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

Plenoptic Imaging and Image Data Conversion
Plenoptické zobrazování a konverze obrazových dat

Master's thesis

Bc. Adam Zizien

Master programme: Electronics and Communications
Branch of study: Media and Signal Processing
Supervisor: Ing. Karel Fliegel, Ph.D.

Prague, May 2019

Thesis Supervisor:

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

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zizien** Jméno: **Adam** Osobní číslo: **434855**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

Studijní obor: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Plenoptické zobrazování a konverze obrazových dat

Název diplomové práce anglicky:

Plenoptic Imaging and Image Data Conversion

Pokyny pro vypracování:

Podejte přehled technik pro efektivní reprezentaci dat a konverzi mezi modalitami plenoptického zobrazování s ohledem na současné výzkumné aktivity, zejména pak JPEG Pleno. Zpracujte přehled volně dostupných vhodných obrazových databází a výpočetních nástrojů. Na vybraných příkladech vyhodnoťte účinnost a analyzujte omezení studovaných metod.

Seznam doporučené literatury:

[1] Ebrahimi, T., Foessel, S., Pereira, F., Schelkens, P.: JPEG Pleno: Toward an Efficient Representation of Visual Reality, IEEE MultiMedia, 23 (4), 2016.
[2] Viola, I., Rerabek, M., Ebrahimi, T.: Comparison and Evaluation of Light Field Image Coding Approaches, IEEE Journal on Selected Topics in Signal Processing, 11 (7), 2017.
[3] Blinder, D., Ahar, A., Bettens, S., Birnbaum, T., Symeonidou, A., Ottevaere, H., Schretter, C., Schelkens, P.: Signal processing challenges for digital holographic video display systems, Signal Processing: Image Communication, 70, 2019.
[4] Ng, R.: Digital Light Field Photography, PhD dissertation, Stanford, 2006.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Fliegel, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **08.02.2019** Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

Ing. Karel Fliegel, Ph.D. podpis vedoucí(ho) práce	prof. Mgr. Petr Páta, Ph.D. podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání Podpis studenta

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, May 2019

....
Bc. Adam Zizien

Abstract

More and more data is captured in today's world, with sophisticated devices. Having multiple cameras in a new smartphone is almost a given. Different techniques with different intentions in mind were developed to acquire as much information about a scene as possible and further leverage that information in many ways. In this work, three imaging modalities are presented, that being Light field, Point-cloud and Holography.

General principles for each modality and the connections they have to the plenoptic function are mentioned. The current techniques for conversion between the modalities are explored. So are compression approaches.

The effects of light field image compression with state-of-the-art encoders on disparity estimation are evaluated and discussed. Video encoders showcase better performance than still image encoders, thanks to temporal coding. However, all tested encoders exhibit scene dependent behaviour.

There is no existing standard for compression of plenoptic images nor for conversion between the different representation as JPEG Pleno is still in development. Existing algorithms are showing promise, both in terms of compression and conversion. However, further research still needs to be done.

Keywords: light field, digital holography, point cloud, compression, conversion, image processing.

Abstrakt

Dnešní sofistikované přístroje zaznamenávají stále více obrazových dat. Prakticky každý nový mobilní telefon je vybaven více než jedním fotoaparátem. Pro zachycení a zpracování dat, a to pro různé účely, bylo vyvinuto mnoho technik. V této práci jsou prezentovány tři zobrazovací metody – light field, point cloud a holografie.

Jsou zmíněny obecné zásady každé modality společně s vazbami na plenoptickou funkci. Dále jsou zkoumány současné postupy konverze a komprese.

V práci je diskutován vliv komprese light field obrazových dat na odhad disparity. Zkoumané video kodeky vykazují lepší výsledky než metody určené čistě pro kompresi statických snímků, protože lépe využijí podobnost jednotlivých snímků. Avšak všechny kodeky vykazují závislost na testovaných datech.

V současnosti neexistuje žádný standard pro kompresi či konverzi plenoptických dat, jelikož JPEG Pleno je stále ve vývoji. Existující algoritmy vykazují slibné výsledky v oblastech komprese i konverze. Další výzkum v této oblasti je však nezbytný.

Klíčová slova: light field, digital holography, point cloud, komprese, konverze, obrazové zpracování.

Acknowledgements

This thesis would not be as it is without the help of some incredible people who gave me advice, supported me and encouraged me. I would like to thank Professor Tinne Tuytellars and my supervisor Ing. Karel Fliegel, Ph.D. for finding time for me in their busy schedules. For being available for discussion, while offering helpful suggestions. I would also like to thank my family and friends, for being there for me, even though I was not able to always be there for them. A special thanks goes to my parents, Marta and Adama Zizien, for always standing behind me and pushing me forward in whatever I decided to do and my sister Anna, for always making me smile and brightening my days. Merci beaucoup.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

Humans are taking and sharing more pictures than ever. To be able to capture the world around us becomes more important each year. Gone are the days of bulky cameras that can capture images in just black and white. The industry understands the value of information which could be captured at any moment in a scene. Thus the jump to colour photography was made followed by a shift from analog to digital and subsequently computational photography. It is therefore not only about the captured data but also about its manipulation. So came the ability to stitch images together to make a panorama or to combine them in such a way, that a High Dynamic Range (HDR) image is created.

The next innovation was to capture more than a 2D representation of a scene. The scene can be captured from two different positions to simulate the Human Visual System (HVS). This 3D representation incorporates even more data about the scene, which can be further manipulated to achieve specific goals. The next logical step to get even more information about the scene is to acquire even more views.

Different techniques with different intentions in mind were developed to acquire as much information about a scene as possible. Three main modalities will be introduced and discussed in this work, that being Light field, Point-cloud and Holography. Each of which has some advantages and disadvantages, but the information captured is in a lot of ways similar or even the same. The conversion between the modalities should thus be possible (depicted in Figure 1.1). One could, therefore, be able to leverage the best aspect of each.

To understand the principles of each modality, the introduction of a plenoptic function is first required (Chapter 2). The basics of the modalities are then described in Chapters 3, 4 and 5. Current methods of conversion between the different representations are discussed (Chapters 6, 7, 8). So are compression approaches (Chapter 9). Available

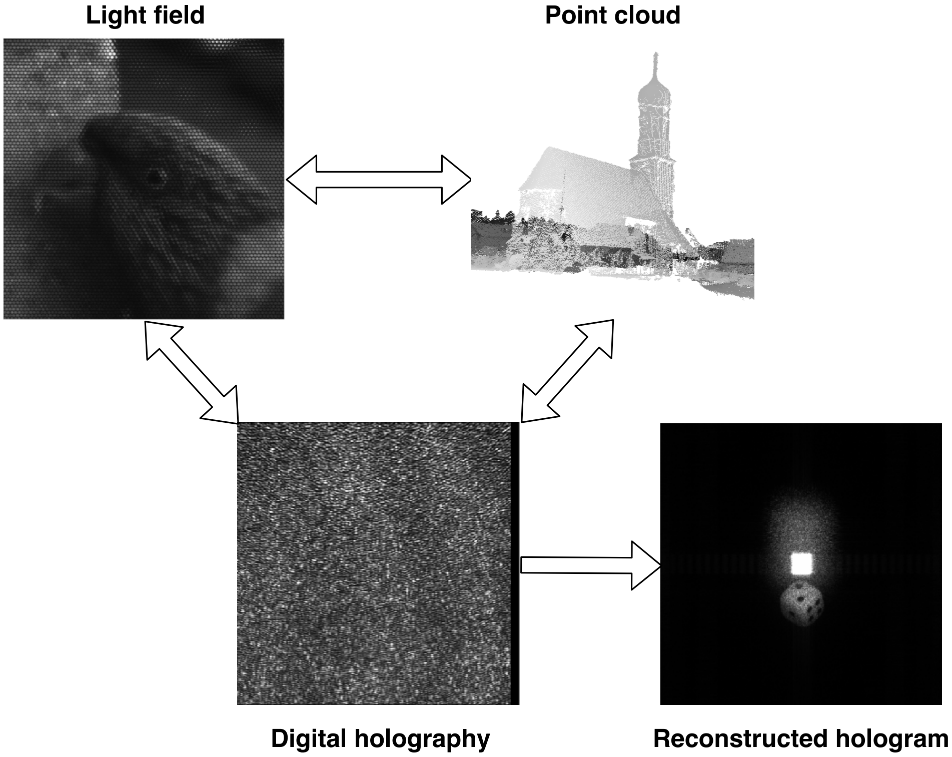


Figure 1.1: Conversion between the different imaging modalities with examples[2]. (holography images taken from [1]).

implementations of the techniques are mentioned together with datasets and software used to test them in Chapter 10. In Chapter 11 the effects compression of light field images has on disparity estimation is examined. It can be expected that some form of compression will have to be applied to light field images before they can be stored. The chosen compression algorithm will inevitably alter the image in a way that some amount of distortion will appear. In Chapter 11 the effects compression of light field images has on disparity estimation is explored. Various state-of-the-art image and video compression algorithms are tested on several estimation methods.

[2] Light field image and point cloud provided as part of: `http://dgd.vision/Tools/LFToolbox/` and `http://vision.middlebury.edu/stereo/code/` (used 20.5.2019)

Chapter 2

Plenoptic function

A region of space filled with visible light, a narrow part of the whole electromagnetic spectrum which can be viewed by the HVS, contains visual information for every point in that space. A dense array of light rays of various intensities passes through each point in the space. This set of rays is mathematically termed a pencil P [2]–[4].

