to be large. This calls for efficient and fast compression techniques which are necessary for fast data transmission and storage [36]. On the other hand, light field acquisition with Lytro cameras is basically about recording the same scene from different and close viewpoints, therefore there is a spatial data redundancy which has be exploited in order to achieve high compression ratios [6].

The increase of various devices capturing approximation of plenoptic function calls for a unified standard for formatting and compression of plenoptic data. A number of commercial options to capture omnidirectional, depth-enhanced, point cloud, holography or light field content have emerged recently, from which all have different ways of data creation, format conversion, encoding, decoding, rendering and displaying. JPEG committee put themselves a task to develop a standardized framework to facilitate capturing, representation and exchange between different modalities [36]. JPEG Pleno was launched in 2015, work item has officially started in 2016 and a first working draft should be produced in second half of 2017 (with an aim to produce a first international standard in 2018). JPEG initiative aims to create file format with plenty of interesting features and potential applications [36]. JPEG Pleno will have the ability to change the field depth after capture, change the focus (refocus on objects) after capture, change the lightning in already captured (or synthesized) scene, change the perspective and viewpoint position and allow analysis and manipulation of objects within a scene [36]. But for now, there is no such standard for captured light field data and each manufacturer or developer creates its own file format, structure or compression technique.

In July 2016 another initiative, called ICME 2016 Grand Challenge, has launched with a goal to achieve efficient image compression techniques, visual quality assessment methodologies

State-of-the-art compression schemes	
Still Image	JPEG2000 - [44], [45], [46]
	JPEG XR - [47], [48]
	SPIHT - [46], [49]
Video	AVC/x264/H.264 - [46], [50]
	HEVC/x265/H.265 - [51], [52], [53]
Ad hoc solutions for light field data compression	
Transform Coding	[54], [55], [56], [57], [58]
Predictive Coding	[58], [39], [43], [42], [40]
Pseudo-sequence Coding	Sub-aperture Images Sequence Coding - [59]
	Data Formats for High Efficiency Coding of Lytro-Illum Light Fields - [60]
	High Efficiency Coding of Light Field Images Based on Tiling and Pseudo-temporal Data Arrangement - [38], [61], [62]
	Pseudo-sequence-based Light Field Image Compression - [39]
	Interpreting Plenoptic Images as Multi-views Sequences For Improved Compression - [43]
	Other - [63], [64], [65], [66], [67], [41]

Table 7.1: Overview of compression schemes (groups) described in following subsections and respective references to literature.

and test materials for light field images [37]. This challenge summoned for contributions to find effective compression of light field image data [38]. Authors of the ICME Grand Challenge informed its contributors in detail on the call and evaluation procedure to be used for assessment of proposed algorithms. Several response articles ([38], [39], [40], [41], [42]) from this challenge are also mentioned in this thesis. Another challenge call for proposals took place in September 2017, called International Conference on Image Processing (ICIP) Grand Challenge 2017 - Light Field Image Coding. However during period of this thesis, only one paper [43] has been published online and is mentioned later pseudo-sequence coding section.

During recent years there has been a large number of papers presenting about this topic. It is important to say that currently there is no standard in compression of light field data and there is a long way to go to achieve such unified compression scheme. Especially when the compression algorithms are divided also by the type of light field data acquisition. Compression schemes can be divided into three main groups - progressive/transform coding, predictive coding and pseudo-sequence coding [22]. As will be seen in this paper, the challenge of light field compression can be addressed in different ways. Table 7 shows an overview of compression schemes described in following subsections. Note that some of the references fall under more groups as the definition is not always straightforward.

7.1 State-Of-The-Art Compression Schemes For Conventional Image Data

Several state-of-the-art compression schemes will be discussed in this chapter as these are being exploited in plenoptic content compression (for Lytro cameras). Several works were dedicated to analysis of their performance when applied directly to the raw data or some other modification of the plenoptic content.

7.1.1 JPEG2000

JPEG2000 is a wavelet based standard for digital still image compression, published by JPEG as International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) standard and International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) recommendation [44]. The standard was initially meant for compressing different types of image data (bi-level, multicomponent, ...), different applications (scientific, natural, text, synthetic, ...) and different imaging technologies. Its most known predecessor, JPEG, was used for more than a decade by the time JPEG2000 was standardized, so there was need for new compression algorithm as the demand for small file sizes and high image quality and size is still increasing. Figure 7.1 shows generalized JPEG2000 engine. This simplified diagram may look similar to classical JPEG encoder/decoder with the exception of Discrete Wavelet Transform (DWT) instead of Discrete Cosine Transform (DCT), but there are differences along the whole process. The performance of the compression scheme itself and all of its features is well analysed and explained in [45], [46], [44] and only the basics are summarized in the following sections.

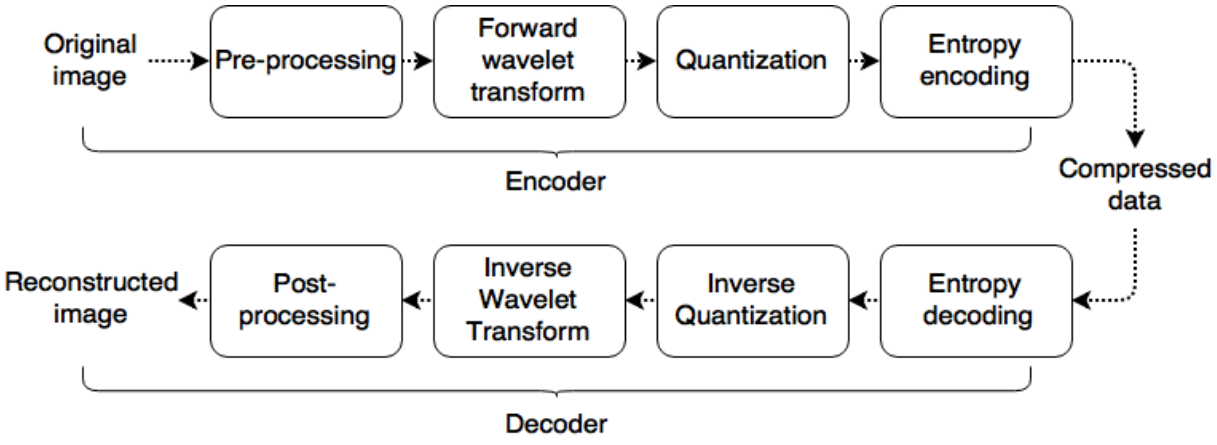

Figure 7.1: Generalized block diagram of JPEG2000 encoder/decoder.

Compression scheme

Firstly, in pre-processing part, the input image is tiled - partitioned into rectangular, non-overlapping tiles which are further processed independently [46]. Tile size can vary and equal to

the dimensions of the input image (one tile). Tiling process makes the compression engine less memory dependent and it is also found useful for Region Of Interest (ROI) compression. ROI is used in JPEG2000 when part of an image is more important than the rest and therefore it is to be transmitted in better quality. It has been shown that size of the tiles affects the quality of reconstructed image both objectively and subjectively [46].

Next step in pre-processing is DC level shifting. Before the tile is processed by DWT, all samples of the tile are subtracted by the same value to get DC level shifted. The values (only unsigned) are subtracted from $2^{(n-1)}$, where n is bit depth of the colour component. DC level shifting ensures that the values stay within the range -2047 to 2048 for 12-bit depth content.

After tiling and DC level shifting is done, component transformation comes in place. Irreversible/reversible (for lossy/lossless compression) component transform matrices are applied to individual components in order to achieve colour decorrelation. For example colour components RGB are being transferred into luminance and chrominance components YC_rC_b. In JPEG, chrominance components were subsampled (4:2:0 or 4:2:2), but in JPEG2000 the format stays 4:4:4 as the subsampling if happening later in wavelet transform in encoding process.

DCT was used in preceding still image compression scheme, JPEG, but there have been several reasons to use DWT in JPEG2000 in order to achieve set goals, described more in detail in [46]. The wavelet transform is applied to tiles components after the pre-processing is done. Discrete wavelet transformation provides different decomposition levels, which contain coefficient of vertical and horizontal frequency characteristics of an input image (tile). Two types of filters are used in JPEG2000, Daubechies 9-tap/7-tap filter for irreversible transform and Le Gall and Tabatabai's 5-tap/3-tap with integer coefficients for reversible transform [46].

After the DWT, quantization takes place. Quantization is a process during which samples are reduced in precision. All coefficients are linearly quantized by dead band zone quantizer [46]. This process is usually lossy unless the quantization step is 1 and the coefficients are integers (case of reversible transform). Quantization step can vary across tiles and across bands, but one step is allowed for one sub-band in each tile [46]. Size of quantization step is decided based on perceptual importance of particular band HVS or by some other aspect like bit rate availability [46].

Quantized coefficients are entropy coded in order to achieve a compressed stream of bits. JPEG2000 uses Embedded Block Coding with Optimized Truncation (EBCOT), which was also selected in order to meet the set goals [46]. In the EBCOT process, sub-bands are partitioned into rectangular non-overlapping code blocks and these code blocks are independently encoded.

Features

JPEG2000 comes with several very useful features which should be mentioned. One of the most significant features is the option to select ROI in the input image. This is found to be useful in applications where some parts of the image are more important than the others. Let's say that in medicine, X-ray image may have important part with a fracture that requires attention and

the rest are not or less important. This region is then compressed at higher quality and during the transmission is transferred with higher priority or simply first. JPEG2000 uses MAXSHIFT method, which does not need to have shape information about ROI at decoder [45]. This is because all coefficients which are in ROI are scaled (shifted) above the background in a way that their bits are at higher level [46]. Experiments in [45] have shown that for an image with resolution of approximately 2000×2000 with ROI of 25% relative size of the image, the method increases bit-rate by approximately 1%.

Next interesting feature is the scalability in the spatial domain and in Signal-to-Noise Ratio (SNR). In spatial scalability, images with several resolutions can be rendered from single compression, single bit-stream [45]. At least two resolutions/layers are required. The lower layer is coded simply as a low-resolution image to provide the basis. Then any other higher resolution is coded like enhancement layer, which makes use of the lower layer by interpolating into full/higher resolution [45]. This is done easily thanks to the principles of DWT by prioritizing lower bands before higher bands on bit plane level.

The SNR scalability, similar to spatial scalability, produces at least two quality different versions of the same image with same resolution from a single bitstream. And again, the lowest quality image (with acceptable SNR) is considered to be a base layer and the remaining parts of the bitstream are called enhancement layers as they are used to enhance the base layer to produce multiple images with increasing SNR. JPEG2000 supports also the combination of spatial and SNR scalability [45].

Other features include several error resilience tools to cope with channel/transmission errors, visual frequency weighting which weights frequency bands based on HVS and new file format (JP2) with intellectual property rights information [45].

7.1.2 JPEG XR

JPEG XR, block-based compression scheme, can be seen as a follower of JPEG not only for the High Dynamic Range (HDR) photography. The lossy and lossless compression scheme was originally developed and patented by Microsoft under the name HDPhoto in 2006 [47]. Later in 2007, JPEG together with Microsoft announced that HDPhoto will be considered as JPEG standard under the name JPEG XR. And in 2009 JPEG XR was announced to be ITU-T recommendation and ISO/IEC standard [47].

The objectives of JPEG XR were to produce new compression format which supports HDR photography formats, web imaging, interactive applications, better compression for enhanced quality, cost-effective computational performance and new progressive coding schemes for powerful image access and manipulation [47]. Classic JPEG supports bit depth from 8 to 12 bit maximum, however, JPEG XR is constructed to support up to 32 bits per pixel. It outperforms original JPEG as it can achieve the same perceivable quality with up to twice higher compression ratio [47].

JPEG XR coding process is very similar to traditional JPEG and it shares principles with other

image compression schemes [48]. The input image is deprived of redundancies by using linear decorrelation matrix, similar as in JPEG2000, and transferred into luma-chrominance colour space. JPEG XR then applies 4×4 orthogonal overlapped block transform unlike JPEG2000, which is using DWT. Coefficients are quantized by dead zone band quantizer and quantization parameters can vary by blocks [48]. Quantized data are then entropy coded in a very similar way as it is done in JPEG, with some differences more described in [48].
JPEG XR supports tile structure as the image is partitioned into tiles and each tile is processed separately [47]. During decoding, the tiles can be accessed separately, therefore, the regions of our interest can be accessed selectively without needing to decode the whole image. JPEG XR also supports more colour accuracy with higher bit depth and also by supporting multiple colour spaces (CMYK, grayscale, multi-channel).

7.1.3 SPIHT

Another, wavelet-based, powerful tool in the field of still image compression is Set Partitioning in Hierarchical Trees (SPIHT). SPIHT, which was introduced in 1995, is also based on DWT and it is efficient extension of Embedded Zero Tree Wavelet (EZW) [46]. Both of these techniques exploit the fact that there is certain magnitude correlation between the decomposed bands. Algorithms use a tree structure to detect and exploit similarities across subbands. It can be said that SPIHT exploits characteristics of the wavelet transformed images to increase coding efficiency [46]. The uniqueness is in the fact that SPIHT does not transfer pixel values or its coordinates, but decisions which have been made in each step of the trees that define the image structure [49]. And because only decisions are transferred and if the encoder and decoder will have an identical algorithm, the identical image can be reproduced. Whole compression scheme is well described in [46] and [49].
SPIHT allows such a features like scalability or progressive image transmission, which is important in web applications, when a low detailed image is loaded first and its quality is enhanced with a number of bits received [46].

7.1.4 AVC/x264/H.264

Advanced Video Coding (AVC) (x264/H.264) is probably the currently most used video codec, which was formally introduced in 2003 [46]. It is still being used in high number of applications. x264 uses Macro Block (MB) of size 16×16 as a basic coding unit, where each MB can be further divided into smaller blocks if necessary to obtain higher compression gain [46]. These smaller blocks are either intra or inter coded. Group of MBs is called slice - there are no limitations in slice sizes (one slice can contain one MB or whole frame). Introduction of slices within the frame was novelty between video compression mechanisms. Slices can be fixed with number of MBs (different sizes of packets) or fixed with bytes (almost same byte sizes of packets, different number of MBs). Predecessors of x264 were using coding tools applied on frame types (I,P,B), however here are applied to frame slices [46] and each slice type is coded differently.

Residual pixels after inter/intra prediction are zig-zag scanned, quantized and entropy-coded (together with motion vectors and addressing information) [46]. Whole compression mechanism is thoroughly described in [46].

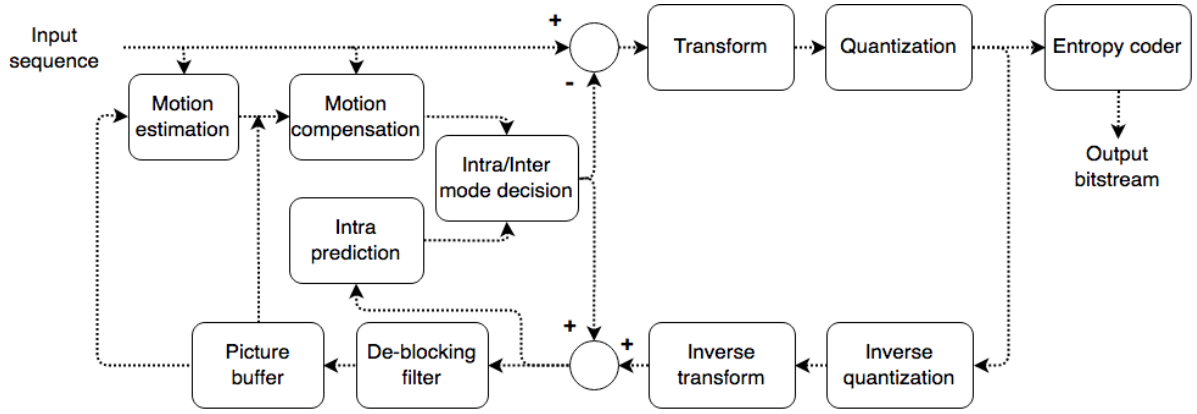

Figure 7.2: Generalized block diagram of x264 encoder.

7.1.5 HEVC/x265/H.265

High Efficiency Video Coding (HEVC) [51] is the latest coding standard for video. It will be shown later that compression standard which is initially intended for the video can be used for coding of static plenoptic content. First version of HEVC, also known as x265, H.265 or MPEG-H Part 2, was developed by joint of groups from ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) [51]. The scheme was developed in 2013 and then in years from 2014-16 several versions were released containing extensions for 3D video, multi-view, range extensions, scalability, screen content coding etc. The main goal was to developed new coding mechanism which will in future replace x264 (AVC) with addition to efficiently encode high resolution content together with the use of parallel processing techniques [51].

In general, the coding scheme is as follows. The input picture is partitioned into Coding Tree Blocks (CTBs) (which can vary in dimensions in contrast with HEVC predecessor, x264, which is strictly using 16×16 dimensions of MB). The possibility to change size of CTBs can be chosen based on the image content or if there are computational or memory restrictions [52]. One Coding Tree Unit (CTU) is formed by luma CTB and two chroma CTBs, each chroma CTB contains half of the samples of luma CTB. CTU is basic unit of HEVC - can be seen as MB in previous standards. CTBs can be further partitioned into smaller blocks, called Coding Blocks (CBs), of variable sizes based on the content and characteristics of the content under the particular CTB [52]. Smallest CB can be size of 8×8 and can go up to the size of CTB. Luma CB and chroma CBs together form Coding Unit (CU). Intra and inter-picture is computed for each CU, where luma and chroma CB can be formed by one to four blocks called Prediction Block (PB) [52]. PBs under each CB can have the same size of can have asymmetric sizes. Prediction signals are generated and encoded for all partitioned sample locations. Explaining all princi-

ples of mechanisms used in HEVC is beyond scope of this thesis and are described in [51] and [52].