If we position a pinhole camera at a point (x, y, z) in space, a specific pencil of rays $P(x, y, z)$ is captured. To describe the intensity distribution of the light, spherical coordinates (θ,ϕ) can be used [2], [4], [5], extending the pencil definition to $P(x, y, z, \theta, \phi)$. Adding color into the mix, thus considering how the intensity varies with wavelength λ, the pencil definition can be further extended to $P(x, y, z, \theta, \phi, \lambda)$ [2]. Lastly a temporal dimension is added if the scene to be captured is dynamic, yielding a function $P(x, y, z, \theta, \phi, \lambda, t)$, which is commonly called the plenoptic function, seen in Figure 2.1 [3].

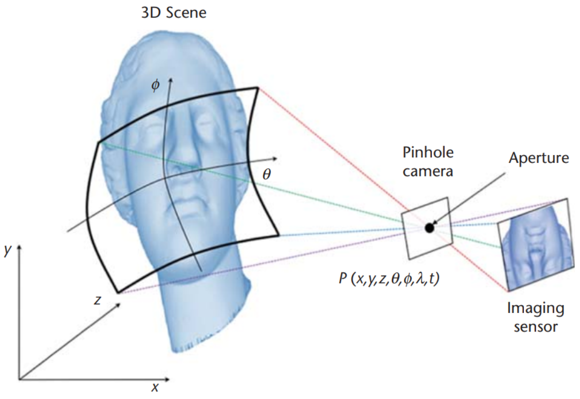


Figure 2.1: Visualization of the plenoptic function (taken from [4]).

The seven dimensions of the plenoptic function imply an enormous amount of data to be captured and represented. It is therefore essential to reduce the dimensionality.

Only a subset of the information is usually addressed in current representation models. This subset is further sampled and quantized. For example, the number of pinhole camera positions is defined, the full wavelength information is reduced to represent R, G, B colour channels or the number of frames per second captured is limited, usually to the point that only static images are acquired [3], [4].

Different sensors and camera systems are used to acquire some version of the plenoptic function. Metadata may also be acquired or inserted. The information captured may then be converted to a different representation format, which can include [3]:

- omnidirectional content often captured by a single camera or multiple cameras as a 360° panorama or a complete sphere mapped to a 2D image,
- depth-enhanced content extracted from multiple images captured with a digital camera or acquired with depth-sensing cameras or structured light systems,
- point cloud content commonly generated by a 3D scanner or a Lidar (Light Detection and Ranging) scanner. The information, in this case, must often be combined with additional modalities, so that a more complete plenoptic representation of the real world is characterized,
- light field content captured with an array of cameras or a single camera equipped with a microlens structure,
- holographic content that is computer-generated, and thus called Computer Generated Hologram (CGH), or acquired using interferometry. If the time dimension is omitted, the full plenoptic function can then be portrayed to a 2D map [4].

Chapter 3

Light field

Having one digital camera lets us capture one view of a specific scene. Adding a second camera with a slight offset from the first one, thus simulating HVS, gives us a stereoscopic view of the scene, therefore providing additional information about the scene. This configuration of cameras meets the common definition of a 4D light field, derived from the plenoptic function, as $P(x, y, \theta, \phi)$.

3.1 Light field acquisition

The goal of light field photography in comparison to traditional photography is to capture additional spatial information about the scene. There are many ways of achieving this goal. This section gives a brief overview of the most common techniques of acquisition.

3.1.1 Single Lens Stereo

The first method is single lens stereo. The most straightforward way to capture additional angular information about the scene is to capture the scene from two nearby viewpoints, thus simulating HVS [5]. This system, sometimes referred to as a binocular stereo system, can be highly effective in some situation, but like any other system, it also suffers from difficulties, mainly with camera calibration [6].

3.1.2 Camera array

An improvement on single lens stereo is a camera array. Instead of using just two cameras to capture the scene, the camera array uses multiple cameras usually positioned in a grid. A variety of viewpoints can be captured this way, thus acquiring additional angular information about the scene. Synchronization and calibration of the array are crucial in this instance. The output of the system is an array of images, which are shifted slightly horizontally, vertically or both from each other, creating a parallax. Data processing is then needed to visualize the scene [5].

3.1.3 Plenoptic camera

A plenoptic camera differs from the camera array in that it can capture all of the images in a single snapshot on one photodetector element (for example a charge coupled device, also called a CCD) [6]. There are two main ways of doing so, both derived from a proposal made in 1908 by Gabriel Lippmann to use microlenses in front of the image plane [7]. The two approaches are called Plenoptic 1.0 and Plenoptic 2.0.

In the Plenoptic 1.0 approach, first implemented by Ren Ng in 2006 [8], a microlens array is added to the usual camera design of a digital sensor, main optics and an aperture. The array is placed precisely at the focal length from the sensor, in the focal plane of the primary lens. The incoming light is then split by the microlens array according to the direction it is coming from [7]. One point viewed from different angles will appear under each microlens. The image under each microlens will thus appear blurry, which can be seen in Figure 3.2 [5]. The size of the microlenses is determined by the aperture of the main optics and should be neither too big, nor too small so that the sensor area is not wasted or the adjacent images under the microlenses do not overlap. The number of microlenses further determines the spatial resolution of the rendered view.

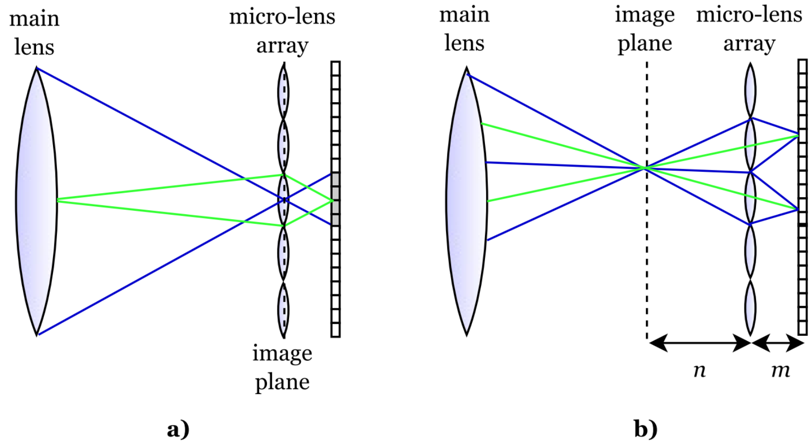


Figure 3.1: The different position of microlenses in a) Plenoptic 1.0 and b) Plenoptic 2.0 (taken from [5]).

The difference between the Plenoptic 1.0 design and the Plenoptic 2.0 design is the position of the microlenses, as seen in Figure 3.1. The latter also called the Focused Plenoptic design, places the microlens array at a specific distance m, so that the microlenses are focused on the image plane of the primary lens. One can then visualize each microlens as a pinhole camera, which observes a fraction of the virtual image inside the camera from a particular position. Under each microlens is, therefore, a sharp inverted low-resolution image, as seen in Figure 3.2. This approach achieves high spatial resolution [5], [7].

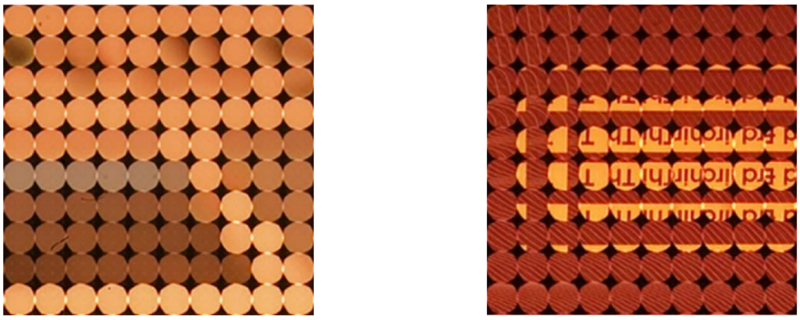


Figure 3.2: The different microimages in Plenoptic 1.0 (left) and Plenoptic 2.0 (right) [1].

3.2 Data representation of light fields

There are various ways of representing light field data. The first and most straightforward way of representing light field data is a lenslet structure. In this method, the images under each microlens are shown and a lenslet image is made (Figure 3.3). In the case of a Lytro camera[2], the images under each microlens have a resolution of 15×15 pixels. These images are then arranged into a 625×434 lenslet array. The array has the same shape as the microlenses and thus is circular. What is also noticeable is that some information is missing under the microlenses. This is due to the circular shape of the microlenses, thus the construction of the camera. There are some methods of filling in the missing information [5].

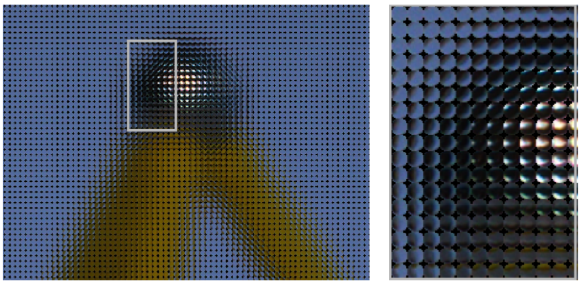


Figure 3.3: A lenslet structure (taken from [5]).

[1] Source: `http://www.tgeorgiev.net/EG10/Focused.pdf` (used 23.10.2018)

[2] Lytro Illum to be exact. More info about the camera can be found for example in `https://doi.org/10.1007/s00138-019-01013-z`.

Another way of representing light field data is with sub-aperture images, shown in Figure 3.4. These images are constructed by extracting the same pixel in terms of the position under each microlens, thus reconstructing one specific view of a scene. Considering all the views, an array of sub-aperture images can be constructed. For the above mentioned Lytro camera, the array will have a size of 15×15 individual sub-aperture images [5].

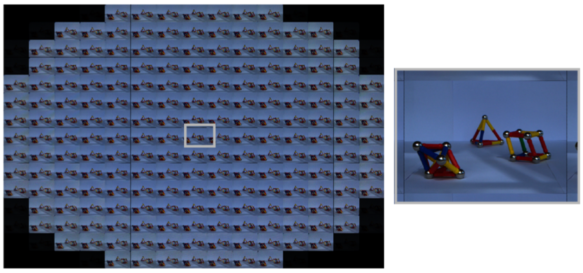


Figure 3.4: An array of sub-aperture images (taken from [5]).

EPIs are a third way of representing light field data. To understand this method, we must first introduce another way the plenoptic function can be described. If we consider a light ray passing through two planes, which are positioned some specified distance apart, then because the light ray remains constant through free space, we acquire additional information about the angle in which the light ray intersects both planes (Figure 3.5).

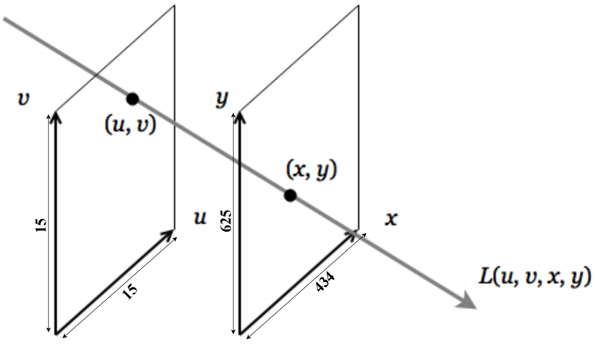


Figure 3.5: Capturing angular information about a light ray with the use of 2D planes. Dimensions represent the spatial and directional resolutions for the Lytro Illum camera (edited image from [5]).

In the EPIs, v and y stay fixed, while u varies horizontally and x vertically. This gives each EPI a spatial resolution of 434 pixels and a directional resolution of 15 pixels, for the Lytro camera in question. The depth of the scene can then be estimated based on the slopes of each EPI, see Figure 3.6 [5].

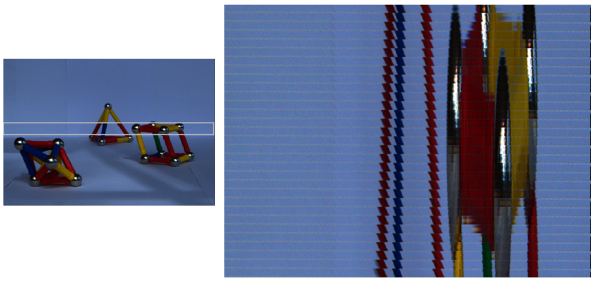


Figure 3.6: Epipolar images (taken from [5]).

3.3 Displaying light fields

To display light field content one can use a variety of different viewing devices, having drastically different experiences with each. Using a 2D system is not ideal in the sense that only a subset of the content can be displayed. A conventional 3D systems can be bothersome, for most the user needs 3D glasses, wearing which may lead to visual fatigue. Autostereoscopic displays, which do not require those glasses, can be preferable for the user. Two types of displays satisfy all the human vision depth cues, a light field 3D display (LFD) and a holographic display. Both can reproduce horizontal and vertical parallaxes [9].

An LFD does, ideally, the perfect inversion of the light field acquisition. Therefore the light rays captured in the scene are reproduced with the correct intensity, colour and direction. The original discovery, known as integral photography, was made by Lippmann in 1908 [10]. The digital realization of this technology is usually referred to as integral imaging, with the display being called an integral display. Another way of reproducing light fields is with so-called multiview 3D displays. They can be considered as LFD if the number of views one can observe is sufficiently large. Lastly, holographic stereograms, which are based on wavefront reconstruction rather than light rays, can also be used [9].

Chapter 4

Point cloud

A point in geometry specifies a unique location in space, commonly characterized by Cartesian coordinates (x, y, z). Spherical coordinates may also be used, in which case, each point is represented by the horizontal and vertical angles, θ and ϕ, together with a measured distance r [11]. In both cases, a parallel to the plenoptic function can be drawn. A point also has no length, area or volume. A point cloud represents a set of these multi-dimensional points (see Figure 4.1). Additional information about each point such as colour information, transparency, time of acquisition and reflectance can be captured as well [12], [13].

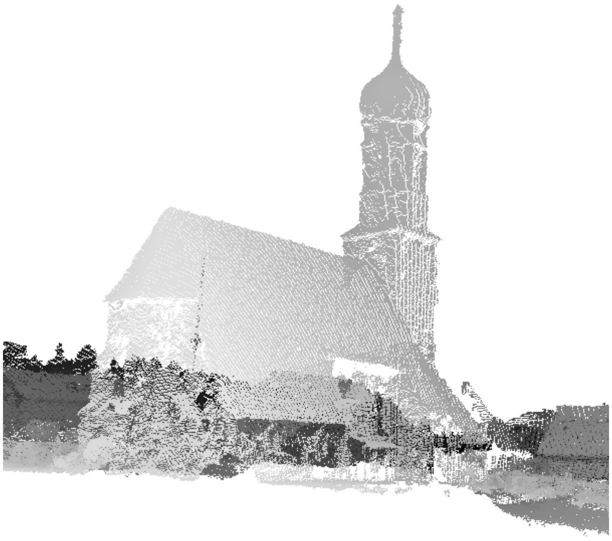


Figure 4.1: Example of a point cloud. A set of points in space can be seen, with noticeable gaps between them.

4.1 Point cloud acquisition

The acquisition of point cloud was extensively discussed and researched over the past decades. One can categorize the different techniques based on the measurement principle. Firstly, we can look at the distinction between active and passive techniques. The latter focuses on collecting information from scenes with reasonable ambient lighting, while the former actively manipulates the scene itself. For passive techniques, a further distinction of single view or multiple views of a scene can be drawn. Active techniques may be further divided into monostatic and multistatic sensor configurations, based on the position of the emitter and receiver component [13]. In practice, point cloud content is typically captured with either a 3D scanner or a Light Detection and Ranging (Lidar) scanner or computed from an array of images, each depicting the scene from different viewpoint [4].

4.1.1 Passive techniques

Point cloud generation based on passive techniques relies only on radiometric information, which is represented by 2D imagery, in the form of simple intensity measurements per pixel. There are two main strategies for the reconstruction of 3D structures from the captured intensity images. In the first strategy, the acquisition of multiple images of a scene is exploited. For this purpose, either multiple cameras or a single moving camera can be used. Stereo matching techniques are then applied to estimate the respective 3D structure of the scene, by finding points of corresponding intensity in the images and converting their 2D locations into 3D depth values. The goal of the second strategy is the simultaneous recovery of both the 3D structure of the scene and the location of the image, which correspond to the camera pose. Structure-from-Motion techniques are used for this [13].

4.1.2 Active techniques

Instead of relying on simple intensity measurement per pixel for each scene, active techniques actively manipulate it. This involves emitting signals and recording the observations in the scene. The emitted signal may be a coded structured light pattern in either the visible or infrared spectrum, or electromagnetic radiation in the form of laser light. In the former case projected light patterns manipulate the illumination of the scene, so that particular projected labels can be easily decoded in the captured images. Triangulation is then used to recover the 3D coordinates. A further classification based on the used pattern codification strategy can be made, as is mentioned in [13], namely: Direct codification, Time-multiplexing and Spatial neighbourhood codification strategies.

4.2 Point cloud representations

Whereas laser scanners record the geometric and radiometric information from the scene by successively considering points on a discrete, regular raster, range cameras use an internal sensor array for simultaneous consideration of all points on the same discrete, regular raster. Thus, acquisition of a dynamic scene can be performed with a sufficiently fast range camera [13].

A specific scene can have a high number of points, posing significant memory demands. To store and represent this data, special data structures are used, including tree structures, such as the k-d tree and octree, or methods utilizing the grid map method. In the latter case, the captured data are represented with depth images generated from point clouds, mapping 3D coordinates onto 2D image coordinates [11].

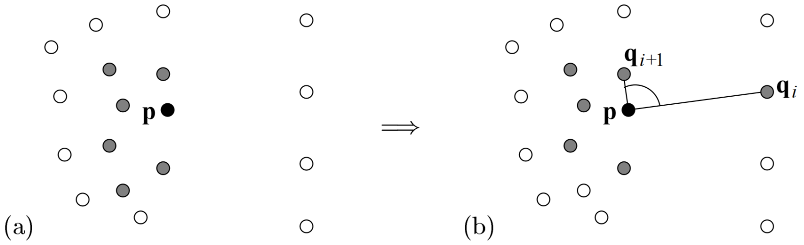


Figure 4.2: Definition of the environment: a) k-nearest neighbours and b) angle criterion (taken from [14]).

The connectivity of triangular meshes, which contain faces or surfaces, is replaced by the environment of each point in the point cloud representation. We assume that the nearest neighbours of a point contribute to the environment. We can then define the environment of a point by its k-nearest neighbours. This is not an optimal choice, because the k-nearest neighbours do not have to cover the whole environment, as can be seen in Figure 4.2. It is better to select k-neighbours which are distributed all around the point by introducing an angle criterion. The environment can then be visualized by a triangle fan and the whole point cloud by the rendering of all the triangle fans [14].