7.2 Transform Coding

Techniques in this group of compression schemes are mainly based on some type of transform. In recent years there is an increasing number of scientific papers, exploring the borders of current compression algorithms and proposing novel compression ad hoc algorithms for light field data [38].

Study on how current state-of-the-art algorithms perform on light field data was addressed in [54]. Light field data structure is significantly different from classic image data, therefore authors main goal was to answer two questions - first, if existing compression schemes can be used for plenoptic images and second, what effect do they have on rendered images. Three, well-known, image coding standards JPEG, JPEG2000 and SPIHT were subjectively compared. The analysis was made not on plenoptic image reconstruction, but on rendered views. Compression analysis framework was following: first, the plenoptic image was processed by compression algorithms. After that reconstructed plenoptic image was acquired after lossy decoding and multiple views were rendered. Quality of these rendered views was then evaluated by objective metrics - Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Rendered views directly from a plenoptic image (without being processed by any compression scheme) were used as a reference for objective assessment. In general JPEG2000 and JPEG outperformed SPIHT scheme and JPEG2000 was preforming a bit better than classic JPEG compression. This holds true for both metrics - PSNR and SSIM. JPEG2000 was found to be best compression scheme when it comes to compression of common still images [54] and this was found to be true even with plenoptic images. SPIHT was found to be most effective for lower bitrates, therefore, it can be useful in remote applications where lower bandwidth is typical and where its lower complexity (compared to JPEG2000) can be found useful too. One of the disadvantages when compressing with standards used for common images were blocking artefacts when the compression block does not match the size of micro-images [54].

In order to understand and develop new algorithms specially tailored for plenoptic raw data, it is necessary to evaluate and explore the performance of the state-of-the-art compression algorithms and also to research statistical properties of raw plenoptic data [55]. In [55], the performance of state-of-the-art lossless and lossy compression schemes was analysed when applied to plenoptic raw images. Another goal of the paper was the evaluation of quantization effects to the quality of final reconstructed views. A dataset of raw plenoptic images taken with the first generation of Lytro camera was used for experimental analysis [55]. As dataset contained images with resolution 3280×3280, 12 bits per pixel, compression schemes which are capable of compressing this bit depth had to be used (JPEG2000 and JPEG XR). Performance of compres-

sion schemes was evaluated by measuring the PSNR and SSIM between original extended focus image and decoded extended focus image [55]. Reference approach was formed by quantization and entropy encoding (7zip). JPEG2000 and JPEG XR outperformed the reference in both used metrics. It has been shown that compression (both JPEG formats) gives almost same results at 4.36 bits per pixel (bpp) when compared to a lossless version of JPEG2000 (8.80 bpp). Both compression schemes started to show visible artefacts around 2 bpp which was confirmed by objective metrics as well. JPEG2000 performed slightly better than JPEG XR, but the difference is nearly negligible. Results proved that it is possible to use current state-of-the-art compression standards to raw plenoptic data with no perceivable difference [55].

Later in 2017, the same author used again JPEG2000 compression scheme in [56]. In this case, JPEG2000 encoder was applied not directly to raw light field data format, but now on rendered views. Performed experiment showed that this change of steps performs better than applying JPEG2000 or JPEG XR directly to 2D raw light field data format [56].

3D DCT compression algorithm was presented in [57]. This algorithm exploits the intra sub-image correlation. First, three-dimensional DCT is applied to achieve the coefficients. Then, three-dimensional quantization array is applied to obtain quantized coefficients which are entropy coded by hybrid run-length/Huffman coding. Input data are sequences of N images placed after each other along the third dimension and 3D DCT then produce a de-correlated group of sub-images. Groups of $N = 4; 8$ sub-images were tested in two different grouping methods (1 by 8 or 2 by 4 for $N = 8$), where each sub-image was 8×8 pixels. Results showed that this method outperforms baseline JPEG drastically. It was also shown that the scheme performs differently on different grouping methods.

7.3 Predictive Coding

In this group of coders, prediction algorithms are applied to some form of plenoptic content. It should be mentioned that it is not easy to unambiguously divide compression schemes into individual groups as they may contain coding tools from each group in some form. For example, authors in [58] presented prediction based algorithm with wavelet packet. The whole scheme is more of a combination of predictive and transform coding. Firstly the images are decomposed into sub-bands using wavelet packet transform. These wavelet packets are divided into predictable and not predictable bases. This selection is based on several criteria - relative energy, relative amplitude, correlation. One group of sub-bands with significant coefficients has large relative energies and are highly correlated, the second group contains information which is isolated for each image [58]. The first group is called predictable and the second group unpredictable. The disparity map is estimated based on symmetrical neighbouring images, this map is used afterwards in coding. Images decomposed into sub-bands and partitioned into a group of basis are then coded by DCT and Huffman coding. Group of images is reconstructed

by first predictable basis (maximum relative energy). If the reconstructed images meet wanted quality (which was determined prior to the coding), no more information needs to be added. If no, another predictable basis is added (second maximum relative energy) until reconstructed images meet the criteria. Center image is then predicted from these four corner images. The precise order of prediction is more described in [58]. However, algorithm was evaluated on two images only with no other reference method.

7.4 Pseudo-sequence Coding

Another group of coding methods for plenoptic content is called pseudo-sequence coding, because these methods are using current state-of-the-art video compression schemes. As can be seen, in Figure 6.2 and as have been already mentioned, individual rendered views are highly correlated. Therefore it is straightforward to exploit spatial redundancy in plenoptic content. A typical sequence of light field data formats can be seen in Figure 7.3. Of course, most of these algorithms could be also mentioned in the previous group (predictive coding) as these algorithms such as HEVC are using prediction motion.

7.4.1 Sub-aperture Images Sequence Coding

One of the first proposed method to code views rendered from plenoptic raw data as pseudo-video sequence was presented in [59]. Authors extracted sub-aperture images directly from light field obtained by Lytro Illum camera (an example of sub-aperture images arranged in an array can be seen in Figure 6.2). Images were then rearranged by line or rotation scan mapping into the sequence of images which can be thought of as a video stream. Authors used x264 video compression algorithm without any modifications to the standard video encoder to compress the video stream made of rendered views [59]. Two ordering types were evaluated - standard line and rotation scan mapping to rearrange images from a 2D array (Figure 6.2) into a video stream. Results have shown that rotational scan mapping outperforms the later in terms of PSNR. In addition, a performance of x264 on multiple rendered views was also analysed in contrast with classic JPEG encoder applied either on the rendered views as well as on plenoptic raw image data. Obviously, the proposed method with x264 encoder outperformed conventional JPEG compression scheme used directly on lenslet image and on sub-aperture images in terms of PSNR. It was again confirmed that JPEG applied on sub-aperture images showed superior performance when compared to JPEG applied on lenslet raw image data.

7.4.2 Data Formats for High Efficiency Coding of Lytro-Illum Light Fields

Another approach to encoding Lytro Illum camera light fields is presented in [60]. In this case authors used HEVC/x265 on five different light field data formats. Two of the formats were sequences of views from sub-aperture views matrix shown in Figure 6.2. Sequence formation

was used exactly the same as in previously mentioned paper - line scan mapping (here called raster), where views are gathered from top to bottom and from left to right to produce pseudo-video sequence. The second one was again rotation scan mapping (here called spiral), where views are gathered from the centre of all views matrix and going outwards [60]. Remaining three formats were following - lenslet image (same as previous work), sub-aperture views matrix (Figure 6.2) and again lenslet image, but in this case corners of each micro-lens were filled in by neighbouring pixels. Three latter light field data formats were compressed as a still image using the HEVC Still Image Profile and the former two were compressed by HEVC video encoder with following configurations: Random Access, Low Delay P, Low Delay B, All Intra. This work proved that even though only still images are used for compression, their format/arrangement can have a significant impact on coding performance [60]. By exploiting different format and different HEVC configurations an improvement of 10 dB in PSNR can be obtained. Coding of the lenslet image proved to perform the worst of all selected scenarios on average. This is due to the high-frequency content made by micro images structure (small repetitive images, dark corners) [60]. The poorest performance of light field data format compression was in cases where even the image had a higher frequency content. Nevertheless; in cases, where an image does not contain much high-frequency content, All Intra HEVC configuration achieved worse results. HEVC video coding scheme performed much better than still image compression as it could be expected. In contrast with [59], it was shown that there is no significant difference between spiral and line scan mapping in terms of coding performance when HEVC is used. The conclusion of the paper is that there are not negligible differences of coding performance between different light field data formats and therefore further research is needed in order to develop a coding scheme which would perform more consistently across various configurations and light field data formats with various content [60].

7.4.3 High Efficiency Coding of Light Field Images Based on Tiling and Pseudo-temporal Data Arrangement

Another approach which can be included in the category of pseudo-sequence compression was presented in [38], from group of authors with previously mentioned work [55], [68], [60]. This work was also a response paper to the call for proposals of ICME Grand Challenge 2016 mentioned before.

This low-complexity method is preprocessing raw light field data structure into a pseudo-temporal sequence of frames which are compressed by standard HEVC [38]. The proposed method is trying to exploit the data structure by form of reorganization of source data. The first step lays in partitioning raw light field data (shown in Figure 6.1) into non-overlapping tiles with resolution $T_W \times T_H$. Authors empirically found dimensions $(T_W, T_H) = (464, 320)$ to provide sufficient correlation between tiles which are further encoded. Tiles are reordered in raster/line scan mapping (same as in previous work, but here applied to different light field data format). This reordering scheme is based on assumption that adjacent tiles are more correlated

than non-adjacent tiles. The last step of whole compression scheme is HEVC encoder, which in this case can make use of both spatial and angular information redundancy. Spatial redundancy is exploited by the inter-frame prediction and angular redundancy is exploited by intra-frame prediction as motion displacements. The proposed method performed better when compared to a reference JPEG compression applied to the whole lenslet image, especially at lower bitrates. For lower compression ratio (bpp = 1) the difference between scheme which exploits spatial and angular data redundancy (proposed) and scheme which only exploits spatial dimension (JPEG) seems to vanish [38].

In [61] authors went a little bit further and tested what effect does the tile has on the coding performance of the same method. In contrast to previous work (rectangular shaped size), square shaped tiles of following sizes were chosen: $(T_W, T_H) = \{(64, 64), (256, 256), (512, 512), (768, 768)\}$. However, analysis performed showed that changing the tile size has almost zero impact on coding performance.
Same authors evaluated proposed compression scheme against JPEG2000 in [62]. The procedure was the same - light field raw data partitioned into tiles and then encoded as pseudo-sequence. JPEG2000 compression scheme applied directly to light field raw image data. JPEG2000 outperformed HEVC in the experiment in terms of objective metrics (PSNR, SSIM).

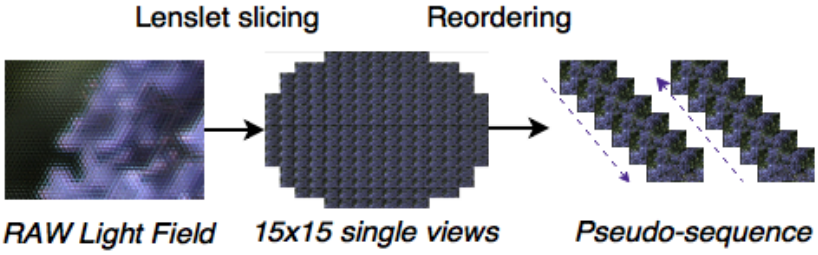

Figure 7.3: Sequence of light field data formats which is typical for pseudo-sequence based coding schemes.

7.4.4 Pseudo-sequence-based Light Field Image Compression

Another pseudo-sequence coding scheme was presented in [39], which was also a response paper to ICME Grand Challenge 2016 call. In this scheme, the raw light field image data are transferred into multiple views, which are more natural to us and also to current state-of-the-art coding schemes. Coding order is tailored in a way that high correlation between adjacent views is exploited, both vertically and horizontally. More image similarity can be found within the images in the centre area as has been also shown in this thesis in Figure 6.5. With this in mind, a centre view can be used for better prediction of other views. In the proposed scheme, the centre views are encoded as I frame (in 2D structure as shown in Figure 6.2). The remaining frames partitioned into different layers and are encoded as P and B frames in a symmetric 2D hierarchical structure [39]. Higher layer images are predicted by images from lower layers. Therefore lowest quantization parameters are assigned to I frame and higher quantization parameters to

higher layer images. Two reference video coding mechanisms, HEVC and JEM, were used and compared with two still image coders, HEVC Intra frame coder and JPEG. JEM is novel coding scheme which is based on HEVC and can be seen as its follower. Both video schemes proved consistently better performance on all tested images. However, there were some exceptions, where still image HEVC Intra compression showed better coding gain for same objective quality. The reason was not explained and is under exploration [39]. Also, it was shown that HEVC with different quantization parameter performed better over HEVC with constant quantization parameter for all images - this proves necessity of rate allocation between views in order to achieve better compression [39].

7.4.5 Interpreting Plenoptic Images as Multi-views Sequences For Improved Compression

Novelty between compression schemes using HEVC coding algorithm was proposed in [43], where Multi View (MV) extension of HEVC is being used. The proposed scheme, which is a response to ICIP 2017 Grand Challenge, represents two-dimensional prediction and rate allocation within the multi-views structure. Proposed method (same as the previous method) is only using central 13×13 out of 15×15 sub-aperture images which are obtained from Lytro Illum cameras. The reason was not explained, however it could be because of the edge images are often variously distorted. HEVC-MV extension allows this scheme to use intra and inter-frame (view) prediction and thus exploit correlation which within each image and between different view images [43]. Two-dimensional prediction scheme starts from an initial base image which is quantized with base quantization parameter. Then level 1 and level 2 images are defined, where level 1 images are predicted mainly from base frame and level 2 images use level 1 of base frame for prediction. All other frames are named as leaf frames and are predicted from all previously mentioned frames, but cannot be used for further prediction. The advantage is that leaf frames are efficiently coded because the neighbour frames are already coded [43]. Level of predictor frames is taken into account in rate allocation, where quantization parameters are distributed in a way that better quality is obtained by compression. Quantization parameter is dependent on both two dimensions of prediction scheme which is more described in [43].

Chapter 8

Light Field Image Quality Evaluation

Level of quality, both objective and subjective, is the most important factor regarding image compression in most of the applications. Few published papers were focused mainly on comparison of several plenoptic content-compressing techniques such as [54], [69], [70] and [71]. Most of them used mainly PSNR and also SSIM objective metrics and later two also performed a subjective assessment. Some of the used objective and subjective techniques are briefly described in following sections.

8.1 Subjective assessment

Quality of user experience is the utmost importance in light field data acquisition, compression, rendering and displaying technologies. Not many publications dealt with subjective and objective metrics targeted for light field data quality assessment. In [69] and [70] objective assessment, but more importantly also subjective evaluation was performed. Subjective assessment in [69] was performed on five rendered sub-aperture views from sub-set of images from EPFL light field image dataset [30]. The evaluation was performed by subjective tests based on Double Stimulus Continuous Quality Scale (DSCQS), where two images (reference and compressed with compression rate under test) were presented side-by-side. Subjects rated image quality by marking from 1 to 5 (Bad to Excellent). The position of reference and compression was randomized and not shared with subjects. Subjective tests were performed using Quality Crowd framework [69], a crowd-sourcing approach which moves a testing effort to the internet community making the subjective test more affordable, available and less time-consuming. However, this method only evaluates pairs of rendered views (still images) in a traditional way. Therefore this approach does not measure subjective perception of light field data in initially intended enriched format, where user can interact with the content [72].

In [72] new methodology for subjective tests and assessment was proposed. This approach takes into account the fact that light field content should be viewed interactively and not as the pair of rendered views only. In this, completely new approach, it is not easy to determine how many and which of multiple possible rendered views (made by refocusing, change of perspective,

...) are to be tested. Therefore the user is free to interact with the light field content using user interface. Several stimuli-comparison methods are used to obtain measured data from a subject for respective test material [72]. Data gathered from subjects were then calculated by Mean Opinion Score (MOS) using following equation:

$$MOS_i = \frac{1}{N}\sum_{j=1}^{N} m_{ji} \tag{8.1}$$

where N is number of subjects performing the test, j participant under test, i showed stimulus, m_{ji} score for i and j.

Another approach was presented in [71], where authors made 3D dense light field dataset (made of 9 synthetic and 5 real scenes to cover a large variation of scene types and lightning conditions) together with various distortions and subjective scaling [71]. And because there are different light field processing methods, different artefacts were simulated. Transmission (HEVC encoder), reconstruction (several interpolation techniques to obtain dense light field form sparse views) and display (crosstalk between adjacent views) sources of distortion were modelled for every scene from dataset. After that, large subjective assessment experiment was performed, where 40 participants compared presented light field data (each participant compared around 120-180 pairs) in pair-wise based rating method and subjective results were analysed in terms of Just Objectionable Difference (JOD) [71]. Subjective scores gathered from interactive 3D light field viewing layout were used to evaluate seven different objective quality metrics (popular image, video, stereo and multi-view metrics). Mentioned paper [71] was published during the time this thesis was being written and it was not possible to implement the objective metrics which performed well with subjective tests. However, Gradient Magnitude Similarity (GMSD) performed reasonably well and is mentioned in following sections and also implemented in compression tool which is described later in chapter 9.

8.2 Objective assessment

Numerous papers which were mentioned in chapter 7, proposed some kind of compression scheme and also performed some preliminary performance evaluation of proposed scheme. What all the papers had in common is that only objective metrics were used and most frequently the used metric was PSNR sometimes followed by SSIM. It should be mentioned, that currently there is no standardized or by researchers agreed objective metric which would be used for assessment of light field content.