Two major problems have to be solved before applying a specific visualization method, that being the handling of a large amount of data and distortion. In the case of oversampling, thus having a high sample rate, the point cloud contains redundancy. We can then consider each point to have a certain amount of information content, which gives information about the captured scene. The deletion of points of low information content, thus low entropy, leads to the reduction of data. What is also useful, for example in networking, is to store separately general low-resolution information about the scene and detail information, in the form of vectors. When displaying a complex object, the low-resolution part is first shown and then progressively improved by the detail information. Storing the detail information in vectors is also beneficial in terms of interactive editing of the scene. The user can edit the global shape or structure of the scene while leaving the small detail unchanged in a process called multiresolution editing, as can be seen in Figure 4.3.

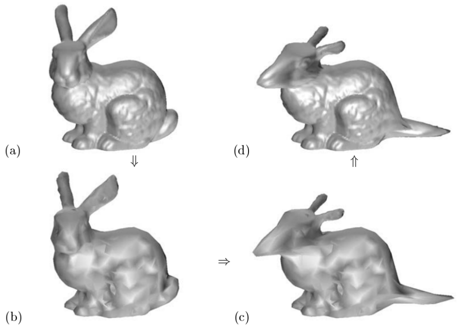


Figure 4.3: Multiresolution editing (taken from [14]).

Smoothing is used to eliminate distortion. Several smoothing methods exist, including various smoothing operators and filters. Some operators try to make edge lengths as uniform as possible, which can be problematic. For example in the case of modelling purposes, it is important to use such smoothing operators which do not affect the geometrical aspects of the scene in question, as is shown in Figure 4.4.

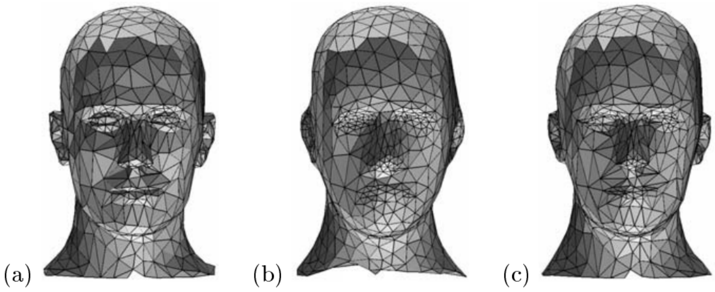


Figure 4.4: Application of different smoothing operators (taken from [14]).

Chapter 5

Digital Holography

The word holography is a combination of Greek words ”holos” meaning complete or entire and ”graphein” meaning to write. The term was created by Dennis Gabor in 1948 together with a method of recording and reconstructing amplitude and phase of a wave field. A hologram is a recording of an interference pattern between a wave field scattered from an object and a reference wave. The image contains information about the entire three-dimensional wave field [1].

5.1 Fundamental principles

Two terms form the basis of holography, that being diffraction and interference. Diffraction is a phenomenon which occurs when light encounters an obstacle. The Huygens principle (Figure 5.1) explains diffraction as follows:

> “Every point of a wavefront can be considered as a source point for secondary spherical waves. The wave field at any other place is the coherent superposition of these secondary waves” [1].

Interference refers to the superposition of two or more waves in space. A term that is closely tied to interference is coherence, which is the ability of light to interfere. There are two aspects of coherence, spatial and temporal. The former describes the correlation of different parts of the same wavefront, while the latter describes the correlation of a wave with itself at different instants.

An optical set-up is usually used to record a hologram. It consists of a light source (further details in Chapter 7.1), mirrors, and lenses. A beam splitter is used to split the light into two partial waves. One, called the object wave, illuminates the object, scatters, is reflected and recorded. The other also called the reference wave, illuminates

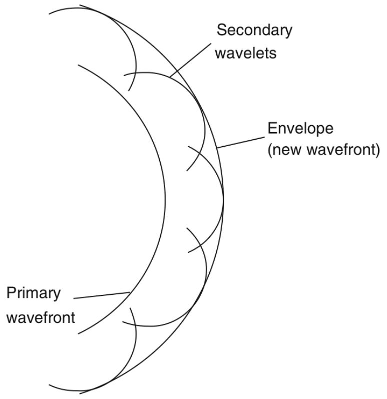


Figure 5.1: The Huygens principle (taken from [1]).

the recording medium directly. The interference pattern of both waves is recorded. This pattern is traditionally called a hologram. Reconstruction of the original object wave is done by illuminating the recorded hologram with the reference wave. If the reference wave has the same parameters as the one used for the recording, the reconstructed virtual image should appear in the same position as the original object. Changing the wavelength, for example, results in a coordinate shift and thus distortion. Both the recording and the reconstruction can be seen in Figure 5.2.

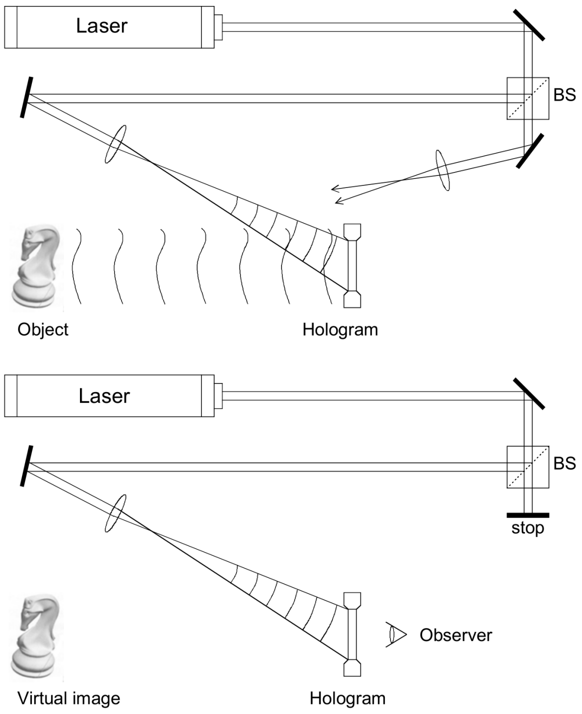


Figure 5.2: Hologram recording (top) and reconstruction (bottom) (taken from [1]).

5.2 Hologram acquisition

To map the displacements of rough surfaces, a method named Holographic Interferometry is used. This method enables the comparison of stored wavefronts existing at different times, has high sensitivity, thus high accuracy, and is non-contact and non-destructive. The difference between this method and general holography is in the recording phase. Two wave fields are recorded on a single photographic plate instead of one, each for a different state of the recorded object. The object is then represented in two states, its reference (undeformed) state and its loaded (deformed) state. When the reference wave is then used to reconstruct the image, only one of the superimposed images is visible, see Figure 5.3.

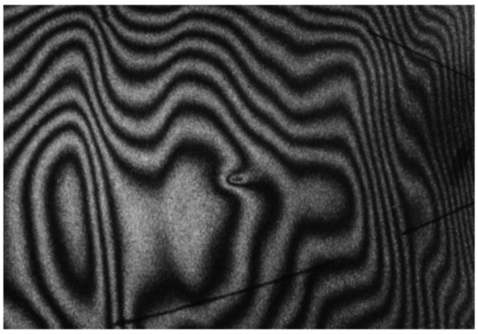


Figure 5.3: Holographic interferogram (taken from [1]).

In Digital Holography, the interference of the waves happens at the surface of an electronic array sensor. Therefore the hologram is recorded electronically. Purely digital methods can then be used for the simulation and reconstruction. The recording of digital holograms has several advantages over the recording on a holographic film. Vibrations do not affect the recording as much, because the sensitivity of the digital imager is higher. The imager is also linear, which is not the case for holographic film. The main issue in digital holography is with Shannon's criterion. Specifically, the maximum fringe frequency, which is given by the minimum inter-fringe distance on the recorded fringe pattern, must be lower than the Nyquist frequency of the imager so that aliasing does not occur [15], [16].

A hologram can also be produced without any optical interference. In this case, the term Computer Generated Hologram (CGH) is used. There are three categories of CGH algorithms, based on their basic processing unit. A point-based CGH approach assumes that the object in question consists of multiple points. Elementary holograms for each point can, therefore, be calculated and superimposed to acquire the final one. The plane-based approach uses the Fourier transform to represent the wavefront propagation on object planes onto the hologram plane. The hologram is thus a superposition of Fourier transforms of object planes in depth. Lastly, a mesh-based CGH is used to represent the 3D object in a more realistic way. It finds corresponding spatial frequencies between given and reference triangles and calculates the angular spectrum of each mesh based on it [17].

5.3 Holographic data representation

There are various ways of representing holographic data. The hologram data can be represented by the following formats:

- intensity information or interferograms, for which the intensity of the complex wavefield is represented by three different phase shifts of the reference wave, being I1, I2 and I3 with the shifts of 0, pi/2 and pi,
- phase shifted distances, which need only two components (D1 and D2) to represent the complex wave field. This is due to the fact that the complex wavefield has only two components, amplitude and phase,
- real-imaginary represents the interference wavefield using Cartesian coordinates, therefore two components of a complex number,
- amplitude-phase is similar to the above-mentioned format, with the difference being that in this case polar coordinates are used instead of Cartesian coordinates. This means that the two components are now amplitude and phase [18]. The difference between both formats can be seen in Figure 5.4.

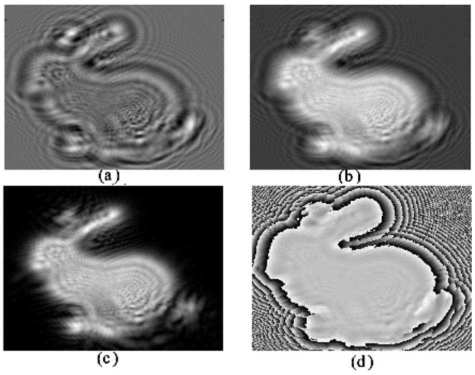


Figure 5.4: The difference between real (a) - imaginary (b); and amplitude (c) - phase (d) (taken from [18]).