8.2.1 Peak Signal-to-Noise Ratio and Mean Squared Error

Peak Signal-to-Noise Ratio, PSNR, is the ratio between the maximum possible signal power and the power of the noise signal. In compression, a noise signal is considered to be a residual signal between original image and image reconstructed using compression decoder. PSNR for two-dimensional signal (one channel) is computed as follows:

$$PSNR(x,y) = 10\log_{10}\frac{MAX^2}{MSE(x,y)} \qquad \text{[dB]} \tag{8.2}$$

(x,y) are indices of each image in sub-aperture structured matrix (example in Figure 6.4), $MAX = 255$ for 8 bit depth image, MSE is Mean Squared Error (MSE), which is calculated followingly:

$$MSE(x,y) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[IM(i,j) - REF(i,j)]^2 \tag{8.3}$$

where (m,n) are dimensions of one view ($m = 434; n = 625$). For example in Figure 6.4, $IM(i,j)$ is the image under test image and $REF(i,j)$ is reference, original image pixel value at corresponding pixel coordinates (i,j).

This means that higher PSNR value means higher image quality and vice versa. PSNR is the most elementary and most popular full reference image quality assessment metric together with MSE. The main reason of their popularity is their simplicity and apparent interpretation [73]. On the other hand these two metrics does not follows the principles of how human visual system perceives quality. It is known fact that PSNR and MSE achieved bad results when compared to structural based metrics since some types of distortions applied to the same image can result in same values of MSE and therefore PSNR (this was also demonstrated in [73]). In [74] it was found that PSNR is more sensitive to additive Gaussian noise when compared to structural based metric (SSIM).

8.2.2 Structural Similarity Index

SSIM, in contrast to PSNR, is image quality assessment model based on assumption that HVS is adjusted on extracting structural information from observed scene as it measures degradation of structural information [73]. SSIM is not calculated directly as the numerical difference between images, but as a combination of three factors - luminance comparison, contrast comparison and structure comparison.

Luminance of each of the two input signals is computed as mean intensity, then the mean intensity is subtracted from each. Contrast component is calculated as standard deviation (also for both input signals). After that the standard deviations are used for normalization of input signals which leads to structure component. Subsequently after these steps, 6 components are compared by using three comparison functions described more in detail in [73]. Three results of comparisons are finally combined (multiplication) into SSIM index, where all three

components can be importance-adjusted with their respective parameters. SSIM performs better and brings more usable information when performed locally rather then globally [73]. Authors used it initially with 11×11 circular-symmetric Gaussian weighting function to prevent blocking artefacts in SSIM index map. To obtain one number which tells about image quality, mean value of SSIM index map is calculated. Local evaluation can be also exploited when there is ROI and values in SSIM index map can be weighted accordingly [73]. The main drawback of SSIM when compared to PSNR or MSE is its higher computational complexity. In [74] it appears that values of SSIM can be predicted from values of PSNR and the other way around. SSIM is more sensitive to artefacts made by JPEG compression, but in overall it was discovered that PSNR and SSIM mainly differ on their degree of sensitivity to image degeneration [74].

8.2.3 Multi-Scale Structural Similarity Index

Multi-Scale Structural Similarity Index (MS-SSIM) was derived from its predecessor and can be seen as extension in terms of scalability. SSIM performs on single scale, but subjective evaluation of given images happens with different observation settings - display resolution, distance between image plane and observer [75]. This multi-scale method includes image at different resolutions into index calculation. First, contrast and structural comparison is calculated like in SSIM. Then, low-pass filter (down-sampling by factor of 2) is applied on distorted and reference image N-times to achieve $N-th$ scale. After each iteration of low-pass filtering, contrast and structural comparison is calculated. At the last iteration, $N-th$ scale, luminance comparison is calculated as well. The overall SSIM index is estimated from combination of obtained values at different scales similarly to SSIM. MS-SSIM was shown to outperform SSIM for all objective criteria evaluated in [75] when properly calibrated.

8.2.4 Gradient Magnitude Similarity Deviation

GMSD is a novel FR-IQA (Full Reference (FR), Image Quality Assessment (IQA)) metric, proposed in 2014 [76] and later found to perform reasonably well on light field data [71]. First, in GMSD, local quality map is computed by locally comparing gradient magnitude maps of the reference and compressed (distorted) image. Two Prewitt filters (horizontal and vertical directions) are convolved with the distorted and reference images to obtain horizontal and vertical gradient images of both. Gradient magnitudes are calculated from the results of convolution with kernels for vertical and horizontal lines. Gradient magnitude similarity map (local quality map) is pixel-wise calculated from the individual magnitudes. It was shown that gradient magnitude similarity map is highly sensitive to distortions like blocking artefacts in mostly flat areas, which is in line with the HVS [76].

Another step is to calculate average of gradient magnitude similarity map. This averaging assumes that each pixel is equally important in the objective estimation. Other similarity-based algorithms are using weighted pooling instead of average pooling, which does not every time get better results and it also increases computational time [76]. JPEG2000 compression intro-

duces usually two types of artefacts - blurring and blocking. Blurring is causing more damage in textured areas and less in flat areas. On the other hand, blocking cause more perceivable degradation in flat areas rather than in textured areas. These facts are however ignored by average pooling as it does not reflect local variations in local quality map [76]. GMSD calculates standard deviation of gradient magnitude similarity map, which takes into account that variation of local quality is correlated with the subjective quality [76].

Image distortions, which can be encountered in digital photography, lead to noticeable and visible structural changes, which are highlighted in gradient domain [76]. Authors found that using magnitude of gradient alone can be very efficient in contrast with other gradient based image quality assessment models, where additional information is being computed from gradients. This additional information can be computationally expensive with no eminent profit [76]. GMSD was found to be more efficient and accurate when compared to other state-of-the-art full reference image quality assessment models.

Chapter 9

Compression Tool

This chapter is dedicated to the practical part of this thesis. Goal of the practical part is to implement a plenoptic data compressing tool, which can be further used for finding the ways how to effectively compress light field data.
A tool which enables the use of several different compression schemes applied to various types of light field data formats is implemented. Several pre-processing techniques, state-of-the-art compression schemes and objective metrics (which are mentioned in 6, 7 and 8 respectively) are implemented. The tool is created in a form of GUI in program environment Matlab. Video codecs used in compression schemes were implemented utilizing open source build of FFmpeg tool, which is described later.

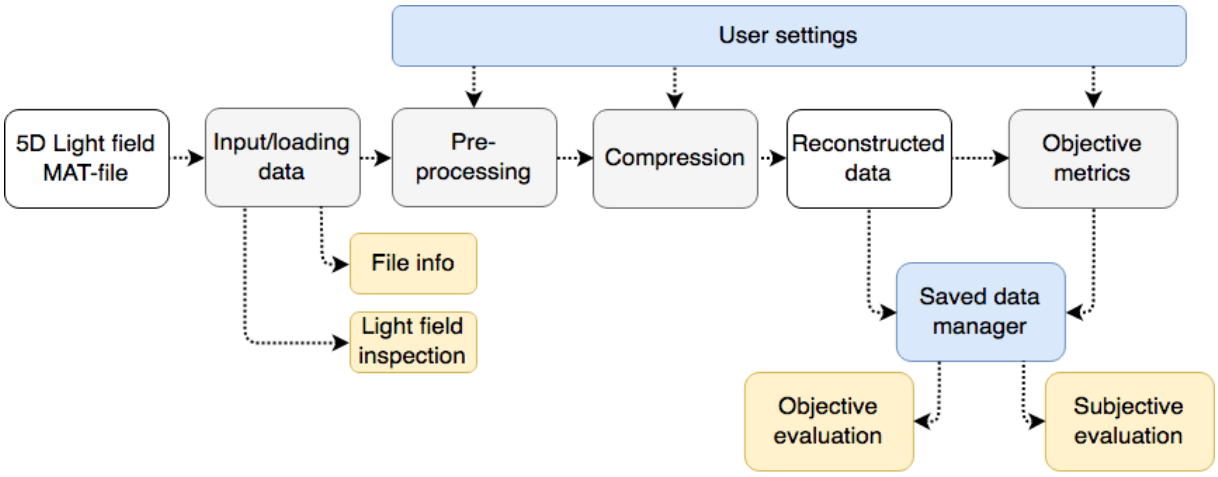

Figure 9.1: Block diagram of implemented compression tool.

9.1 Overview

All individual parts of the implementation are grouped within one GUI, where user is allowed to load light field data, apply compression schemes, modify compression parameters, use different light field data formats, bulk compress with various settings, compute objective metrics, save

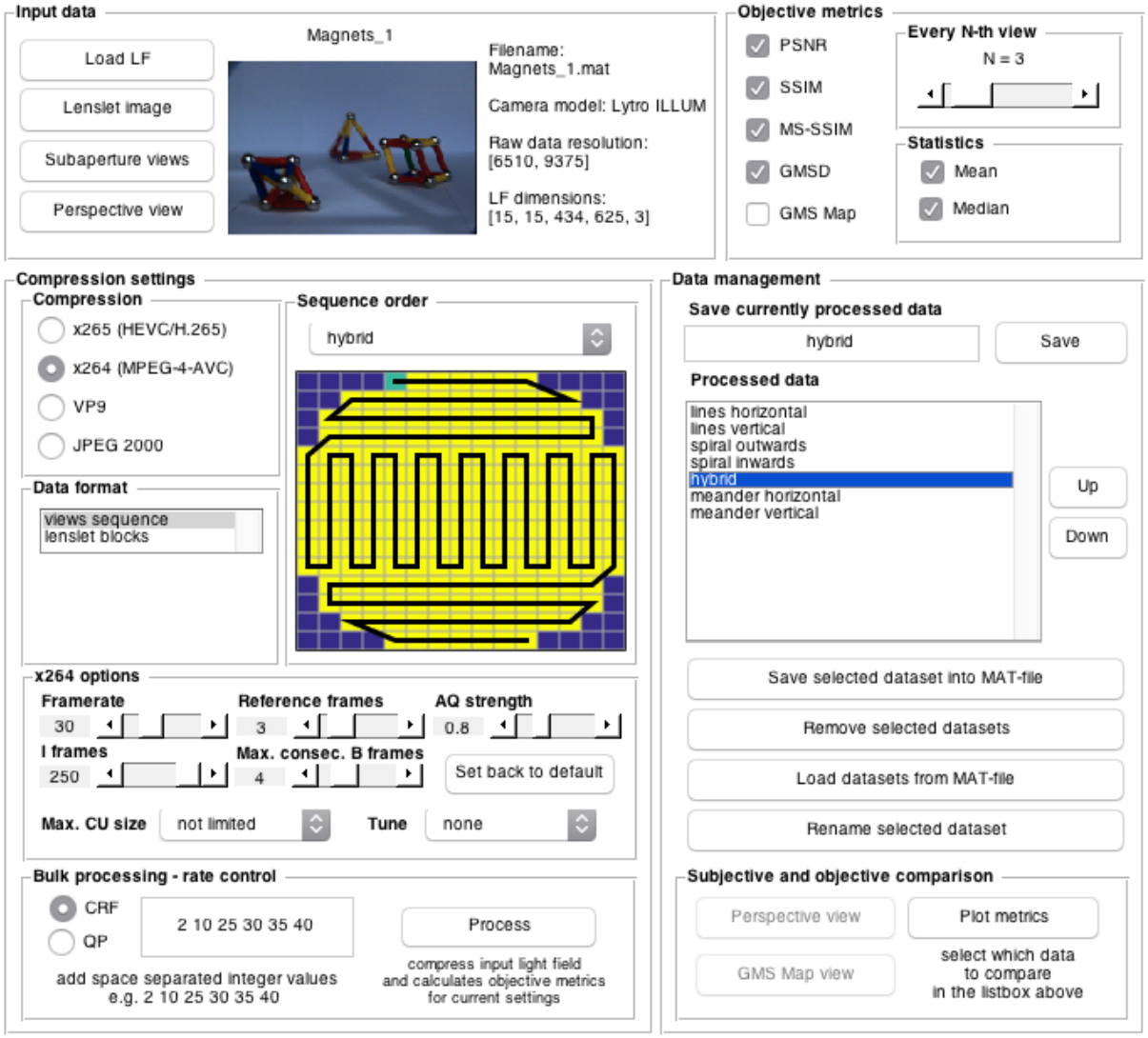

Figure 9.2: Graphical User Interface of compression tool

(load) previously compressed data, plot objective metrics and subjectively compare light field data. Generalized block diagram of implemented compression tool is shown in Figure 9.1. Structure of implemented GUI can be seen in Figure 9.2 and individual blocks/parts and its possibilities are described in following sections.

9.1.1 Input Data

The first step is to load light field data which are to be further processed. This tool was implemented to work with the datasets which appeared the most in current literature. EPFL dataset [30] which was used in Grand Challenges focused on Light Field Image Coding in 2016 and in 2017 by Institute of Electrical and Electronics Engineers (IEEE) (the challenge in 2017 was ongoing during the time this thesis was created). Most of the current light field datasets contain data captured by using the two consumer Lytro cameras. Lytro camera images, which are usually stored in LFP, LFR or RAW format, are transferred into 5D light field data format

using Light Field Toolbox, mentioned in chapter 5.3, and are stored in MAT-files (the file format used for storing data from Matlab environment). Each MAT-file contains the light field, camera metadata, calibration data, white images and also thumbnail image of center view of the particular light field.

Lytro F01 and Illum camera images were primarily tested during the implementation of this tool because the vast amount of freely available light field data is coming from these two cameras. Any photograph coming from those two cameras can be processed by Light Field Toolbox to obtain MAT-file containing mentioned 5D light field data which can be loaded into the implemented compression tool.

Lytro F01 (1st gen.) light field data is stored in the $9 \times 9 \times 381 \times 383 \times 4$ array as it contains 9×9 sub-aperture images/views each with resolution 383×381 and 4 channels (3 for colour components RGB and 1 for pixel confident weight). The size of MAT-file containing F01 light field data is around 50 MB. Lytro Illum (2nd gen.) light field data is stored in the same manner in $15 \times 15 \times 434 \times 625 \times 4$ array and the size of MAT-file containing Illum light field data is around 500 MB.

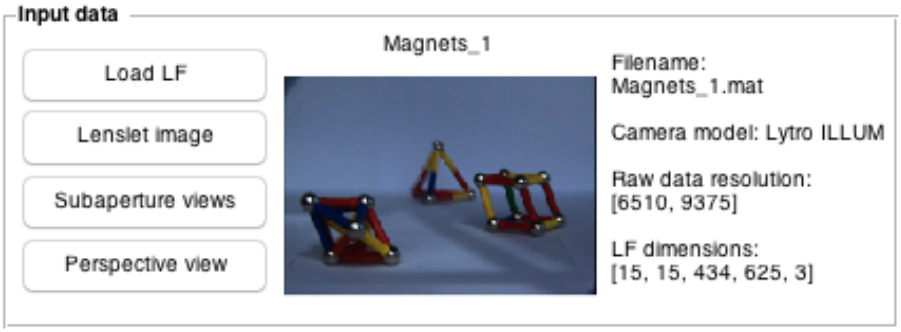

Figure 9.3: Graphical User Interface - data input

After the MAT-file with light field data is loaded, a thumbnail image of centre view and basic information about input light field file (camera type, dimensions of RAW lenslet data, light field dimensions) are displayed. Also several options how to inspect input data more closely are available - lenslet raw data ("Lenslet image"), video sequence showing all sub-aperture images in horizontal line scanning ("Subaperture views") or interactive perspective view ("Perspective view"), where user can navigate between individual sub-aperture images via mouse click and drag control.

9.1.2 Compression Possibilities

The main part of the implemented tool, compression scheme settings, allows the user to choose between several current state-of-the-art still and video image codecs and compression schemes which were proposed in the literature (combinations of different existing compression codecs and

different light field data formats to which the codecs are applied).
The user is able to use and analyse the performance of three video codecs (x265, x264 and VP9) and one still image codec (JPEG2000). All three mentioned video codecs were utilized through cross-platform tool FFmpeg [77]. FFmpeg is a free software project, that encapsulates libraries and tools for recording, converting and streaming of audio and video data. The tool is licensed under the GNU Lesser General Public License 2.1 (or later) and several other optional optimization parts are covered by GNU General Public License 2 (or later) [77]. All three codecs are covered by their codec libraries, which needs to be installed (libraries libx265, libx264 and libvpx). FFmpeg's individual tools are controlled from command line - thorough documentation can be found in [77].
Video codec x265 (HEVC) was selected because it was used in the majority of implementations presented in current literature dealing with pseudo-sequence-like coding of light field data ([60], [38], [61], [62], [39], [43], [40], [41], [42], [65], [67]) and it's predecessor, x264 (AVC), was also implemented in several proposed coding schemes ([59], [63], [64]).
VP9 codec was selected thanks to the fact of an easy extension (another FFmpeg library) to the previously selected video codecs. Another reason was the fact that, to my knowledge, the performance of VP9 coder has not been analysed on light field data yet. However, in [78] x265 was found to be significantly more efficient over VP9 in terms of natural image content, but VP9 can perform very similar to x265 when it comes to synthetic data. Also, computational times of VP9 are much higher when compared to x265 or x264 [79].

Compression Parameters

Several compression parameters are in place to be tuned to achieve the best possible quality. Two rate control modes, which are the same for all three video codecs, are used - Constant Rate Factor (CRF) and Quantization Parameter (QP). CRF is the default rate control mechanism for x264 and x265 encoders. This rate control mechanism keeps output quality at a certain level and varies QP between single frames where needed. This is allowed thanks to the motion between frames, where CRF applies higher QP to frames where motion appears and lower QP to frames with less or no motion. The second control mechanism, QP, simply applies the same quantization parameter to all frames. Benefits of CRF over QP are clear, especially in the scenes with fast movement. CRF can be set from 0 to 51, where 0 is perceivably lossless and 51 is the worst possible quality.

Several options can be modified for both x264 and x265 encoders (these options were selected so they can be set in both encoders, except framerate which can be set for VP9 as well):

- **Framerate** - framerate can be selected for all three video codecs.
  Range from 1 to 100 (integer).
- **I frames** - sets maximum period between Intra (key) frames in Group of Pictures (GOP). If this parameter equals 1, x265/x264 All Intra configuration is set. Intra frame acts as a

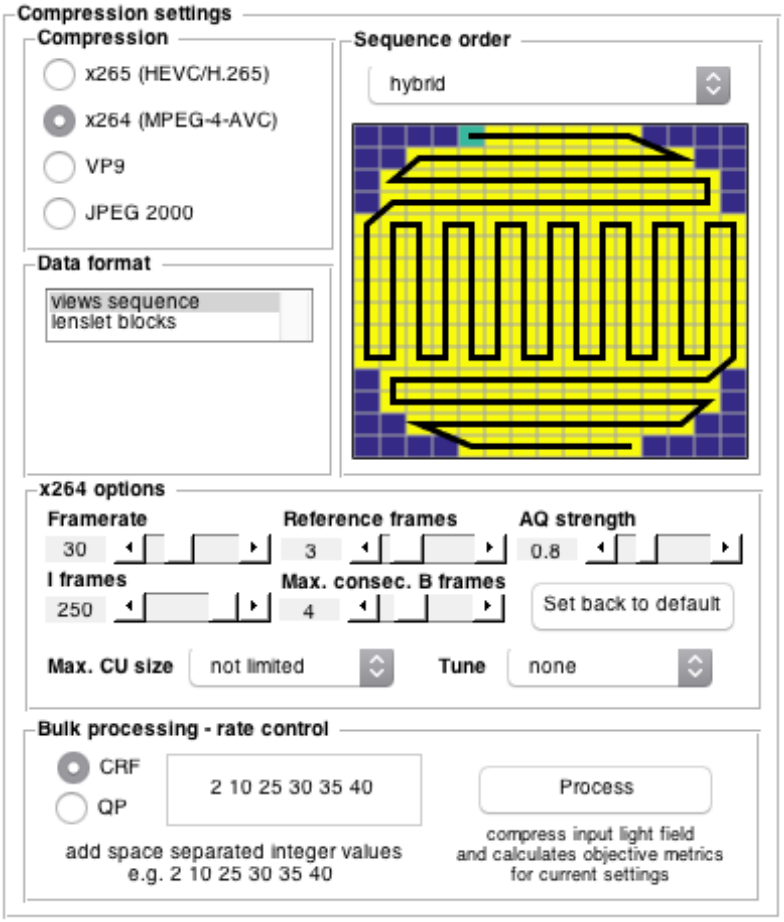

Figure 9.4: Graphical User Interface - compression settings panel

stream partition, no frame can reference to the frames from other side of its I frame. For regular video compression is it usually set to 1×FPS.

Range from 0 to 1000 (integer).

Default is 250 for x264 and x265.

- **Max. CU size** - sets maximum size of CU. Low number means more possibilities for parallelism. High numbers can encode flat large areas more efficiently. Faster presets of x265 usually use lower CU size.

  Range 16×16, 32×32, 64×64 (not limited) for x265. x264 only works with 16×16.

  Default is 16×16 for x264 and 64×64 for x265.

- **Reference frames** - sets maximum allowed number of L0 (List 0) past frame references. It means that size of Decoded Picture Buffer (DPB) can be controlled, i.e. the number of previous frames that P-frames can be referenced to [46]. Higher number increase the computational time, but it can reduce distortion and compression artefacts.

  Range from 1 to 16 (integer) for x264 and from 1 to 6 (integer) for x265.

  Default is 3 for x264 and x265.

- **Max. consecutive B frames** - sets maximum number of consecutive B frames in GOP. When set to 0, P/I frames are forced (low latency mode). This parameter also affects

computational time.

Range from 0 to 16 (integer).

Default is 3 for x264 and 4 for x265.

- **AQ strength** - sets the strength of Adaptive Quantization (AQ) offsets. Setting the AQ strength to 0 disables AQ completely. Higher values are taking more bits from complex areas (edges, structures) and reallocates them to simple (flat) areas to maintain detail there.

  Range from 0.0 to 3.0 (float).

  Default is 1.0 for x264 and x265.

Documentation of x264 and x265 encoder settings can be found in [50] and [53] respectively. All above mentioned parameters can be set back to default settings by clicking on button "Set back to default". Tune parameter is a group of other individual parameters set in a way that can be found useful when encoding the certain type of video data. For x264 the tuning parameter can be set to *film, animation, zerolatency, ssim, psnr, grain, stillimage and fastdecode*. For x265 the list is tighter - *zerolatency, ssim, psnr, grain, fastdecode*. Following list explains each tune's purpose:

- *film* - lower deblocking, used for high quality content
- *animation* - higher deblocking (for large flat areas), more reference frames
- *grain* - preserve grain structure in grainy material, tries to keep minimal QP fluctuation QP between frames
- *stillimage* - lower deblocking, for slideshow-like content, still images
- *psnr* and *ssim* - are used for codec debugging, basically disable all psycho-visual optimizations (optimizations which prioritize perceived visual quality before metric scores)
- *fastdecode* - disables several bottlenecks such as loop filters, weighted prediction and intra prediction in B frames for faster decoding on low computational power devices (also aimed for 4K with high bitrate)
- *zerolatency* - removes the latency both at encoder and decoder site, used for low latency streaming

Note that some of the preset tunes already includes some of the settings which can be set above the tune menu. Setting of any individual tune preset will for encoding overwrite previously selected settings, therefore it should be used carefully.

For a JPEG2000 compression only classic compression ratio and tile size can be set.

Light Field Data Formats

Light Field data comes in a 5D array, which can be reorganized into several data formats. For example, pseudo-sequence-like compression schemes are using separate sub-aperture views as an input sequence of frames into video codec and JPEG2000 can be for example applied to lenslet image (raw demosaiced data from imaging sensor). Another approach, which appeared

in literature, is to apply video compression to lenslet image which was partitioned into non-overlapping blocks. All these approaches are included in the options of this compression tool and user can experiment to find better combination.
Once the input light field is reorganized into individual 15×15 sub-aperture views (for Lytro Illum), JPEG2000 compression can be applied directly. However more interesting is to look at the sub-aperture views as a sequence of correlated frames (video sequence), therefore a pseudo-sequence. This pseudo-sequence can be compressed by using one of the used video codecs mentioned above. In order to increase compression efficiency, several preset order settings were implemented as can be seen in Figure 9.5. Input sequence of images can be reordered based on one of the selected order schemes and this reordered sequence is used as an input for video codec.

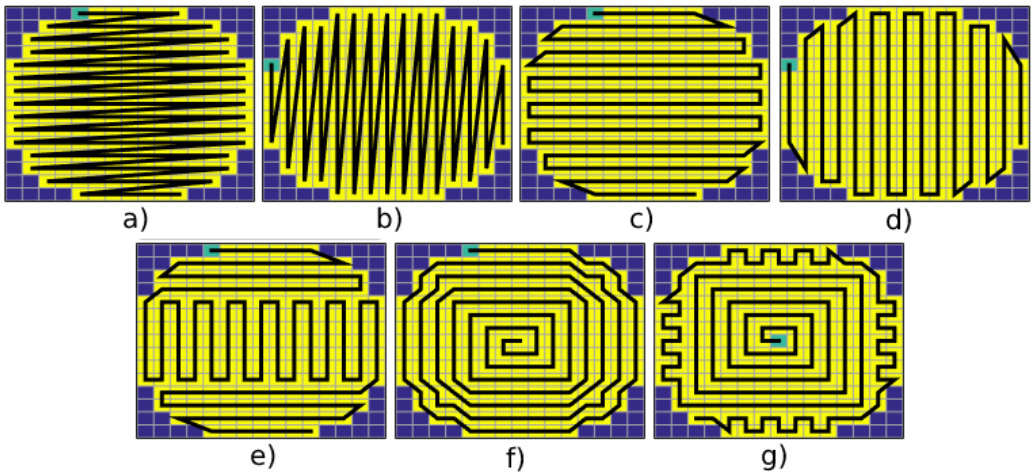

Figure 9.5: Input sequence reordering/scan possibilities for Lytro Illum camera - a) horizontal lines b) vertical lines, c) horizontal meander, d) vertical meander, e) hybrid, f) inward spiral, g) outward spiral.

Figures above are examples only applicable to Lytro Illum camera images, where corner views are completely black and are not included in the pseudo-sequence. After the compression, corner black images are automatically generated only to fill into the 5D array so the data can be used in interactive subjective analysis tool (which is described later in this chapter). Once this reordering was implemented, it could be easily applied for preprocessing of another light field data format - non-overlapping blocks of raw lenslet image. Same scan orders can be seen in Figure 9.6, when applied to square matrix of sub-aperture views of Lytro F01 camera.

Another possible light field data format is the lenslet image. Possible structures of light field data were described in 6. Whole lenslet image can be compressed by JPEG2000 compression as a regular still image and reorganized into the light field after decoding. This approach was used in [55], [59] and [38] in different forms. Here, JPEG200 was selected from still image codecs as it was proved to be the best solution among still image codecs [54] on individual sub-aperture images. Another approach is to partition lenslet image into non-overlapping blocks of required

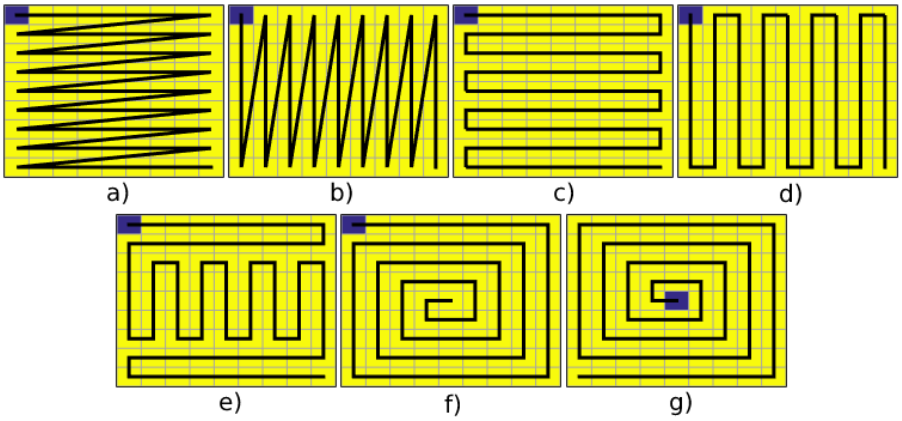

Figure 9.6: Input sequence reordering/scan possibilities for Lytro F01 camera or any rectangular shaped image matrix - a) horizontal lines b) vertical lines, c) horizontal meander, d) horizontal meander, e) hybrid, f) inward spiral, g) outward spiral.

sizes as was shown in [38]. Size of blocks can be chosen, but is automatically precomputed for the user to pick from. The dimensions are calculated so the overall resolution can be divided by selected block resolution to get integer number of blocks in row/columns (so there are no remaining blocks of various dimensions left). After the lenslet raw data is partitioned into blocks, same ordering schemes (as in Figure 9.6) can be applied. The number of available order settings is limited to the options *a, b, c* and *d* because these ordering schemes can be implemented easily for any possible dimensions of the matrix (partitioned blocks of whole lenslet image). During implementation, it was found that in this compression scheme scan order does not significantly influence this type of data format (lenslet blocks) therefore implementation of other ordering schemes was found to be unnecessary.

9.1.3 Objective Metrics

While input data is being compressed and decompressed with selected compression settings, objective metrics in form of PSNR, SSIM, MS-SSIM and GMSD can be calculated for Y component from YCbCr colourspace. First two metrics were selected because they are the most frequently used in literature, where some image processing algorithm performance is being analysed. To my knowledge, so far only few papers were devoted to objective and subjective quality assessment of light field data [69], [70], [72] where PSNR and SSIM were applied to YCbCr colourspace as objective metrics. This is probably due to their stabilized position in image processing objective assessment and the fact that no other objective metric targeted for light field data was developed yet. In [71] GMSD and MS-SSIM were applied and GMSD was found to perform well on light field data. In [70] it was shown, that objective and subjective metrics do not have to show the same trend for certain coding approaches. Description of each used metric can be found in chapter 8.

As can be seen in Figure 9.7 user has the option to choose the number N which says that

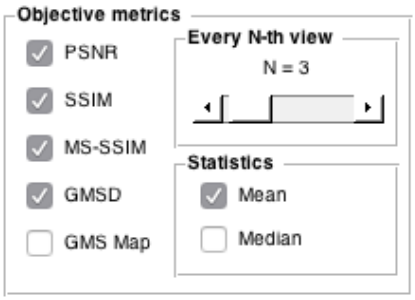

Figure 9.7: Graphical User Interface - objective metrics

chosen objective metric(s) will be calculated for each $N-th$ sub-aperture view (this was used mostly during implementation period to save computational time and still obtain some rough results). YCbCr colourspace is the most frequently selected colourspace in terms of objective assessment of light field coding techniques, primarily the Y component. It is easy to implement calculations in other colourspaces, but it was found to be unnecessary for the scope of this thesis. Calculated metrics are also included in MAT-file which can be saved after each session or can be compared with other compression schemes by plotting them in the data management section (which is described in the following section of this chapter).
Interesting objective metric for further analysis is GMSD and its Gradient Magnitude Similarity Deviation (GMS) Index Map (results of GMSD as well as other metrics are described in chapter 10). GMS Index Map can be computed together with objective metrics during bulk processing (GMSD is derived from GMS), but GMS needs to be calculated for all sub-aperture views so it can be interactively analysed later in "Data Management" section/panel.

9.1.4 Data Management

This area in GUI serves for data management. Currently, compressed data can be saved under the required tag or previously compressed datasets can be loaded. Datasets can be renamed and moved in the list - this may be useful for plotting the objective metrics as the name of each dataset also serves as its name in the legend plotted charts. Most importantly objective and subjective comparison of selected datasets is available within this section.

When the user clicks on "Plot metrics" all computed metrics for selected datasets are plotted in form of Rate-Distortion (R-D) charts. One set of graphs is plotted for mean average (if it was selected and computed) and one set for the median (again if selected and computed). Each set shows charts for PSNR, SSIM, MS-SSIM and GMSD (if selected and computed). This allows the user to immediately carry out an objective analysis of performed data compressions with different settings. All charts shown in chapter 10 are directly saved from charts plotted by using this tool.

Subjective comparison is available by clicking on "Perspective view", which opens new figure showing side-by-side comparison of input light field data with compressed light field data as

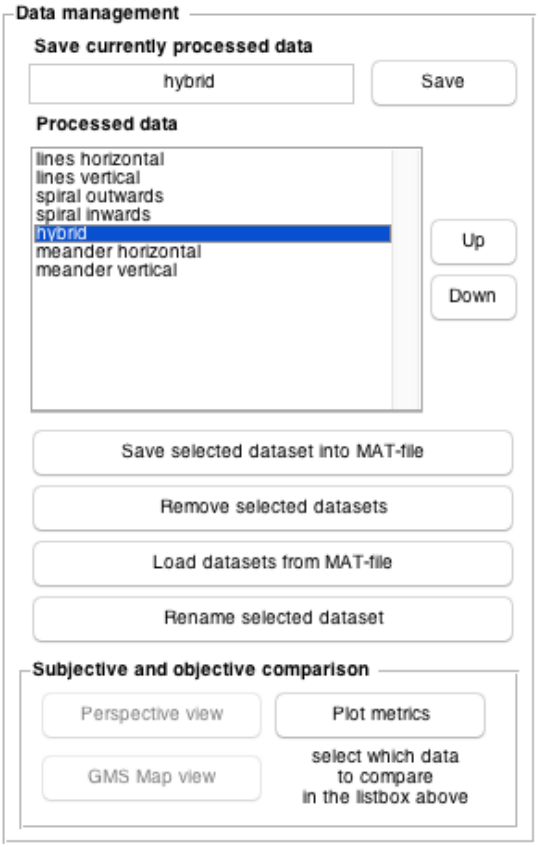

Figure 9.8: Graphical User Interface - data management panel.

shown in Figure 9.9. Initially, the data for first compression ratio/factor (first in bulk processing) is displayed, but the user has the option to navigate through the compression ratios which were selected for bulk processing. Also the centre sub-aperture image is shown initially, but same as in the input data section, the user can navigate through the sub-aperture views by mouse click and drag control over one of light field images. Navigation between both figures is linked, so by controlling the perspective in the left image, perspective on the right changes as well. Therefore the user can check light field images more easily for example for visible compression artefacts. The exact same controls are available, when the user clicks on "GMS Map View". Here, the user can change the perspective within sub-aperture views for compressed light field data in the right window and corresponding GMS Index Map is shown on the left side (again the navigation is enabled on both figures). GMS Index Maps were rearranged into light field data structure so it can be used in this interactive visualization. Example of this feature is shown in Figure 9.10

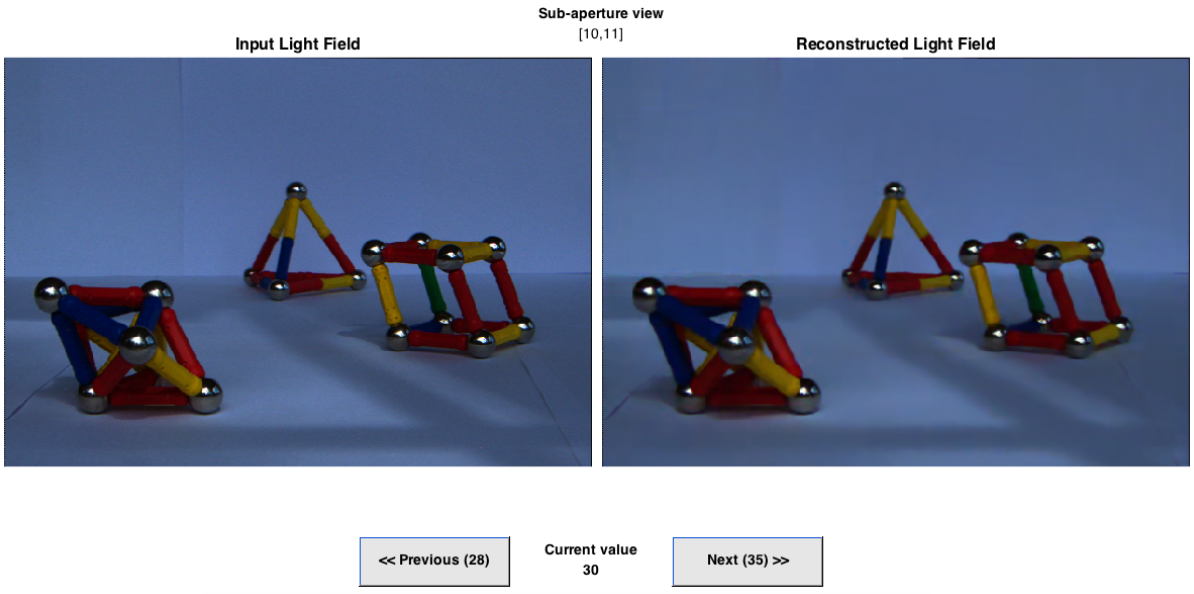

Figure 9.9: Graphical User Interface - interactive subjective comparison, where user can control the perspective.