Chapter 6

Conversion between Point cloud and Light field

The above-mentioned modalities represent the acquired information in different ways and are thus used in different circumstances. Each has various benefits and drawbacks. The conversion between the different representations should be possible.

Some obvious difficulties need to be resolved, for example, what should be done about missing information. For this reason, almost all the works are focusing on the conversion from Light field, which as a modality stores, in general, more information, and not from Point cloud. However, there already exist very detailed coloured point clouds, which should enable the extraction of light field images. Furthermore, to create a dense 360° point cloud, a high number of viewpoints is necessary. This could lead to a troublesome acquisition process and to an increase in the amount of data that needs to be stored. Point clouds could be a better option in those cases than light fields. This chapter gives an overview of the different methods of conversion between Point cloud and Light field, as seen in Figure 1.1.

As [19] mentions, we can divide the methods for converting from Light field to Point cloud into two distinct categories, that being optimization-based and learning-based methods. Each of which can be further divided. A depth or a disparity map is largely the output of the above-mentioned techniques, as it can be used as an intermediate step in the conversion. The whole procedure is, in that case, split into several related steps.

6.1 Optimization-based methods

This category encapsulates a host of distinct approaches with a common description. It can be said that the works in this category are based first and foremost on theories (mathematical, physical, etc.), thus being more general and more comfortable to build upon or recreate. The same cannot be said about the second group, in which machine learning techniques are applied. What can also be seen is a mostly unavoidable trade-off between the computational time and the performance for these conventional techniques [19].

6.1.1 EPI-based

The first subcategory of the optimization methods includes those based on epipolar plane images (EPIs). The benefit of using EPIs is in the reduction of dimensionality compared to the full light field image. Two main ways are used for the calculation of lines or the orientation of the EPIs, a structure tensor or a spinning parallelogram.

The structure tensor is used to compute horizontal and vertical slopes of the EPIs. A relation between the slope level and the depth can then be exploited. [20], for example, formulate this problem as a variational labelling one. For the depth map estimation, they use a global optimization framework.

A SPO can also be used to locate and calculate the lines and orientations of the EPIs, as is shown in [21]. Their method extracts the depth information by maximizing the distribution distances between two parts of a parallelogram window. This technique is further extended in [22], with the use of multi-orientational EPIs (Figure 6.1).

6.1.2 Line-fitting based

The line-fitting based methods could very well be included in the EPI-based section. The line-fitting term encapsulates techniques that use EPIs as well, but in different ways than was mentioned above.

In [23] a fine-to-coarse strategy with further propagation is used. It uses EPIs together with a line fitting function algorithm. A further extension of this method is in [24], which uses the same line fitting concept with additional Semi-Global Matching (SGM) based initialization. The focus is in this case on speed and computational complexity.

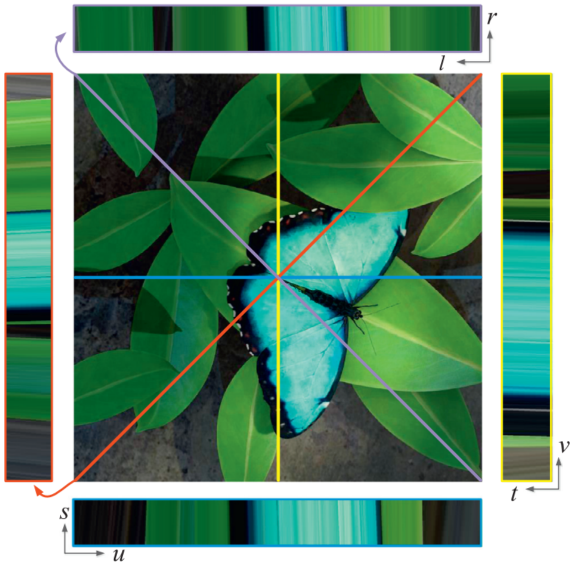


Figure 6.1: Multi-oriентional EPIs. The slopes of the lines are related to disparity (taken from [22]).

6.1.3 Defocus and correspondence based

In this section, methods, which use defocus and correspondence as their basis for estimating depth from light field images, are described. The defocus and correspondence is an umbrella term used to label another attribute of light field images, which is being able to focus at different depths of a scene. EPIs are mainly used for this as well. In traditional photography, multiple image exposures or complicated apparatuses have to be used. There are various ways of exploiting this property.

Patch-based variance measurement is used in [25] to measure the defocus. It is shown that the combination of both cues leads to faster computations and more accurate results than using just one of them. Their whole framework can be seen in Figure 6.2.

Another approach is to use focal stacks, sets of images focused at different distances, to build a cost volume. In [26] four separate stacks are used. These partial focal stacks are designed to help in the presence of occlusion. Different data costs are proposed in [27] with the aim being better occlusion and noise handling, comparing to conventional correspondence data costs.

6.1.4 Multi-view stereo-based

Multi-view stereo-based methods mostly leverage the fact that a light field image is essentially constructed from sub-aperture images. Conventional stereo-based algorithms can then be used to estimate the depth/disparity map.

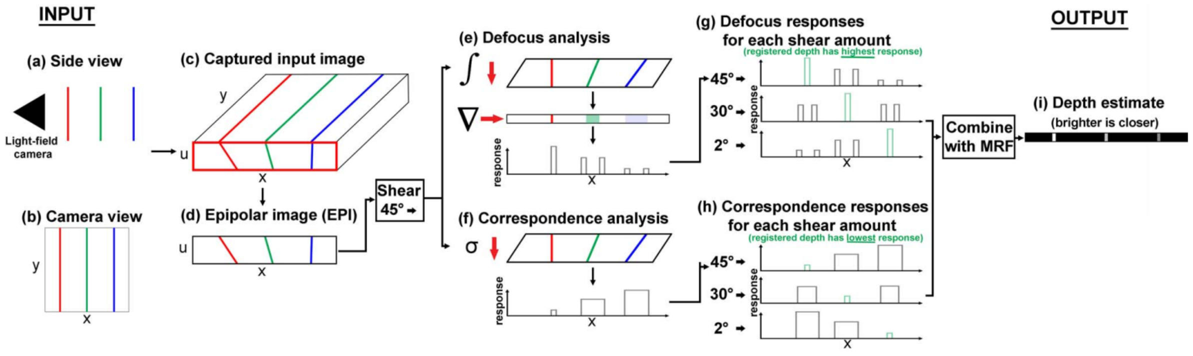


Figure 6.2: Framework taken from [25]. The input image is converted to a set of EPIs, to which a defocus and a correspondence analysis is performed. The output of both cues are then combined using Markov random fields.

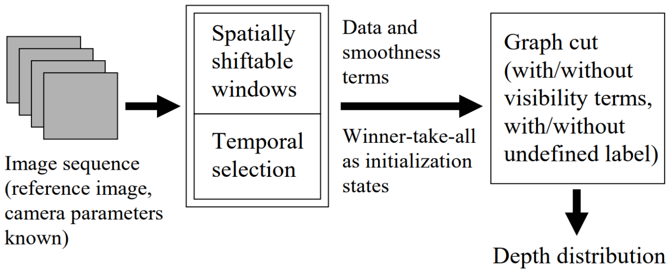

Spatially shiftable windows
Temporal selection
Data and smoothness terms
Winner-take-all as initialization states
Graph cut (with/without visibility terms, with/without undefined label)
Image sequence (reference image, camera parameters known)
Depth distribution

Figure 6.3: The stereo system used in [28] is composed of two approaches. In the first a selection of temporal subsets by the use of spatially adaptive windows. The second uses graph cuts as the basis for a global minimization approach to handle occlusions. (Figure taken directly from source.)

The stereo matching problem is formulated as a Markov network consisting of Markov random fields in [29]. In [28] a combination of shiftable windows and a dynamically selected subset of neighbouring images is used to do the matching. Additionally, a global energy minimization framework is used to label occluded pixels, thus improving the robustness of the algorithm, as can be seen in Figure 6.3. Disparity maps from specific pairs of images are computed in [30], providing a non-dense estimation. An interpolation method is then used to fuse them.

6.2 Learning-based methods

A different approach from the optimization methods is the learning-based approach. In this case, machine learning, neural networks to be more specific, are mostly used to estimate the depth maps. In comparison to other techniques, they yield better results, while also being computationally less expensive. Excluding the time it takes to train them.

The method proposed in [19] can be taken as a basis of what convolutional neural networks can do. As of writing this thesis, they score either the highest or near the top in most tests of the 4D Light Field Benchmark[1]. The design of the network consists of four separate processing streams for four angular directions of the sub-aperture images. The network, therefore, produces meaningful representations in each direction, which are later combined into the final product. This method is highly dependent on the training data. To deal with the problem of insufficient data to train the network on a data augmentation method is also proposed. The whole architecture can be seen in Figure 6.4.

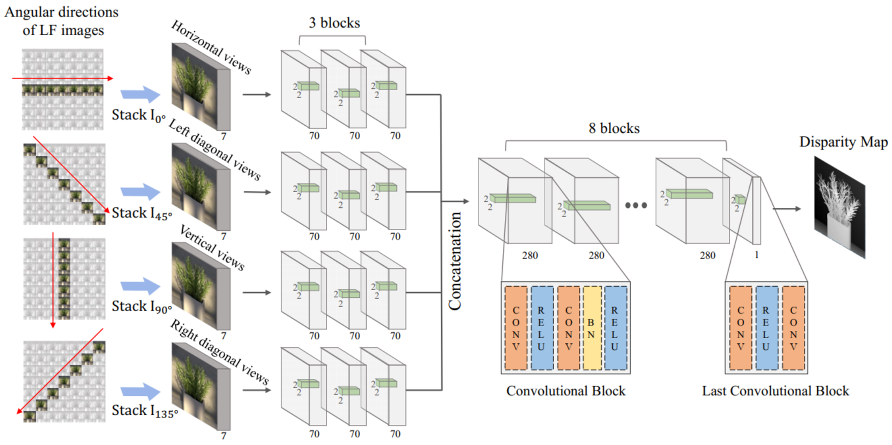


Figure 6.4: Convolutional neural network architecture used in [19]. This multi-stream network takes four viewpoints with consistent baselines into consideration. The image stacks are separately encoded at the beginning of the network and then fed through a number of convolutional blocks. (Figure taken directly from source.)