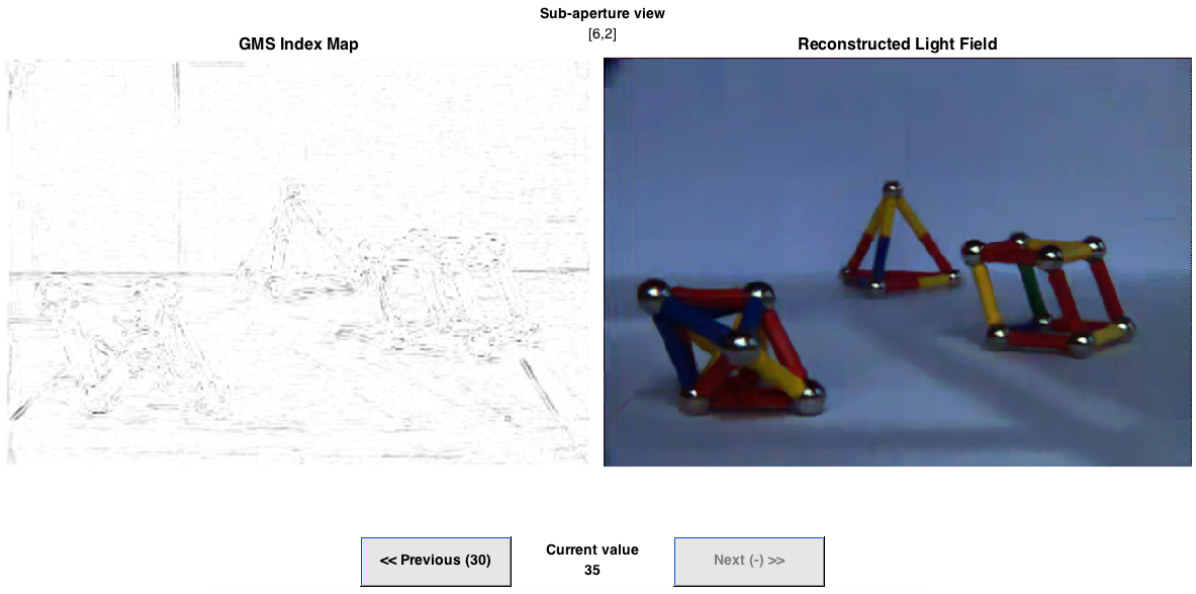

Figure 9.10: Graphical User Interface - interactive GMS Index Map analysis, where user can control the perspective together with corresponding GMS Index Map.

Chapter 10

Performance Analysis

In order to analyse performance of used compression techniques and also the effects of individual parameters, multiple tests were conducted using light field dataset. EPFL dataset [30] contains in total 118 images, which are divided into 10 different categories based on their content. Selection of 12 images (shown in Figure 10.1) from this dataset was used for performance analysis. Images were selected in a way that different content is represented and were also selected based on varying distance between the camera and the object(s) in captured scene.

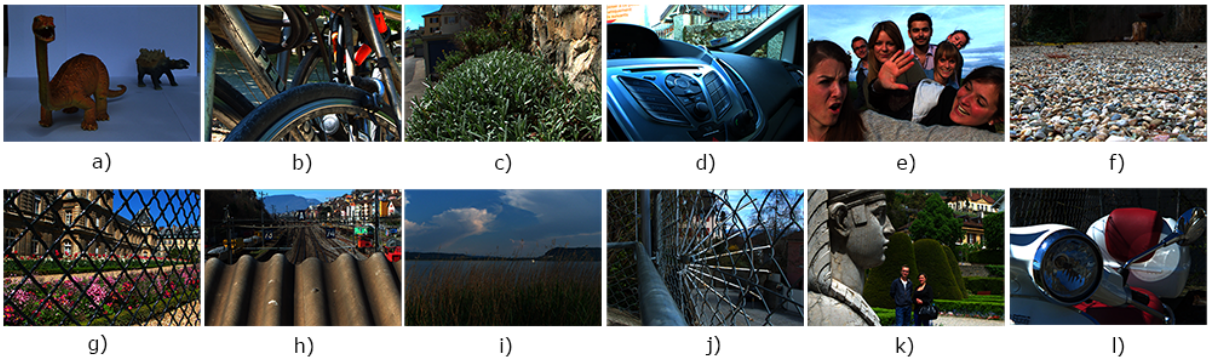

Figure 10.1: Subset of 12 images used for performance analysis - a) Ankylosaurus, b) Bikes, c) Bush, d) Car dashboard, e) Friends, f) Gravel Garden, g) Palais du Luxembourg, h) Railway lines, i) Reeds, j) Spear Fence, k) Sphynx, l) Vespa.

First, comparison of all included compression techniques for four different light field images can be seen in Figure 10.2, Figure 10.3, Figure 10.4 and Figure 10.5.

In Figure 10.2, JPEG2000 applied on sub-aperture views performed surprisingly well when compared to its performance on other light field images. The number of bits per pixel was computed as the file size of the final image divided by the total number of pixels for the used format of light field data. This may be caused by uniformly coloured background covering roughly half of the whole image which can increase the compression efficiency. Otherwise it performed badly as there is more structural content in all the other tested images.

As can be seen x264 slightly outperforms x265 in most of the tested light field images. The reason may be that x265 (as well as VP9) were designed to best suit for high and ultra high

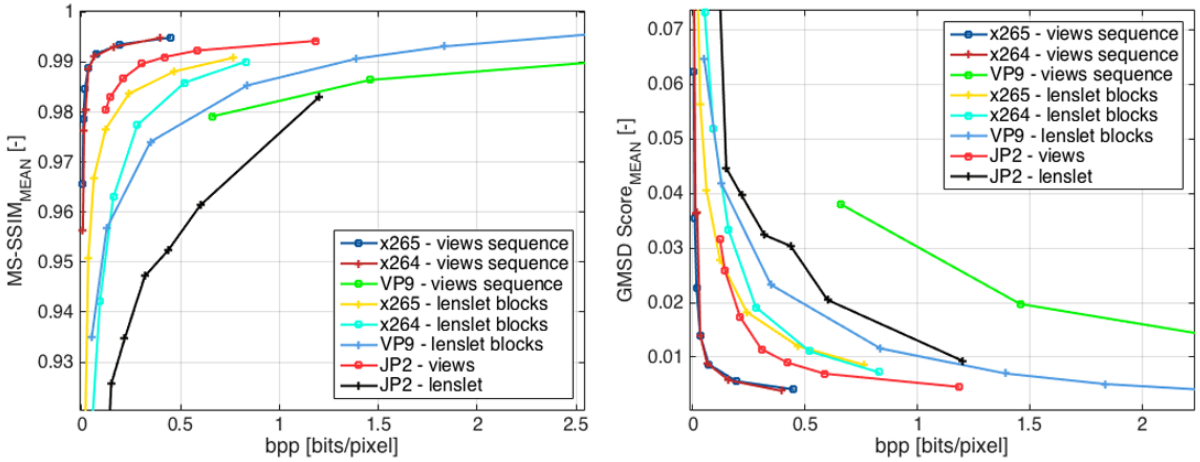

Figure 10.2: MS-SSIM and GMSD for all compression schemes - Ankylosaurus.

resolution content [80]. Resolution of one Lytro Illum view (one frame) is 625×434 pixels which is even below SD resolution. x265 is using variable size of coding blocks (16×16, 32×32, 64×64) based on the content and its predecessor x264 is using only 16×16 size of basic coding blocks. In [52] it was shown that forcing smaller size of coding blocks can change the overall quality and that using higher coding blocks is being more effective on high resolution content. Performance of x265 encoder with different sizes of basic CTU was also tested; however, there was zero difference between the obtained results.

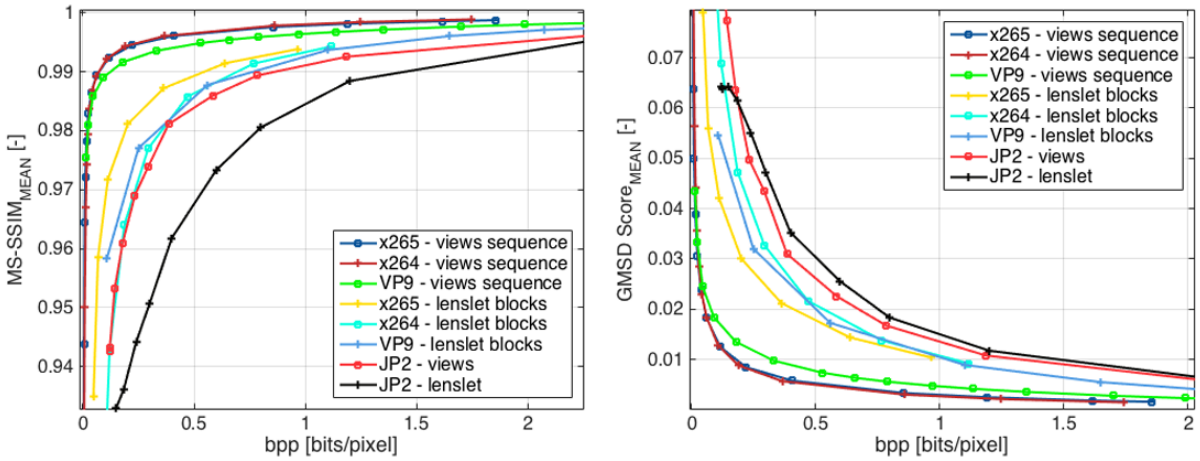

Figure 10.3: MS-SSIM and GMSD for all compression schemes - Friends.

It can be shown that x265 outperforms x264 at small bitrates (high compression ratios). This fact is more visible in the cases where data input was in form of lenslet blocks rather than individual views (x265 and x264 - lenslet blocks corresponds to yellow and cyan R-D curves respectively). Among video codecs, VP9 was outperformed in all cases and in some cases it was performing worse than still image codec. VP9 was found to be the most time expensive (approximately 10 times more than x265/x264), which can be caused by lower effectiveness of VP9 build in the used version of FFmpeg (3.3.1), but it was also shown in [79]. With one exception, JPEG2000 was found to perform better when applied on individual sub-aperture images rather

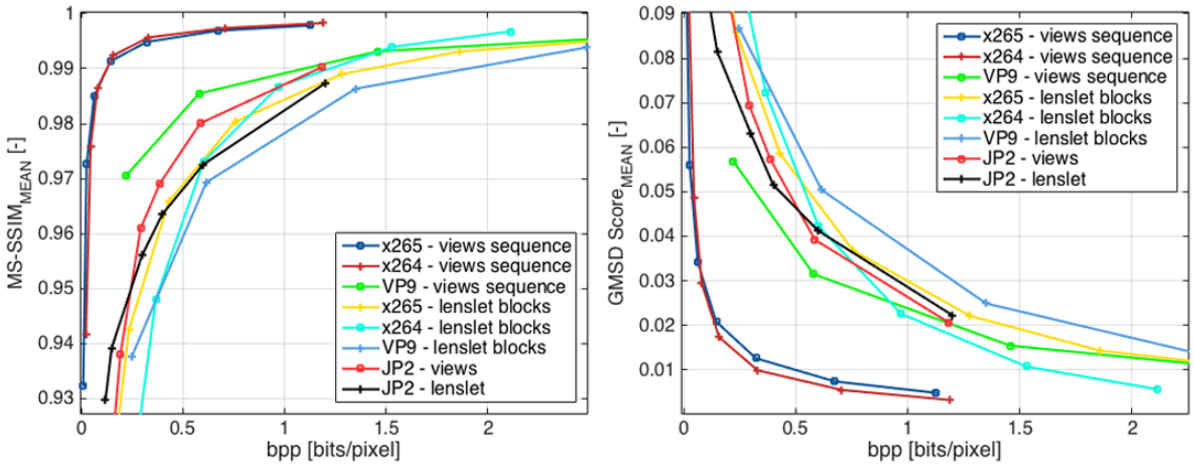

Figure 10.4: MS-SSIM and GMSD for all compression schemes - Gravel Garden.

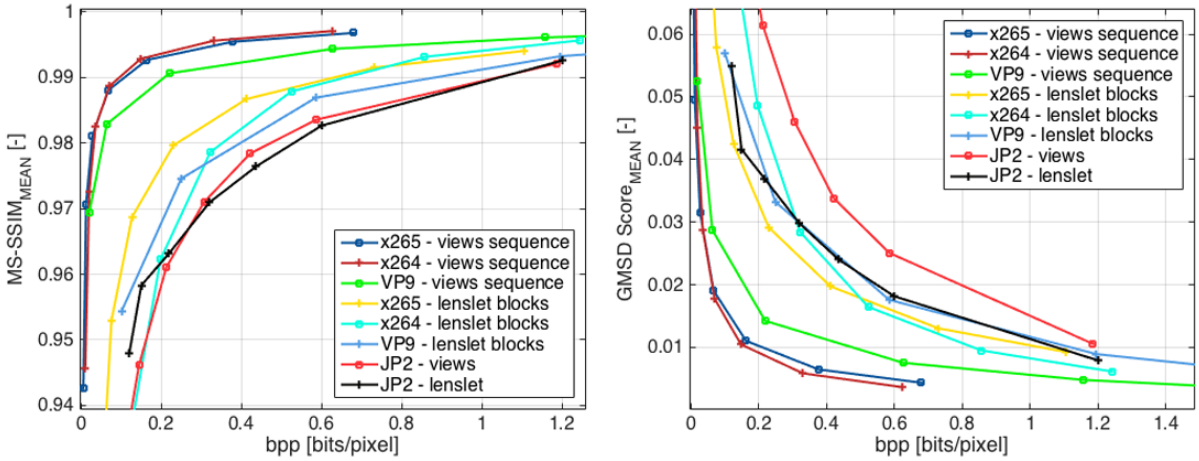

Figure 10.5: MS-SSIM and GMSD for all compression schemes - Railway.

than lenslet image. Mention worthy is the fact that JPEG2000 lenslet technique shows better results with GMSD than with PSNR, SSIM or MS-SSIM. JPEG2000 lenslet compression tends to produce compression artifacts as shown in Figure 10.6. These artifacts starts to be visible already when compression ratio is around 30, which corresponds to values around 0.8 bpp (file size of around 9 MB). The possible solution could be to use the algorithm to fill in the dark pixels in lenslet image.

10.1 Pseudo-sequences

As is mentioned in 7, pre-processed light field data into pseudo-sequence of individual views can be compressed using video codecs. Performance of state-of-the-art codecs and their settings on pre-processed light field data is evaluated in following subsections.

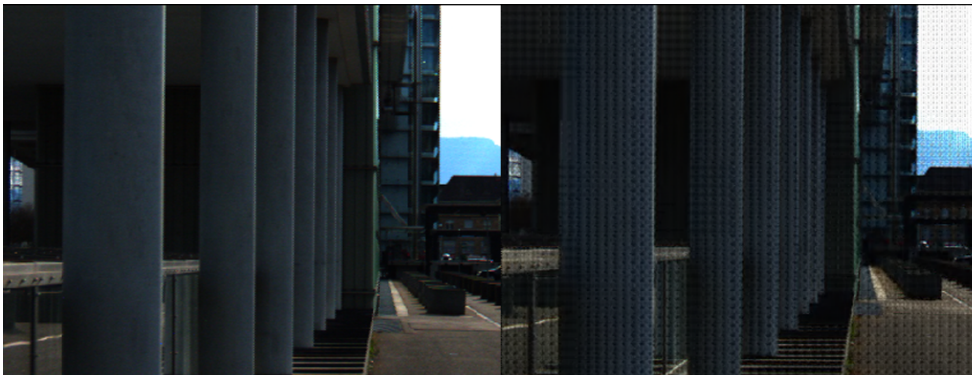

Figure 10.6: Illustration of JPEG2000 lenslet compression artifacts - Pillars (cropped).