6.3 From a depth map to a point cloud

A depth or disparity map can be seen as an intensity image (see Figure 6.5). To calculate the true depth of the scene, some information about the recording device must be known. This information is often referred to as calibration data. In [31] for example, it is the focal length, sensor size, image size, focus distance, the number of viewpoints and their position. The structure of this information varies considerably based on the software used. Each point can then be described based on this additional data and the intensity from the depth map. The general algorithm can be seen in [32].

[1] Source: http://hci-lightfield.iwr.uni-heidelberg.de/benchmark/table?column-type=images&metric=badpix_0070 (used 28.1.2019)

[3] The image used is an example provided as part of the Computer Vision Toolkit (cvkit): http://dgd.vision/Tools/LFToolbox/ (used 13.5.2019)

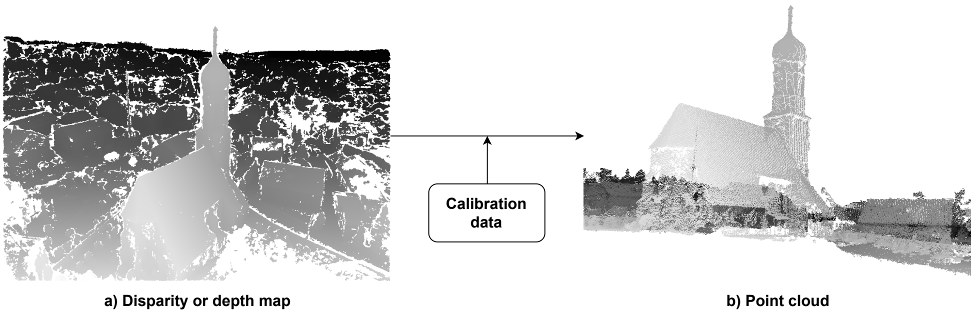


Figure 6.5: Block diagram of a conversion between a) a disparity map and b) a point cloud[3]. Additional data, sometimes called calibration, have to be provided.

6.4 Summary

In this chapter, methods which can be used to convert between a point cloud and a light field are described. They can be split into two categories — more general optimization-based approaches, based mostly on theories, and learning-based methods, which employ neural networks.

In the former case, epipolar images are mostly used to estimate a disparity or depth map, from which a point cloud can then be generated. A structure tensor is discussed, as is a spinning parallelogram operator. Line fitting approaches are also described, as are methods based on defocus and correspondence. Lastly, multi-view stereo-based techniques are mentioned, which use conventional stereo-based algorithms.

In the latter case, algorithms which use convolutional neural networks are discussed. These methods are highly dependent on how they are trained. Their output is a disparity or depth map, which can be converted to a point cloud when combined with additional information about the recording device.

Chapter 7

Conversion between Light field and Holography

The next conversion described is between Light field and Holography. These modalities can be seen as complementary [33]. Each is modelling the light of the scene differently. The former using rays, while the latter using waves [34]. A couple of methods were proposed to convert between the representations, mostly from light field to holography, even though the conversion should be possible in both ways. This chapter gives an overview the proposed conversion methods.

7.1 Coherent and Incoherent

Several difficulties must be overcome for the conversion to be possible. The main being the difference between coherent and incoherent light. Holograms are captured with the former, while light fields in general with the latter.

Coherence describes the correlation between the phases of a wave propagating through space at different points. Two types can be distinguished, being temporal and spatial coherence. Temporal coherence defines the monochromaticity of the light source, while spatial coherence represents the uniformity of the phase of the wavefront [35].

7.2 Point-source approach

The object wave computation is the first step for the generation of a CGH. The first method is one of the most commonly used ones. It uses isolated points, which are regarded as spherical light sources, to analytically describe a 3D scene. Summing the waves scattered by each point gives us the desired object wave [36].

This approach essentially uses a point cloud as an intermediary step. For that reason, it is only mentioned here, while being described in more detail in Chapter 8.

7.3 Wave-field approach

Another technique uses depth layers parallel to the hologram plane to emit a complex wave. Each layer can thus be seen as a source of light and a shield for the propagated scattered light, as can be seen in Figure 7.1. This method, also called Planar layer, is relatively simple while yielding accurate results in most cases.

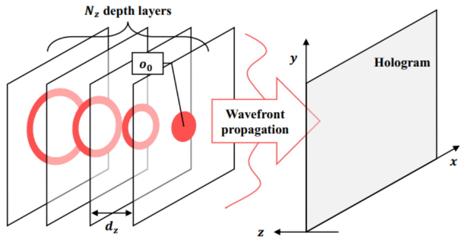


Figure 7.1: Wave-field approach (taken from [36]).

For cases when the depth of a scene is large, the number of depth layers and therefore the computational time would also need to be large. A technique based on polygonal modelling has been proposed in [37] to solve these issues. It uses a set of oriented polygons, which are seen as surface sources of light, to describe the scene. On the hologram plane, a complex light field is formed by the emitted light. One cannot use traditional propagation formulas to compute the field. Several modification were therefore proposed [36].

7.4 Multiple-viewpoint projection approach

The above-mentioned methods generate “coherent” holograms in cases where the 2D view projections of the scene are supplemented with depth maps or are synthetic 3D [34]. For real, incoherent, cases without additional depth information, other methods can be used. One of them is called Multiple-viewpoint projection, and it computes the object wave from a set of multiview images [36]. !!

An assumption is made about the number of views, which corresponds to the desired resolution. The larger the resolution, the more views are needed. The object wave is then obtained by multiplying each view by a given Point Spread Function (PSF) and summing them together. (Figure 7.2).

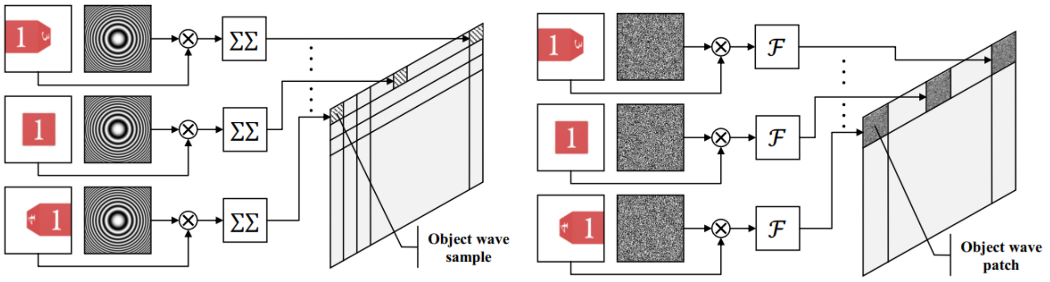


Figure 7.2: Multiple-viewpoint projections (left) in comparison to a Holographic stereogram approach (right) (taken from [36]).

7.5 Holographic stereogram approach

Another method which also uses multiview data is the holographic stereogram. Hogels, holographic segments of the final hologram, are used. Each hogel is computed independently, by multiplying each view by a random phase factor and calculating the Fourier transform of the inner product (Figure 7.2) [36]. Efficient inverse fast Fourier transform (IFFT) algorithm can be used to obtain the inside pattern of each hogel. This method is often used due to its simplicity [34].

The inverse of this operation is not as straightforward as it would seem, as the final hologram is a combination of the above-mentioned sub-holograms, as is described in [33]. The rendering of sub-image for each position has to be made. The resolution of the light field image is dependent on the maximal depth extension of the visible part of the scene.

7.6 Phase reconstruction approach

A hologram can be characterized by its amplitude and phase. The light field has information about amplitude and the direction of the light rays, which could be termed as a phase. In a coherent illumination case, the phase can then be extracted, as is shown in [38].

A breakdown of the definition of phase happens for incoherent waves. A generalized phase is thus described in [38], which directly reduces to a conventional phase for a fully coherent field. The authors show that for a microlens array the phase can be retrieved from a set of defocused images by solving the transport of intensity equation. They confirm the method for a slightly incoherent field.

An inverse operation is also possible under certain conditions. To fully characterize a 4D light field, the knowledge of the source intensity distribution together with the phase of the object is needed [38].

7.7 Wigner distribution function approach

As is mentioned in [39], the Wigner distribution function (WDF) provides a similar representation of space and spatial frequency as a light field. The WDF differs in that it also models diffraction and interference, which ties it to holography. A similar approach to the holographic stereogram can then be employed.

The method in [39] uses hogels to partition the light field into non-overlapping local patches, which can then be optimized independently, as in Section 7.5. The exception being that the random phase factor and the Fourier transform is replaced with the WDF.

7.8 Summary

The chapter concentrates on conversion methods between Light field and Holography. A point-source approach is detailed, which uses the scattering of isolated points to generate the object wave. In the wave-filed approach, the complex wave is emitted from depth layers parallel to the hologram plane. Another way is described in the multiple-viewpoint projection approach, which computes the object wave from a set of multiview images. In the phase reconstruction approach, the phase of the final hologram is retrieved from a set of defocused images. Lastly, in both the holographic stereogram approach and the approach based on the Wigner distribution function, hogels are used. The final hologram is then a combination of them.