	BD-PSNR [dB]													
	x265							x264						
	lines horizon.	lines vertical	spiral outwar.	spiral inwards	hybrid	meander horizon.	meander vertical	lines horizon.	lines vertical	spiral outwar.	spiral inwards	hybrid	meander horizon.	meander vertical
Ankylosaur.	0,00	0,12	**0,84**	0,42	0,64	0,37	0,50	0,00	0,25	0,29	**0,89**	0,79	0,47	0,63
Bikes	0,00	-0,08	**0,77**	0,15	0,71	0,50	0,43	0,00	0,04	0,56	0,38	**0,80**	0,59	0,29
Bush	0,00	-0,11	**0,53**	-0,30	0,44	0,35	0,20	0,00	-0,05	0,39	-0,02	**0,62**	0,33	0,26
Car dash.	0,00	-0,96	0,13	-0,56	-0,05	**0,25**	-0,65	0,00	-0,84	0,18	-0,16	0,22	**0,40**	-0,65
Friends	0,00	-0,20	0,29	-0,22	0,24	**0,35**	0,06	0,00	-0,27	0,43	0,27	**0,51**	0,44	0,29
Gravel G.	0,00	-1,05	0,20	-0,54	0,13	**0,31**	-0,42	0,00	-1,35	0,02	-0,38	-0,13	**0,26**	-0,90
Palais Lux.	0,00	0,04	0,64	-0,04	**0,66**	0,42	0,47	0,00	0,06	0,81	0,30	**0,92**	0,64	0,43
Railway	0,00	-0,38	**0,44**	-0,23	0,27	0,28	-0,20	0,00	-0,36	**0,44**	0,05	**0,44**	0,23	0,02
Reeds	0,00	0,01	0,16	-0,08	0,17	0,16	**0,24**	0,00	0,08	0,18	0,06	0,22	0,23	**0,29**
Spear Fen.	0,00	0,29	0,80	0,11	**0,85**	0,50	0,65	0,00	0,35	0,55	0,51	**1,08**	0,71	0,83
Sphynx	0,00	-0,18	**0,47**	-0,22	0,18	0,25	0,08	0,00	-0,51	0,01	-0,18	0,09	**0,24**	-0,28
Vespa	0,00	0,02	**0,97**	0,42	0,77	0,48	0,47	0,00	0,06	0,74	0,60	0,78	0,47	0,40

Table 10.1: Comparison of different ordering schemes - BD-PSNR - average difference between R-D curves. Scheme *lines horizontal* was used as a reference (0 values). Left - x265, right - x264

10.1.1 Sub-aperture views sequence

Array of sub-aperture images can be reordered into pseudo-sequence and encoded by video codec. Seven different ordering schemes which are shown in Figure 9.5, were evaluated on all twelve images using x265 and x264 video encoders. Differences between the individual R-D curves were evaluated using Bjøntegaard metric (BD-PSNR), which computes average difference between two R-D curves [81]. Experiment results for all twelve images and all schemes are shown in 10.1.

In overall, schemes *spiral outwards, hybrid* and *meander horizontal* performed better than other schemes. *Hybrid* consistently outperformed the rest of the schemes in case of x264 encoder and *spiral outwards* was consistently better for x265 encoder. Line scanning schemes and *spiral inwards* scheme performed badly for all tested images and for both encoders. All four lines and meander-like ordering schemes shows the same trend for both video encoders.

Because PSNR does not correspond well to how HVS perceives quality, MS-SSIM R-D curves (only for x265 encoder) were also evaluated using Bjøntegaard metric (Table 10.2).

In terms of MS-SSIM (Table 10.2), *hybrid* scheme outperformed other schemes on pictures g) Palais du Luxembourg and j) Spear fence (Figure 10.1), which both have similar content -

	BD-MSSSIM x10^3											
	Ankylo-saurus	Bikes	Bush	Car dash.	Friends	Gravel Garden	Palais du Lux.	Railway lines	Reeds	Spear Fence	Sphynx	Vespa
lines horizontal	0	0	0	0	0	0	0	0	0	0	0	0
lines vertical	0,43	-0,84	-0,64	-2,82	0,20	-6,08	2,24	-2,29	-1,71	3,46	-0,80	0,21
spiral outwards	4,12	6,74	7,25	2,01	2,74	4,00	8,36	4,91	1,45	9,25	5,38	7,89
spiral inwards	1,74	0,93	-2,79	-1,42	0,02	-3,76	-0,68	-0,44	-1,10	0,88	-1,40	2,65
hybrid	2,69	5,46	5,13	0,62	2,25	2,22	9,60	2,35	0,81	9,72	2,29	5,29
meander horizontal	1,71	3,94	4,04	1,34	2,37	2,90	6,83	2,07	1,02	6,00	2,22	3,30
meander vertical	2,22	3,19	2,78	-1,72	1,48	-1,65	7,37	-0,67	0,38	6,93	1,04	3,80

Table 10.2: Comparison of different ordering schemes - Bjøntegaard metric calculated for MS-SSIM R-D curves. Scheme *lines horizontal* was used as a reference (0 values).

object (fence) in the near distance from the camera lens. All further tests with pseudo-sequence of sub-aperture views are using *spiral outwards* ordering scheme if not mentioned otherwise.

In terms of preset tune settings for x264, it was found that tune *animation* outperformed all other tunes in all tested images. *Animation* tune, which is initially intended for compression of cartoons, is using more reference frames (B frames and L0 past reference frames), strength of deblocking filter is raised (less detail is preserved) and strength of AQ is lowered (more bits are allocated for complex areas). Results of different tune settings are shown in Figure 10.7.

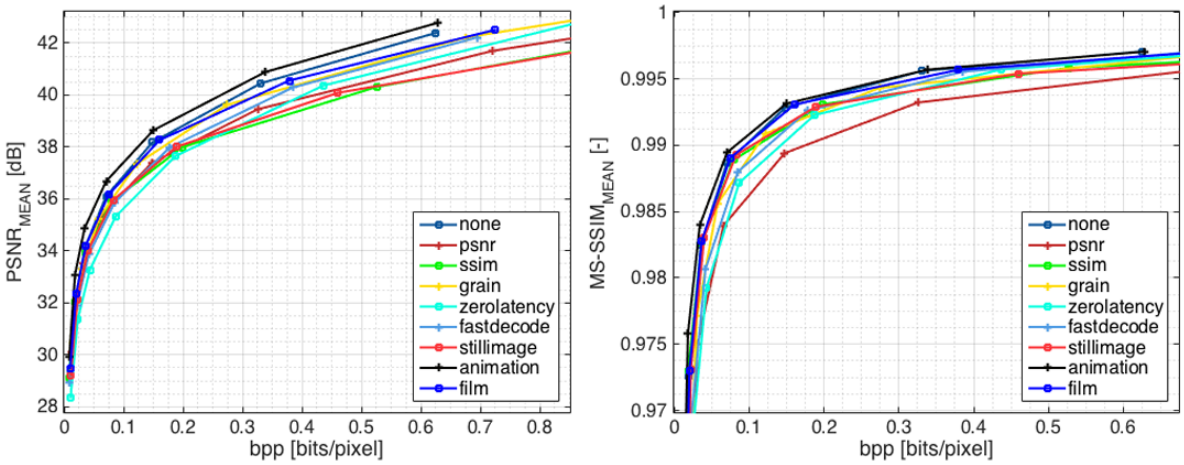

Figure 10.7: Different preset tunes for x264.

Based on this result, parameters such as number of used reference frames or AQ strength were evaluated more thoroughly. AQ strength can be set from 0 (turned off) to 3.0. The higher the number is, the more bits are allocated for compression of flat areas; therefore, bits are taken from areas with structural details. It was found that between 0.4 and 1.0 (default) performs the best for all objective metrics. The results can be seen in Figure A.1 in Appendix A. Demonstration of the AQ strength value effect on flat and detailed areas is shown in Figure A.2 in Appendix A.

Another tested parameter is the maximum number of reference frames. By default the

number is set to 3 for both encoders. As is shown in Figure 10.8, compression can achieve better results when maximum number of reference frames is set to 16. Measured computational time was within the same range for all frame reference settings of x264 encoder; however, this could be a subject for further testing. For x265 encoder only numbers within the range from 1 to 6 can be set. Nevertheless, it seems that this does not have any effect (or very little) on compression performance of x265 applied to *spiral outwards*-ordered pseudo-sequence of sub-aperture views. Also the computational time for x265 with maximum number of reference frames set to 6 was higher.

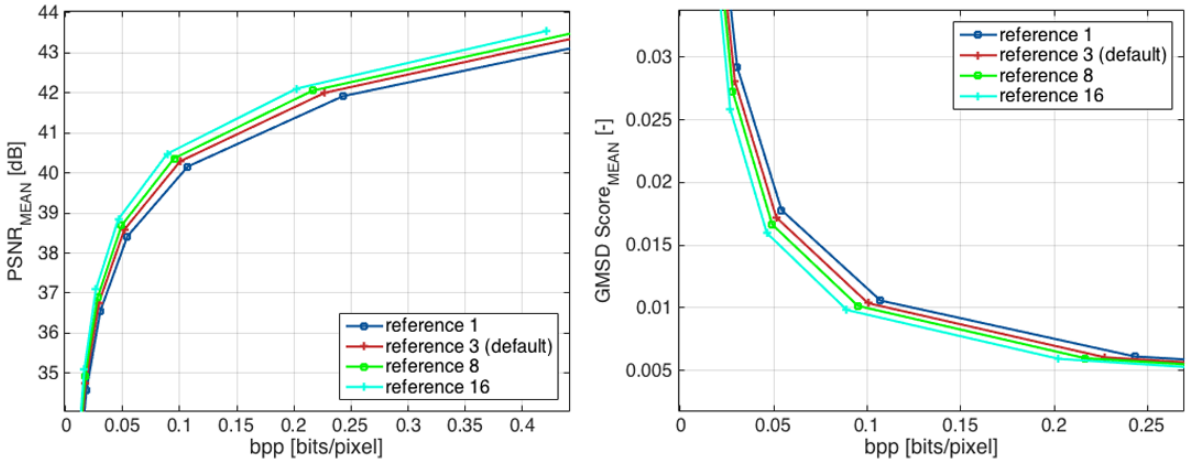

Figure 10.8: Performance of x264 encoder with different maximum number of reference frames. Left - PSNR, right - GMSD.

It is demonstrated that forcing more maximum consecutive B-frames can improve compression efficiency. Figure 10.9 shows that highest possible number of consecutive B-frames in x264 encoder settings is beneficial when encoding sub-aperture views as pseudo-sequence.

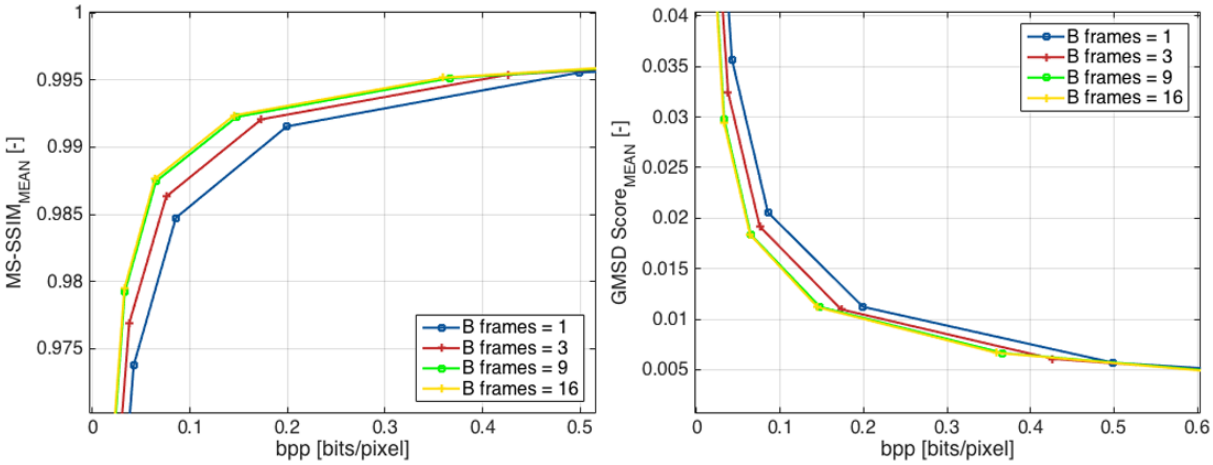

Figure 10.9: Performance of x264 encoder with different maximum number of B-frames. Left - MS-SSIM, right - GMSD.

Two previous results show that compression efficiency is increasing with higher number of reference frames. This is confirmed in Figure A.3 in Appendix A, where more I-frames were

forced to be used in compressed sequence. The number in legend indicates the period with which I-frame occurs. For x264 and x265 in FFmpeg this is, by default, set to 250. It is clear that forcing more I-frames is unnecessary as there are no significant changes between consecutive frames.

Comparison of x264 encoding with default settings against x264 with increased number of B-frames and reference frames can be seen in Figure A.4 in Appendix A. Computed BD-PSNR indicates improvement of 1.04 dB over the x264 encoder with default settings.

10.1.2 Lenslet block sequence

Another possibility to encode light field data as pseudo-sequence is to partition lenslet raw data into non-overlapping blocks of certain width and height, sort them and use as input to any of included video codecs. In Figure 10.10, partitioning of lenslet raw data into individual blocks can be seen (the tool offers tiling into $M \times N$ non-overlapping blocks as was described in 9.1.2). The sequence of blocks/tiles can be then sorted using several ordering schemes. Effect of block size and order of the blocks in sequence was analysed.

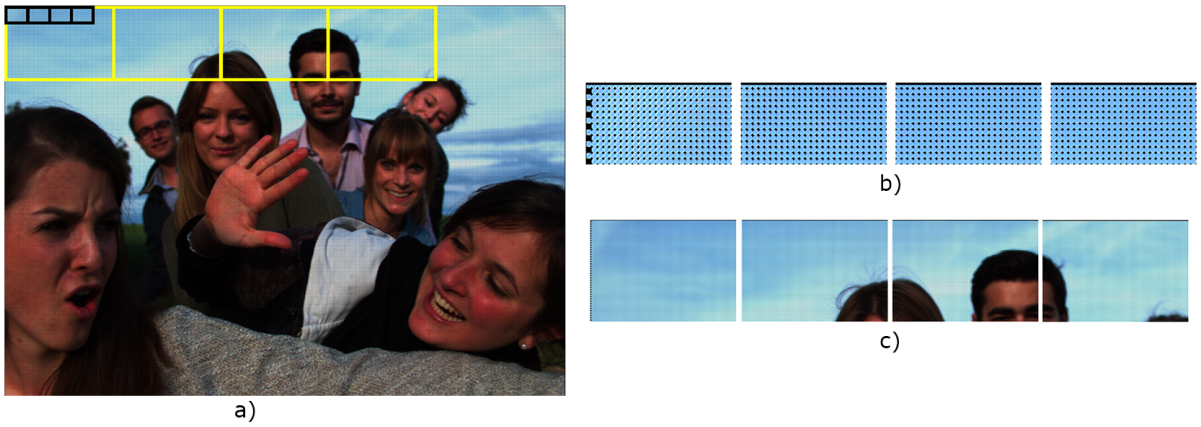

Figure 10.10: Example of lenslet image tiling into blocks - a) lenslet raw image with depicted partition into small (black) and larger (yellow) blocks, b) sequence of small blocks (tiling 31×25), c) sequence of larger (yellow) blocks (tiling 5×5).

Five different dimensions of blocks were tested using horizontal meander ordering. Lytro Illum lenslet image was tiled into 3×3, 5×5, 14×5, 31×15 and 31×25 blocks (number of blocks corresponds to block resolutions 2170×3125, 1302×1875, 465×1875, 625×210 and 375×210 pixels respectively). Using any video codec it was empirically found that tiling into smaller block sizes performs better over larger blocks. Results of only MS-SSIM and GMSD can be seen in Figure 10.11; however, all used objective metrics without difference demonstrated that smaller blocks outperforms larger blocks as was also found out in [38]. This fact can be explained and seen in Figure 10.10, where sequence of smaller blocks tend to be more correlated with contrast to larger blocks which usually have very different content in sequence, therefore the frame prediction is less exploited.

Effect of different ordering schemes was also evaluated by testing four basic ordering schemes

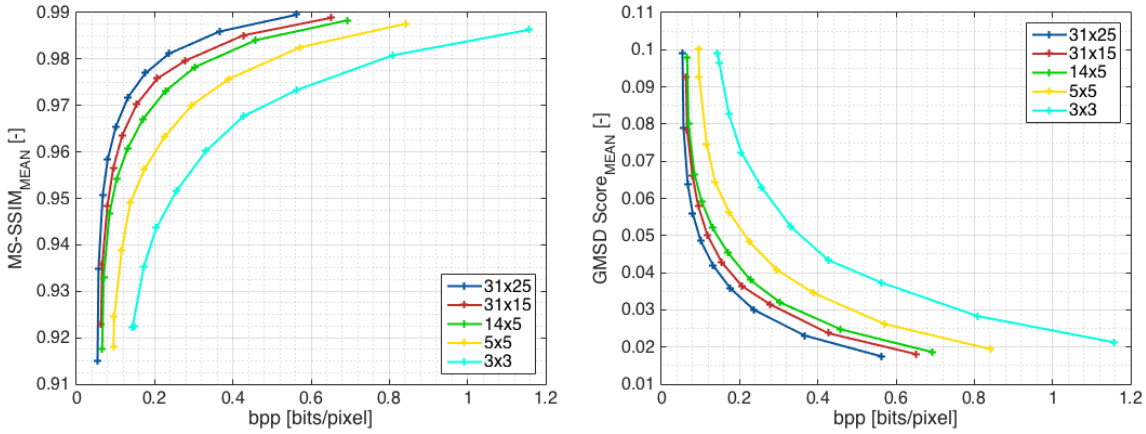

Figure 10.11: Charts - Performance of lenslet image partitioned into pseudo-sequence of blocks with varying size in terms of MS-SSIM and GMSD metrics.