Chapter 8

Conversion between Point cloud and Holography

The last conversion to be mentioned is between point cloud and holography. For the conversion from a point cloud, one technique is mostly used. Each point of the scene is seen as a spherical light source emitting light onto a hologram plane, as can be seen in Figure 8.1. The superposition of all the waves gives us the object wave of the hologram [36], [40], [41]. This approach can be time and resource consuming. Other methods were developed to combat the disadvantages. This chapter gives an overview of the general principle and the methods used to improve it. No methods are mentioned for conversion from a hologram to a point cloud. The conversion should be possible as the amount of data captured in a hologram is larger than in case of a point cloud.

8.1 The general principle

To compute the object wave of a hologram from a point cloud, a general principle, often called point-source [36] is used. In this approach, the selected scene filled with points is imagined as a scene filled with spherical light sources instead (Figure 8.1). The light each point casts onto a plane must be coherent [40]. The sum of all the spherical light waves gives us the hologram. The contribution of each light wave has to be limited to avoid aliasing in the CGH [36].

Occlusion is not taken into account in this principle. To handle this, two approaches are commonly used. That being an object-oriented approach and an image-oriented one.

In the former case, occlusion processing is performed to check the existence of obstacles between the object points and the pixels on the hologram plane. The hologram pixels can

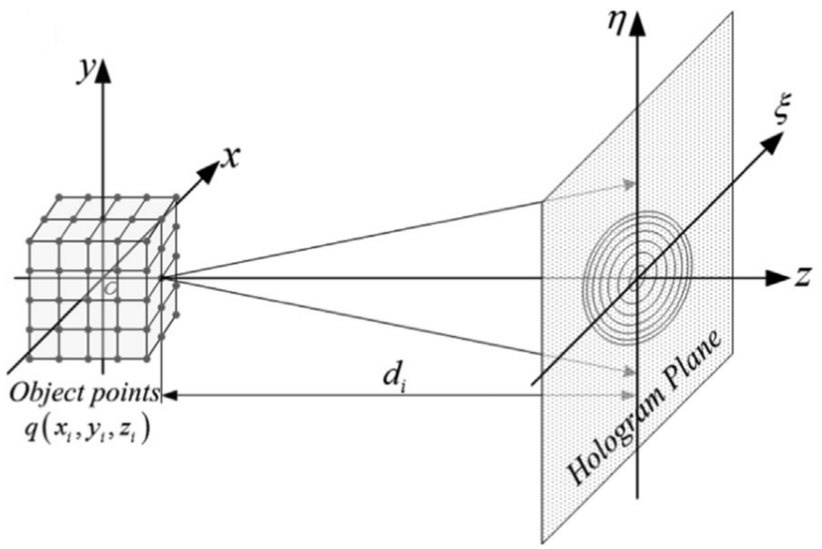


Figure 8.1: Point-source approach - spherical light source cast onto the hologram plane (taken from [41]).

be divided based on proximity into groups which then share the same set of visible scene points. This is done to reduce the computational time [36].

A ray casting technique is used in the latter case. Rays are cast from the hologram pixels onto the scene. Intersections of these waves are then considered to be spherical light sources, the contribution of which is then calculated. The hologram pixels can be divided into groups which share bundles of rays, to reduce the computational demands.

To further decrease the computational and memory demands, other methods were also developed.

8.2 Look-up table methods

The optical wave cast by each point onto the hologram plane can be described by the spatial impulse response (SIR). The sum of all of these from the individual points gives the final object wave. The number of arithmetic operations needed to generate the responses is rather large. Look-up table (LUT) methods leverage the fact that they can be pre-computed and stored in memory. This leads to faster hologram generation, excluding the time it takes to pre-compute all the points [40].

The issue then becomes high memory usage instead of computational time. This problem can be overcome with the Novel look-up table (N-LUT) method [40], which does not store all the responses. Different depth planes, in specified ranges, are instead shifted, grouped and stored together. In the reconstruction, an inverse shift is applied.

To further lower the computational time, run-length coding can be used to group points which have similar intensity and are close to each other. The newly formed regions can then be used to generate the hologram instead of individual points.

A Line scanning method has been developed. To further lower the size of the N-LUT. This approach uses "scan lines", which are radial lines of the SIRs, to lower the memory demands.

Other techniques to lower the size of the N-LUT are the Split look-up table (S-LUT) method and the Compressed look-up table (C-LUT) method.

8.3 Wavefront recording plane method

The wavefront recording plane (WRP) can, in general, be seen as a hologram located close to the object space. The wave that is emitted by each point thus covers a small section of the WRP. The process can be divided into two stages [40].

Each point emits a wave to the WRP which is bounded by its corresponding support to a certain area. A WRP pattern is thus generated as the sum of all the projected waves. This pattern is then converted into a hologram with the Fresnel diffraction equation.

The WRP is an efficient method of reduction of the computational time in cases where the depth range of the point cloud in question is rather small, or the number of object points is reduced. To reduce the latter, one can use down-sampling. The tradeoff is the degradation of image quality.

To lessen this problem, a method that uses interpolation (IWRP) is mentioned in [40]. This method uses small regions of points to generate the diffracted pattern instead of a single one. This leads to improved efficiency compared to the standard WRP.

8.4 Sub-lines method

The principle of the sub-line method is in the simplification of the hologram generation to a one-dimensional process. The whole process can be divided into three stages. A down-sampling operation and a generation of a horizontal sub-line. Interpolation with a low-pass filter to fill in the space between adjacent sub-lines, and lastly, the generation of a hologram [40].

This method can be realized by relatively simple hardware. The data size of the sub-lines can also much smaller than the hologram. A simple delta modulation can be used to compress the sub-line to a one-bit representation. Encryption can be used to secure the content. And lastly, the object image can be magnified with the use of a multi-rate interpolation process.

8.5 Wavelet shrinkage based superposition method

Another method used to reduce computational complexity is the wavelet shrinkage base superposition (WASABI) method [42]. WASABI uses wavelets to accelerate the hologram calculation.

The PSFs of the projected light sources are precomputed and transformed into the wavelet domain. A percentage of wavelet coefficients is selected and superposed in the wavelet domain. The inverse wavelet transform is then used to convert back to the space domain [42].

This method is efficient when the propagation distance between the 3D object and the hologram is small. The combination of this method with the WRP can be used to combat the drawback of both [42], [43].

8.6 Summary

The chapter centres on conversion methods between Point cloud and Holography. A general principle, in which each point of a point cloud is imagined as a spherical light source casting light onto a plane is described. The sum of these light waves is then considered the hologram. Look-up table methods, which are used to combat the issue of high computational time, are then mentioned. They leverage the fact that each arithmetic operation can be precomputed and stored. The wavefront recording plane method also reduces the computational time with the use of the Fresnel diffraction equation, but only for cases where the depth range of the point cloud is small. In the sub-lines method, the generation of a hologram is simplified to a one-dimensional process. This process can be divided into three stages, that being down-sampling, generation of a horizontal sub-line and interpolation. To reduce the computational complexity, a wavelet shrinkage based superposition method can also be applied. It uses wavelets to store precomputed information, similarly to the look-up table methods.

Chapter 9

Compression

Large amounts of data, even for today's standards, are typically captured by the three modalities. The extent of which can significantly vary depending on the modality, acquisition and conversion method. An efficient compression method is thus a necessity.

The Joint Photographic Experts Group (JPEG)[1] committee, aware of this fact, decided to develop a new standardized framework. JPEG Pleno, launched in 2015, aims to solve many tasks, one of them being compression of plenoptic images. The IEEE International Conference on Multimedia & Expo (ICME) launched two Grand Challenges to date tackling coding and compression. Specifically Light-Field Image Compression in 2016 and Point Cloud Coding in 2018. Another IEEE International Conference, this time the IEEE International Conference on Image Processing (ICIP) called a Grand Challenge in 2017 on Light Field Image Coding.

Several methods have been proposed in recent years to compress the three modalities. Some of which were responses to the above-mentioned challenges. This chapter is going to describe them briefly.

9.1 Light field compression

The acquired light field data is often highly correlated, in both the spatial and view angle dimension [44], [45]. This is a property to be targeted by compression, in the ideal case even by today's standards. That is, however, not always the case [44].

In [44], [46], [47] approaches are divided into two categories based on where in the processing pipeline the compression is applied. In [47] they are called direct and indirect.

[1] See: `https://jpeg.org/` (used 13.5.2019)

In the former case, compression is performed immediately after the acquisition of the raw sensor data. In the latter, the compressed image is created after transformation to a 4D LF. From it, sub-aperture images can be created. The acquisition process then does not affect the chosen compression method [44]. This could be crucial for backward compatibility with existing legacy standards [45].

In [48] the categorization is made based on the technique used for compression. Three kinds of methods are mentioned. First, self-similarity based LF compression [49], in which common compression standards are used. Second, pseudo-sequence based LF compression, which employs the inter prediction of common video coding standard to exploit correlations between different views. Third, dictionary learning compression methods, which, as the name implies, use dictionary learning [50] for compression purposes.

An even higher number of categories is mentioned in [5]. However, similarly to the previous paragraph, the methods can be grouped. The first way is into those that use common compression standards used in image and video processing and those that use other methods. The second is to those that use transform coding, predictive coding or pseudo-sequence coding [5], [51]. Note that there are methods that can be included in multiple groups. Because of that, the description below takes a rather general categorization.