- *horizontal* and *vertical lines*, *horizontal* and *vertical meander* (depicted in Figure 9.6). In this case differences between individual ordering schemes were smaller than between different block sizes. Figure 10.12 demonstrates that meander-like ordering schemes tend to perform little better over classic "line by line" ordering. This is due to the fact that there are bigger inter frames differences when sequence-neighbouring blocks (frames) are situated on the edges (left-right for horizontal ordering and top-bottom for vertical ordering) of the lenslet image. For tiling 31×25, the number of these edge transitions is not that high, 29 (for horizontal meander) when the overall number of blocks is 775.

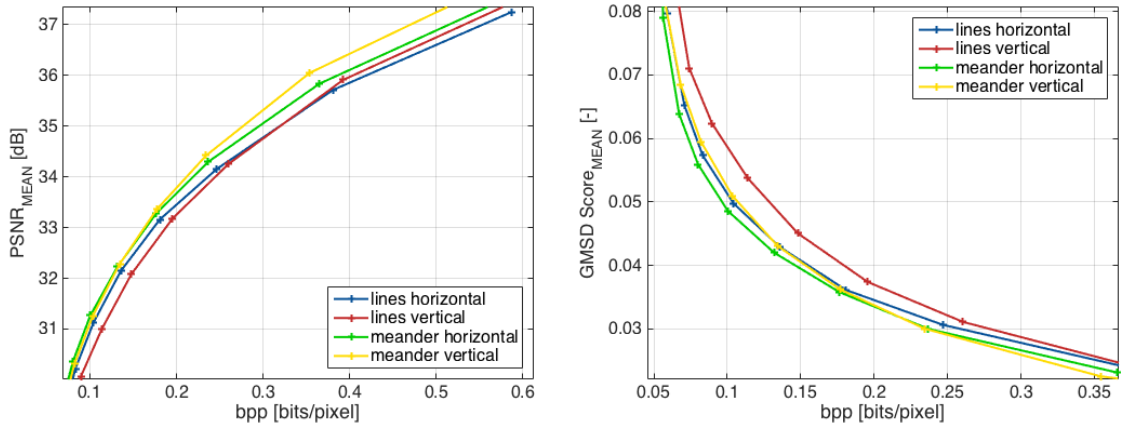

Figure 10.12: Charts - Performance of lenslet image partitioned into pseudo-sequence of blocks with varying ordering in terms of PSNR and GMSD metrics.

Same test was evaluated for block tiling 31×5 for vertical and horizontal meander ordering (1875×210 px). With these dimensions, lenslet image is practically sliced into thin wide blocks. This is depicted in Figure A.5 in Appendix A together with GMSD metric plotted. Vertically oriented meander-like ordering performs slightly better than horizontally oriented. That because there is also more correlation between the slices when they are ordered in columns rather then in rows. Performance of this kind of pre-processing is definitely content dependent, but it is

clear that smaller blocks should for common images perform better over larger blocks.

Chapter 11

Conclusion

This thesis brings an insight into the novel light field technology with main focus on light field data compression. Chapters two till eight are dedicated to the theoretical part, which goal is to bring an overview of light field data capturing, representation, processing, compression and objective and subjective quality evaluation. In second chapter the fundamentals are described in order to understand the process of light field data acquisition. Third chapter is dedicated to explanation of different technologies used for obtaining the light field data. Light field technology allows to capture the scene with a single camera, single lens system, single exposure. Then in post-processing one is able to achieve depth map, different points of view, refocus or perform object manipulation within the scene. All these features are attractive for research, professional applications as well as for regular consumers.

The current light field technology situation is described focusing mainly on consumer devices for light field data acquisition and reconstruction. The main drawbacks of the consumer plenoptic cameras are considered to be the already established market of consumer cameras and the resolution of the final image. That is because of the ease and quality of photos from smartphone cameras is increasing and the conventional photographers still prefer regular DSLRs with high-resolution images rather than the option to focus after taking a photo with SD resolution.

The fifth chapter brings an overview of available software tools for processing and management of Lytro camera files. Nowadays, there is one official software tool, produced directly by Lytro Inc., which is consumer-oriented and offers basic data manipulation. Several free and open platform tools were developed for more advanced processing and manipulation with light field data captured not only with Lytro cameras. The sixth chapter describes more in detail the structure and possible representations of plenoptic data while focusing on Lytro data files. Light field data captured as a raw lenslet demosaiced image can be reorganized into different data formats, which may be useful for different applications and representations.

Chapter seven summarizes several state-of-the-art compression schemes, which are used for light field data encoding. Moreover, three groups of coding techniques are presented with the description of some compression schemes that were proposed in the current literature. The attention is on pseudo-sequence coding mechanisms, which usually exploits already developed

video codecs. It was found that light field data can be effectively compressed with existing video codecs after some data preprocessing or by codec modifications. However, deeper research still needs to be done in this direction.

Next chapter explains subjective and objective quality assessment methods, which are used for evaluation of light field data processing algorithms. Usually only common objective metrics for digital image quality assessment are used for evaluation of reconstructed light field data. In one case, a more ad hoc interactive subjective test was performed for quality assessment. However, at this time there is no standard or recommendation for subjective nor objective quality assessment of plenoptic-data-processing techniques.

Chapters nine and ten are dedicated to processing of theory into practical implementation. The goal of the practical part is to implement a GUI, which enables to apply compression schemes to pre-processed light field data and allows to objectively and subjectively compare different compression approaches. Another goal is to perform objective quality assessment of light field data reconstructed after various compression schemes and its settings. Chapter nine can be regarded as a form of documentation for implemented compression tool. Implemented GUI allows to compress light field data with different compression schemes, with varying pre-processing methods and compression settings. Reconstructed light field data can be objectively and subjectively evaluated within the same interface. The implemented tool allows user to effortlessly adjust the compression settings and analyse obtained results.

The last chapter is dedicated to performance assessment of the individual compression scheme implementations. First, general evaluation of compression approaches is performed. Existing video codecs x264 and x265 applied on pseudo-sequence are found to be the two most effective approaches within the implemented tool. Moreover, their performance is content-consistent, which is not always the case with the other, low-performing, compression schemes. It is demonstrated that pre-processing of light field data into reordered pseudo-sequence of sub-aperture images can increase the encoding performance. Another demonstrated approach on how to enhance compression performance is to adjust the encoder settings to be more suited for light field data. It is shown that current, state-of-the-art encoders can be efficiently used for light field data compression, nevertheless, there is still a room for improvement and further research.

Appendix A

Additional charts and examples archive

Charts and examples in this appendix are referred to from chapter 10, but are placed here due to space saving.

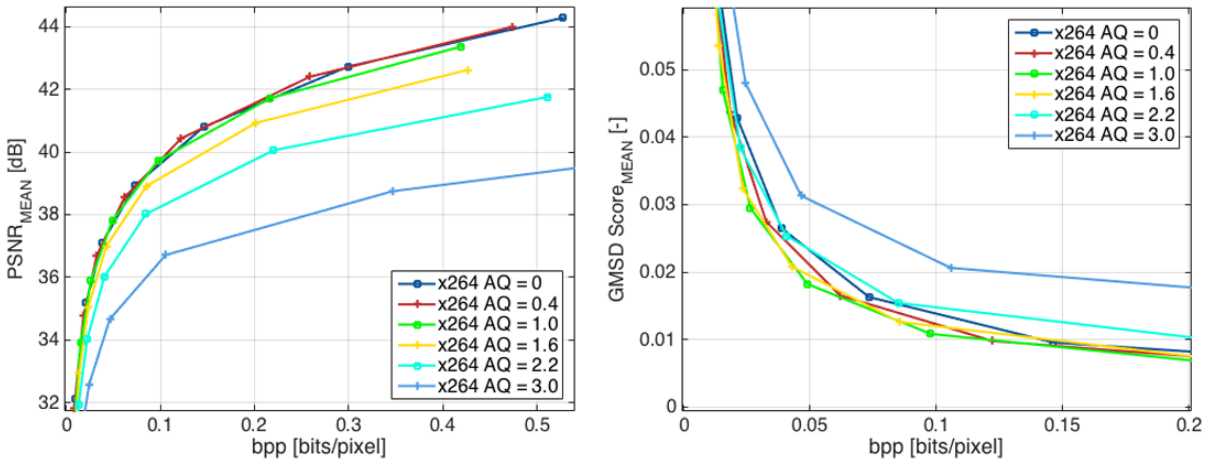

Figure A.1: Performance of x264 encoder with different AQ strength values. Left - PSNR, right - GMSD.

Figure A.2: Example of flat and detailed areas after processing with x264 with different AQ strength. Left - AQ = 0.4, right - AQ = 3.

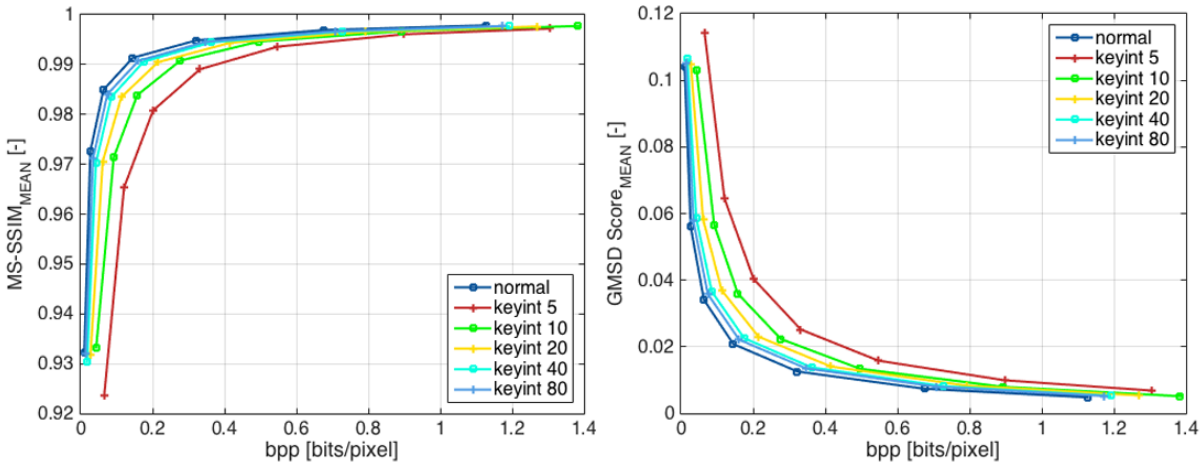

Figure A.3: Performance of x265 encoder with different number of I-frames in the pseudo-sequence. The number indicates the period of I-frame and sequence contains 193 frames. Left - MS-SSIM, right - GMSD.

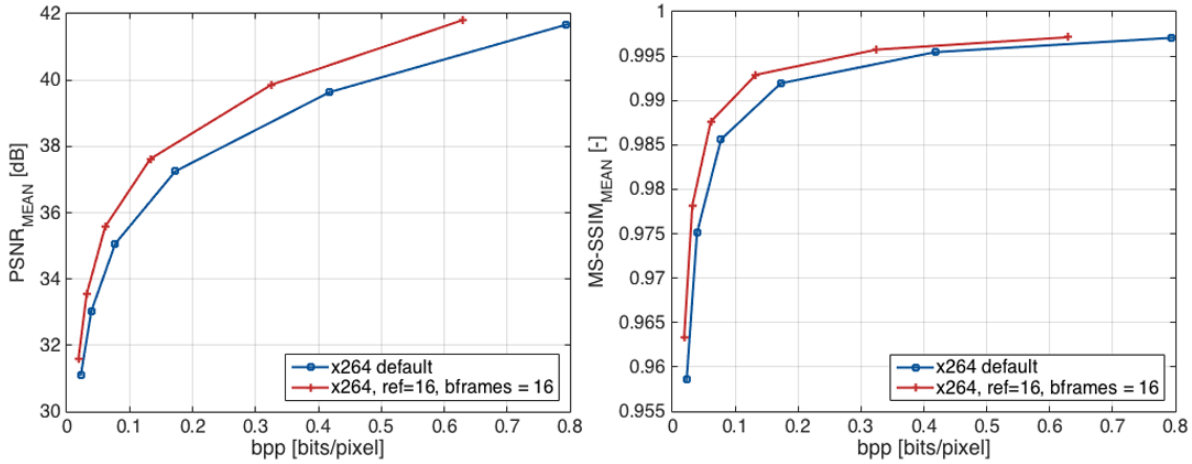

Figure A.4: Performance of x264 encoder with default settings against adjusted numbers of reference frames. BD-PSNR = 1.04 dB (the adjustments show improvement over the default settings). Left - PSNR, right - MS-SSIM.

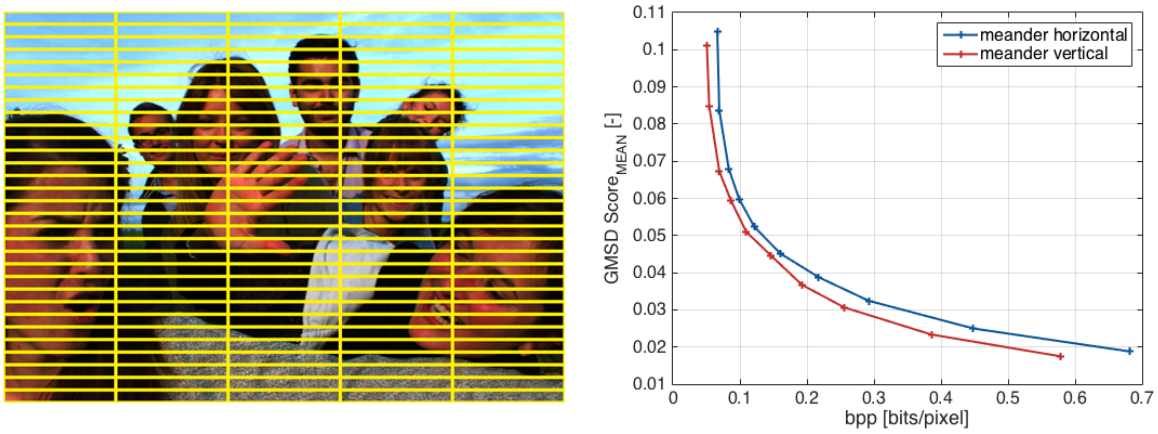

Figure A.5: Left - partitioning of lenslet image into thin blocks, right - performance of horizontal and vertical meander ordering in terms of GMSD metric.

Appendix B

Structure of folders in appendix archive

/Implementation/... Folder containing all the functions needed for functional running of implemented compression tool. Individual functions are described in Appendix C. Folder also contains README file explaining files in this folder and how to use them.

/Measurements/... Folder containing MAT-files with some of the performed tests from chapter 10. MAT-files contain the information about current settings, bitrate and objective metrics. Compressed and input light field data were excluded for limited storage purposes. Folder also contains README file explaining each sub-folder, its files and how to use them.

Appendix C

Overview of functions in Matlab implementation

calcMetrics.m Calculates selected objective metrics for reconstructed light field data.

compressFFMPEG.m Compress input light field data using any video codec from FFmpeg tool. Controls FFmpeg pipeline using the command line.

compressJP2.m Function that is using Matlab implementation of JPEG2000 compression based on compression parameters.

compressTool.m GUI file that encapsulates all functions into one functional box.

compressTool.fig Figure file for compressTool.m.

GMSD.m Function that calculates Gradient Magnitude Similarity Deviation - implementation used from [76].

LFDispMousePan.m Function for interactive displaying light field data, which allows a user to change perspective using click-and-drag mouse controls. Implementation used from LF Toolbox v0.4 [29].

LFDispSetup.m Gelper function for LFDispMousePan.m used for setting up light field display. Implementation used from LF Toolbox v0.4 [29].

LFDispMousePan2.m Function for interactive displaying two light field data images at the same time. Core code used from LF Toolbox v0.4 [29]. Improved to show two light field data images at the same time (or GMS map) and to navigate through more sequence of differently compressed light field data (with different compression ratio etc.).

LFDispSetup2.m Helper function for LFDispMousePan2.m used for setting up light field display. Core code used from LF Toolbox v0.4 [29]. Modified for the needs of function LFDispMousePan2.m.

msssim.m Function that calculates Multi Scale-Structural Similarity Index - implementation used from [75].

ssim_index_new.m Helper function for msssim.m.

plotMetrics.m Helper function used for plotting gathered objective metrics.

processLF.m Loads new MAT-file which contains light field data, process LF data into different data formats.

reorder.m Helper function that is reordering light field data sequences accordingly to wanted order. Adding/removing dark corner images.

scanSequence.m Displaying function that reorders input light field data into video sequence which can be played as video.

ssimOpt.m Calculates SSIM. Implementation used from [73] is more computationally efficient than Matlab implementation of SSIM.

Appendix D

Implementation README file

This README file was created as instructions for compression tool, which was implemented as a part of master's thesis.

Title:	Methods for plenoptic image data processing
Author:	Jan Svihalek
Year:	2017/18
University:	Czech Technical University in Prague
Faculty:	Faculty of Electrical Engineering
Dept.:	Department of Radioelectronics
Study program:	Communications, Multimedia, Electronics
Branch of study:	Multimedia Technology

README before running Compression Tool
1) Please make sure you are running Matlab R2015a or at least newer version (it cannot be guaranteed that all scripts will work perfectly using older or newer versions).

2) All Matlab functions needs to be added in one folder, which has to be included in Matlab working paths.

3) Please make sure that the same folder (folder in which Matlab will be running while using this GUI) also contains FFmpeg build (version 3.3.x or newer) which includes x264, x265 and VP9 codecs.
- More information on how to start and enable FFmpeg tool can be found here http://ffmpeg.org/download.html.
- Without working FFmpeg build you will be limited to JPEG 2000 compression or only to viewing already measured data.

4) Please make sure that you have at least one MAT-file containing LF, otherwise you will not be able to explore the compression tool.
- You can download light field images (or whole dataset) on following website: https://mmspg.epfl.ch/EPFL-light-field-image-dataset
- You can download all image files from the following FTP (please use dedicated FTP clients, such as FileZilla or FireFTP):
  FTP address: ftp://tremplin.epfl.ch
  User name: LytroIllum@grebvm2.epfl.ch
  Password: 48HMd6tm4SxC6s3z
  a) After connecting to the FTP server, go to "4D_LF" folder
  b) Pick one dataset and download it to your computer
  c) Unzip after downloading
  d) Move at least one MAT-file into Matlab folder (for easier manipulation)

4) Run compression tool, by typing compressTool into Matlab Command Window. After that, Compression Tool GUI will show up.

5) Press Load LF to load input light field. Please note that button "Load datasets from MAT-file" is for loading already processed LF files only (with measured data, compressed LF etc.).

6) After successfully loading LF file, you will be able to use the GUI without any constraints.

Shown README file serves as brief instructions for users. It is also included in the attached appendix folder *functions* in *.txt and *.pdf file formats. In these versions it also contains description of individual functions which is described in C.

Appendix E

Comparison of Lytro F01 and Illum cameras

Table E.1: Specification comparison of Lytro 1st and 2nd generation cameras

	Lytro F01 (1st gen.)	**Lytro Illum (2nd gen.)**
Optics		
Focal length	43 - 344mm	30 - 250 mm
Zoom	8×	8×
Aperture	Constant f/2.0	Constant f/2.0
Image sensor		
Type	CMOS	CMOS
Light field resolution	11 megaray	40 megaray
Active area	(4.6 × 4.6) mm	(10.82 × 7.52) mm
Image		
Format	.lfp (Light Field Picture)	.lfp or .lfr (Light Field Raw)
Size ratio	1:1	3:2
2D resolution	1080 × 1080	2450 × 1634
File size	approx. 16 MB	approx. 20 MB (.lfp), 50 MB (.lfr)
Others		
Size	41 mm ×41 mm ×112 mm	86 mm ×145 mm ×166 mm
Weight	214 g	940 g
Release price	400 USD (8 GB version), 500 USD (16 GB version)	1600 USD

Bibliography

[1] H. Maître, *From photon to pixel: The digital camera handbook*, 1st ed., ser. Digital signal and image processing series. Wiley - ISTE, 2015, ISBN: 978-1-84821-847-5,206-207-211-2,9781119238447,1119238447,9781119238638,1119238633,1848218478.

[2] Ren Ng, "Digital light field photography", PhD thesis, Stanford, CA, USA, 2006, ISBN: 978-0-542-70779-7.

[3] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision", in *Computational Models of Visual Processing*, MIT Press, 1991, pp. 3–20.

[4] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera", 2, vol. 14, Washington, DC, USA: IEEE Computer Society, Feb. 1992, pp. 99–106. DOI: 10.1109/34.121783. [Online]. Available: http://dx.doi.org/10.1109/34.121783.

[5] M. Cohen, S. J. Gortler, R. Szeliski, R. Grzeszczuk, and R. Szeliski, "The lumigraph", Association for Computing Machinery, Inc., 1996. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-lumigraph/.

[6] M. Levoy and P. Hanrahan, "Light Field Rendering", in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, Association for Computing Machinery (ACM), 1996. DOI: 10.1145/237170.237199. [Online]. Available: https://doi.org/10.1145%2F237170.237199.

[7] Ren Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-held Plenoptic Camera", Apr. 2005, pp. 1–11. [Online]. Available: https://classes.soe.ucsc.edu/cmps290b/Fall05/readings/lfcamera-150dpi.pdf.

[8] G. Lawton, "3D displays without glasses: Coming to a screen near you", *Computer*, vol. 44, no. 1, pp. 17–19, 2011, ISSN: 0018-9162. DOI: 10.1109/MC.2011.3.

[9] F. Ives, *Parallax stereogram and process of making same.* US Patent 725,567, Apr. 1903. [Online]. Available: https://www.google.com/patents/US725567.

[10] G. Lippmann, "Épreuves réversibles donnant la sensation du relief", *J. PHYS. THEOR. APPL.*, vol. 7, no. 1, pp. 821–825, 1908. DOI: 10.1051/jphystap:019080070082100. [Online]. Available: https://hal.archives-ouvertes.fr/jpa-00241406.

[11] 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 Trans. Graph.*, vol. 24, no. 3, pp. 765–776, Jul. 2005, ISSN: 0730-0301. DOI: 10.1145/1073204.1073259. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073259.

[12] T. G. Georgiev and A. Lumsdaine, "Resolution in plenoptic cameras", in *Frontiers in Optics 2009/Laser Science XXV/Fall 2009 OSA Optics & Photonics Technical Digest*, Optical Society of America, 2009. DOI: 10.1364/COSI.2009.CTuB3. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=COSI-2009-CTuB3.

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

[14] T. G. Georgiev, "New results on the plenoptic 2.0 camera (invited paper)", 2008.

[15] T. G. Georgeiv, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala, "Spatio-angular resolution tradeoff in integral photography", in *In Eurographics Symposium on Rendering*, 2006, pp. 263–272.

[16] B. Duvall, *About Lytro*, https://www.lytro.com/about.

[17] Lytro Inc., *Lytro Illum user manual*, 2014. [Online]. Available: https://s3.amazonaws.com/lytro-corp-assets/manuals/english/illum_user_manual.pdf.

[18] ——, *Lytro Lytro user manual*, 2012. [Online]. Available: https://www.lytro.com/press/releases/lytro-brings-revolutionary-light-field-technology-to-film-and-tv-production-with-lytro-cinema.

[19] Lytro Inc., *Press release: Lytro brings revolutionary light field technology to film and tv production with lytro cinema*, Apr. 2016. [Online]. Available: https://www.lytro.com/press/releases/lytro-brings-revolutionary-light-field-technology-to-film-and-tv-production-with-lytro-cinema.

[20] LightField/ Forum.com, *Adobe Light Field Camera Prototypes*. [Online]. Available: http://lightfield-forum.com/light-field-camera-prototypes/adobe-lightfield-camera-protypes/.

[21] T. G. Georgiev and C. Intwala, "Light field camera design for integral view photography",

[22] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview", *IEEE Journal of Selected Topics in Signal Processing*, 1–1, 2017, ISSN: 1941-0484. DOI: 10.1109/jstsp.2017.2747126. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2017.2747126.

[23] P. Blanche, "Toward the ultimate 3-D display", *Information Display*, vol. 28, no. 2-3, pp. 32–37, Feb. 2012, ISSN: 0362-0972.

[24] T. Balogh, P. T. Kovacs, Z. Dobrányi, A. Barsi, Z. Megyesi, Z. Gaál, and G. Balogh, "The holovizio system – new opportunity offered by 3D displays", in *Proceedings of the TMCE 2008*, TMCE, 2008.

[25] T. Iwane, "High-resolution 3D light-field display", *SPIE Newsroom*, 2017. DOI: 10.1117/2.1201611.006623. [Online]. Available: https://doi.org/10.1117/2.1201611.006623.

[26] ——, "Light field camera and integral 3D display: 3D image reconstruction based on light field data", in *2014 13th Workshop on Information Optics (WIO)*, IEEE, 2014. DOI: 10.1109/wio.2014.6933289. [Online]. Available: https://doi.org/10.1109/wio.2014.6933289.

[27] T. Balogh, P. T. Kovacs, and A. Barsi, "Holovizio 3D display system", in *2007 3DTV Conference*, IEEE, 2007. DOI: 10.1109/3dtv.2007.4379386. [Online]. Available: https://doi.org/10.1109/3dtv.2007.4379386.

[28] N. Patel, *Light-Field-Completition*. [Online]. Available: https://github.com/nrpatel/lfptools.

[29] D. G. Dansereau, *Light-Field Toolbox for Matlab v0.4*. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4.

[30] M. Řeřábek and T. Ebrahimi, "New light field image dataset", in *8th International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, 2016*, EPFL, 2016. [Online]. Available: http://mmspg.epfl.ch/EPFL-light-field-image-dataset.

[31] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras", in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '13, Washington, DC, USA: IEEE Computer Society, 2013, pp. 1027–1034, ISBN: 978-0-7695-4989-7. DOI: 10.1109/CVPR.2013.137. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.137.

[32] I. Viola, M. Rerabek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi, "Objective and subjective evaluation of light field image compression algorithms", in *Picture Coding Symp.*, 2016.

[33] A. S. Raj, M. Lowney, and R. Shah, "Light-field database creation and depth estimation",

[34] R. Ng, "Fourier slice photography", *ACM Trans. Graph.*, vol. 24, no. 3, pp. 735–744, 2005. DOI: 10.1145/1073204.1073256. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073256.

[35] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Linear volumetric focus for light field cameras", *ACM Trans. Graph.*, vol. 34, no. 2, 15:1–15:20, Mar. 2015, ISSN: 0730-0301. DOI: 10.1145/2665074. [Online]. Available: http://doi.acm.org/10.1145/2665074.

[36] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an efficient representation of visual reality", *IEEE Multimedia*, vol. 23, no. 4, pp. 14–20, 2016, ISSN: 1070-986X.

[37] Y. Hu, L. Zhang, J. Li, and S. Mehrotra, "ICME 2016 image recognition grand challenge", *2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2016. DOI: 10.1109/icmew.2016.7574663. [Online]. Available: http://dx.doi.org/10.1109/ICMEW.2016.7574663.

[38] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement", in *2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, IEEE, 2016. DOI: 10.1109/icmew.2016.7574671. [Online]. Available: https://doi.org/10.1109/icmew.2016.7574671.

[39] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression", *2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, vol. 00, pp. 1–4, 2016. DOI: doi.ieeecomputersociety.org/10.1109/ICMEW.2016.7574674.

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

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

[42] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction", in *2016 IEEE International Conference on Multimedia and Expo Workshop, ICMEW 2016 :*, 2016, ISBN: 978-1-5090-1552-8. DOI: 10.1109/ICMEW.2016.7574673.

[43] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting plenoptic images as multi-view sequences for improved compression", 2017. [Online]. Available: http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Amiun%3Adiva-31455.

[44] M. W.M. a. David S. Taubman, *JPEG2000 image compression fundamentals, standards and practice*, 1st ed., ser. The Springer International Series in Engineering and Computer Science 642. Springer US, 2002, ISBN: 978-1-4613-5245-7,978-1-4615-0799-4.

[45] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: An overview", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000. [Online]. Available: http://jj2000.epfl.ch/jj_publications/papers/006.pdf.

[46] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*. [Online]. Available: 729000/1e396cb728d3f4f5030dcc4274a23ea6.

[47] F. Dufaux, G. Sullivan, and T. Ebrahimi, "The JPEG XR Image Coding Standard", *IEEE Signal Processing Magazine*, vol. 26, no. 6, 2009. DOI: 10.1109/MSP.2009.934187. [Online]. Available: http://dx.doi.org/10.1109/MSP.2009.934187.

[48] T. Richter, "Visual quality improvement techniques of HDPhoto/JPEG-XR", *2008 15th IEEE International Conference on Image Processing*, pp. 2888–2891, 2008.

[49] F. Zhang and J. Wang, "Study of the image compression based on SPIHT algorithm", *Intelligent Computing and Cognitive Informatics*, vol. 00, pp. 130–133, 2010. DOI: doi.ieeecomputersociety.org/10.1109/ICICCI.2010.70.

[50] *X264 encoder settings documentation*, 2011. [Online]. Available: http://www.chaneru.com/Roku/HLS/X264_Settings.htm.

[51] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard", *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221191. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2012.2221191.

[52] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)", *IEEE Trans. Circuits Syst. Video Techn.*, vol. 22, pp. 1669–1684, 2012.

[53] MulticoreWare Inc., *X265 encoder settings documentation*, 2014. [Online]. Available: http://x265.readthedocs.io/en/default/cli.html.

[54] R. S. Higa, R. Fredy, L. Chavez, R. B. Leite, R. Arthur, and Y. Iano, "Plenoptic image compression comparison between JPEG, JPEG2000 and SPITH", 2013.

[55] C. Perra, "On the coding of plenoptic raw images", in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, IEEE, 2014. DOI: 10.1109/telfor.2014.7034539. [Online]. Available: https://doi.org/10.1109/telfor.2014.7034539.

[56] C. Perra and D. Giusto, "JPEG 2000 compression of unfocused light field images based on lenslet array slicing", in *2017 IEEE International Conference on Consumer Electronics (ICCE)*, 2017, pp. 27–28.

[57] A. Aggoun, "A 3D DCT Compression Algorithm For Omnidirectional Integral Images", *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, pp. II–II, 2006.

[58] X. Dong, D. Qionghan, and X. Wenli, "Data compression of light field using wavelet packet", *2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2004. DOI: 10.1109/ICME.2004.1394394.

[59] F. Dai, J. Zhang, Y. Ma, and Y. Zhang, "Lenselet image compression scheme based on subaperture images streaming", in *ICIP*, IEEE, 2015, pp. 4733–4737.

[60] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of Lytro-Illum light fields", in *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*, 2015, pp. 494–497. DOI: 10.1109/IPTA.2015.7367195.

[61] C. Perra, "Light field image compression based on preprocessing and high efficiency coding", in *2016 24th Telecommunications Forum (TELFOR)*, 2016.

[62] C. Perra and D. Giusto, "Raw light field image compression of sliced lenslet array", in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2017.

[63] A. S. Akbari, N. Canagarajah, D. Redmill, D. Bull, and D. Agrafiotis, "A novel H.264/AVC based multi-view video coding scheme", in *2007 3DTV Conference*, IEEE, 2007. DOI: 10.1109/3dtv.2007.4379433. [Online]. Available: https://doi.org/10.1109/3dtv.2007.4379433.

[64] U. Fecker and A. Kaup, "H.264/AVC-Compatible Coding of Dynamic Light Fields Using Transposed Picture Ordering", in *13th European Signal Processing Conference (EUSIPCO)*, (Antalya, Turkey), Sep. 4–8, 2005.

[65] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo sequence based 2-D hierarchical reference structure for light-field image compression", *CoRR*, vol. abs/1612.07309, 2016. arXiv: 1612.07309. [Online]. Available: http://arxiv.org/abs/1612.07309.

[66] H. P. Hariharan, T. Lange, and T. Herfet, "Low complexity light field compression based on pseudo-temporal circular sequencing", in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2017, Cagliari, Italy, June 7-9, 2017*, IEEE, 2017, pp. 1–5. DOI: 10.1109/BMSB.2017.7986144. [Online]. Available: https://doi.org/10.1109/BMSB.2017.7986144.

[67] C. Conti, P. T. Kovács, T. Balogh, P. Nunes, and L. D. Soares, "Light-Field Video Coding Using Geometry-Based Disparity Compensation", in *3DTV Conf. - 3DTV-CON*, 2014, pp. 1–4.

[68] C. Perra, "Lossless plenoptic image compression using adaptive block differential prediction", in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015. DOI: 10.1109/icassp.2015.7178166. [Online]. Available: https://doi.org/10.1109/icassp.2015.7178166.

[69] I. Viola, M. Rerabek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi, "Objective and subjective evaluation of light field image compression algorithms", in *32nd Picture Coding Symposium*, Nuremberg, Germany, 2016, ISBN: 978-1-5090-5966-9. DOI: 10.1109/PCS.2016.7906379.

[70] I. Viola, M. Rerabek, and T. Ebrahimi, "Comparison and evaluation of light field image coding approaches", *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2740167.

[71] V. K. Adhikarla, M. Vinkler, D. Sumin, R. K. Mantiuk, K. Myszkowski, H. Seidel, and P. Didyk, "Towards a quality metric for dense light fields", *CoRR*, vol. abs/1704.07576, 2017. arXiv: 1704.07576. [Online]. Available: http://arxiv.org/abs/1704.07576.

[72] I. Viola, M. Rerabek, and T. Ebrahimi, "A new approach to subjectively assess quality of plenoptic content", in *Applications of Digital Image Processing XXXIX*, ser. Proceedings of SPIE, vol. 9971, San Diego, California, USA: SPIE, 2016, pp. 99710X–1 –99710X–13, ISBN: 978-1-5106-0333-2; 978-1-5106-0334-9. DOI: 10.1117/12.2240279.

[73] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 4, pp. 600–612, 2004.

[74] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM", in *ICPR*, 2010.

[75] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment", in *In Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, 2003, pp. 1398–1402.

[76] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index", *IEEE Trans. Image Processing*, vol. 23, no. 2, pp. 684–695, 2014. DOI: 10.1109/TIP.2013.2293423. [Online]. Available: https://doi.org/10.1109/TIP.2013.2293423.

[77] "FFmpeg", [Online]. Available: https://ffmpeg.org/.

[78] M. Rerabek and T. Ebrahimi, "Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments", *Applications Of Digital Image Processing XXXVII*, vol. 9217, EPFL-CONF-200925 2014.

[79] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, *Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders*, Dec. 2013.

[80] J. Bienik, M. Uhrina, M. Kuba, and M. Vaculik, "Performance of H.264, H.265, VP8 and VP9 compression standards for high resolutions", pp. 246–252, Sep. 2016.

[81] G. Bjøntegaard, *Calculation of average PSNR differences between RD-curves*, 2001.