9.1.1 Conventional image compression methods

Traditional image compression techniques can be used to compress light field images, as [44], [52], [53] show in their works. These come under the direct group of compression methods. Legacy JPEG, which uses a discrete cosine transform (discrete cosine transform (DCT)) in combination with block segmentation, can be used. The same can be said about the newer JPEG 2000, which uses a discrete wavelet transform (DWT) and is thus not backward compatible. A modification of the Embedded Zerotrees of Wavelet transforms (EZW) compression algorithm called Set Partitioning in Hierarchical Trees (SPIHT) can also be used [52]. So can be, JPEG XR, one of the newer JPEG standards. The comparison in [52] and [44] shows, that in terms of the compression of raw light fields, JPEG 2000 together with SPIHT achieve the best results. Beating out the block coding transform of JPEG XR. The same results were shown in [46], where compression algorithms of the ICME 2016 Grand Challenge were evaluated.

While they showcase lower computational complexity [52], than the later-described methods, compatibility can be an issue. Mainly with the wavelet-based encoders. Perra et al. [54] also show, that a better understanding of the plenoptic signal structure can lead to even higher compression ratios, as none of the above encoders is adapted for the use with plenoptic images.

9.1.2 Multiview methods

A solution could be the use of multiview image compression methods, as they can better exploit the redundancies in spatial and viewpoint dimensions [55]. These classify as indirect, as they no longer apply directly onto the raw image. Two methods that both use this technique and outperform the more traditional JPEG 2000 are proposed in [56]. The first one uses a wavelet transform, while the other uses distributed source coding (DSC) [54].

In [57] a two-dimensional level based prediction scheme is proposed to control the structure within frames of the multiview sequence. In [58] additional information about the depth is added to the encoding process. In both cases, a High Efficiency Video Coding (HEVC) video encoder is used in the compression pipeline.

9.1.3 Conventional video compression methods

HEVC encoding is often much better at exploiting the redundancies in the image than JPEG standards, lowering the compassion rations in the process. Different profiles can be used to tweak the encoding process further. In [59] low delay coding is used, which codes all except the first frame as B frames. A similar approach is taken in [60], where only the first frame is coded as I frame and all others as P frames.

As a study in [61] shows, the application of HEVC on light field images depends besides its configuration also on the specific arrangement of images. An optimal sequence should be found, to maximize performance [45].

9.1.4 Pseudo-sequence methods

Pseudo-sequences are a technique used in most of the proposed compression methods. A pseudo-sequence falls under the indirect category. That is because the raw image is first transformed into a 4-D representation, from which a sequence of sub-aperture images is extracted. These sub-images can then be arranged into a number of different sequences, based on many factors. The main one is the similarity between adjacent views. Scan orders can thus have a variety of shapes, such as a zig-zag, spiral, raster, Hilbert, rotation and others [62], [63]. A video sequence is then composed, which can be encoded with a standard video codec, such as Advanced Video Coding (AVC) or HEVC [60], [64], [65].

Different scan orders have different effects on the final compression ration. However, that is not the only modification that can be done to increase the effectiveness of the compression. In [62], for example, a CNN-based angular super-resolution approach is used to explore the inter-view correlation before a pseudo-sequence is applied. An orthogonal projection is firstly formed in [66]. In [63] the images are sparse-sampled. Multiview video coding as in [59] or a JEM encoder as in [67] can be used instead of HEVC. Other methods that include a pseudo-sequence are [68]–[72].

9.2 Point cloud compression

As the first point cloud acquisition and processing methods were proposed earlier than in the case of light fields, so was the case for pioneering works in compression of this modality. As is noted in [73], one of the methods in question uses a somewhat unusual technique. Specifically, a shape-adaptive wavelet coding is used in [74]. Later works, fuelled mainly by the research done in autonomous driving, use many different approaches. Some are similar to the light field compression methods, in that they use traditional image compression methods. Other venture more into the works done in computer vision and 3D modelling. Grouping is thus quite problematic.

9.2.1 Tree-based compression

In [75] and later in [76], [77] and [78], octrees are used in the encoding process. They can be used for geometry coding, as noted by [73], or the encoding of color information of RGB point clouds [75], as noted in [79].

The combination with other techniques is also possible, as is shown in [77]. It that case a real-time octree-based coder is used to exploit temporal redundancies on an octree byte stream. Encoding of a dynamic point cloud is thus possible. An extension of this framework is then proposed in [78], which includes a standard image codec for the encoding of colour information.

9.2.2 Transform-based compression

For efficient compression of colour attributes a Graph Transform method was introduced in [80]. An extension for the use on dynamic point clouds was then proposed in [81]. This method, also shown in [82], taps into the previously mentioned tree-based approaches, as it uses a time-varying point cloud based codec which predicts graph-encoded octree structures of adjacent frames. These techniques achieve top performance [73].

A similar approach is proposed in [83], based on Gaussian Process Transforms (GPTs). The point colours are assumed to be samples of a stationary Gaussian Process, which is defined over an infinite 3D space. The statistics of the points are then modelled. After a few derivations, a Karhunen-Loève transform (KLT) is acquired. The authors, however, call the KLT a GPT, to distinguish it from the above mentioned Graph Transform.

Lastly, a region-adaptive hierarchical transform (RAHT) coder has been proposed in [84] for the compression of the colour attribute. The transform itself resembles an adaptive variation of a Haar wavelet. It is then combined with arithmetic encoding. An octree scanning method is used for the geometry encoding part of the point cloud. This method was shown to be computationally efficient.

That is why variations of the RAHT approach are also used in [73]. Four techniques, to be exact, are proposed. Two of which extend the transform over the sphere that represents the plenoptic function. Two other involve a combined RAHT with 1D transforms. From all four the RAHT-KLT variation showcases the best overall performance.

9.2.3 MPEG standards for point cloud compression

The Moving Picture Experts Group (MPEG) in 2017 proposed a point cloud compression (PCC) technology, which should be delivered in 2020 as an ISO standard[2]. This future standard consists of three main categories: Lidar point cloud compression (L-PCC), surface point cloud compression (S-PCC) and video-based point cloud compression (V-PCC), and a fourth called geometry-based point cloud compression (G-PCC) [79].

As the name implies, L-PCC is designed to compress Lidar point clouds. These display highly irregular sampling. An octree-based method is thus used to compress the geometry information. A Level-Of-Detail (LoD) structure is then built and further used for prediction purposes [79].

S-PCC codec aims to efficiently compress static point clouds. These usually express highly sampled and therefore highly detailed scenes. The encoder is composed of several

[2] See: `https://mpeg.chiariglione.org/standards/mpeg-i/point-cloud-compression` and `https://www.iso.org/committee/45316.html` (used 20.5.2019)

modules, which in turn follow these steps: Voxelization, Block determination, entropy encoding of both blocks and vertices, spatial transform and quantization. A similar process is then applied in reverse for the decoding [79].

Existing video codecs, such as AVC, HEVC and AV1, are to be utilized in V-PCC for compression of geometry and texture information of dynamic point clouds. Points of the point cloud are to be projected into a sequence of 2D frames. To generate the final V-PCC, additional metadata are to be multiplexed with the video generated bitstream. The amount of metadata in comparison to the video sequence is relatively small.

9.2.4 Other point cloud compression methods

A slew of other methods is mentioned in [85]. The usage of standard image coding methods is proposed in [86] and [87]. In the former case, the acquired data is to be split into height maps before compression. In the latter case, the raw data is first converted into range images, before image or video coding algorithms are employed for encoding. Tu et al. in [88] combine image compression methods with a Simultaneous Localization and Mapping (SLAM) based prediction. This method is designed to work in autonomous driving systems.

An adaptive radial distance prediction method that outperforms conventional image and video coding algorithms is used in [89]. Both [90] and [91] propose the use of a Random Sample Consensus (RANSAC) method. While the former uses RANSAC to extract planes from a point cloud, as a part of its lossy compression scheme. In the latter, fitting plane parameters are calculated using RANSAC, after the point cloud is decomposed into blocks. DCT then performs the compression of the colour information.

A clustering method inspired by a DMM technique adopted in [92] is proposed in [85]. This approach is used to remove spatial redundancy of point cloud data. Besides that, residual compression with ZStandard, LZ4, LZ5, Lizard, Deflate, BZip2, JPEG and JPEG 2000 is also explored in [85].

9.3 Holography compression

Firstly it must be noted, that in the case of holograms, even though they contain the information about the entire three-dimensional scene, the information is completely scrambled. The compression of such data is not as easy as was the case for 3D or multi-view data. These, as was shown earlier, exhibit high redundancy and therefore better predictability.

The efficiency of the proposed techniques thus highly depends on how they can exploit the structure of the scene [93]. Summaries of the bellow mentioned approaches can be seen in [94] and [95].

The earliest approaches encoded holograms generated with phase-shifting interferometry with conventional lossless coding methods, such as LZ77, LZW and Huffman encoding [93], [96]. In [97], the Linde-Buzo-Gray algorithm is applied to perform scalar and vector quantization [98]. No transformation is used in either approach and both are applicable on raw data.

9.3.1 Wavelet-based approaches

The works in [99] and [95] explore the use of the JPEG 2000 standard for the compression. In [100] the efficiency of JPEG-based encoders is further analyzed. JPEG-OPT, which similarly to the legacy JPEG standard uses DCT and guarantees backward compatibility with it, is compared to QT-L, JPEG 2000, SPIHT and HEVC algorithms.

Other wavelet-based approaches or adaptations of the common ones are also discussed. Bang et al. in [101] propose an algorithm that uses a dual wavelet transform to track the direction of the fringes. To better align the transform bases with the directional characteristics of the fringe pattern a directional adaptive DWT (DA-DWT) method is explored in [102]. In [103] a vector lifting scheme is proposed to design specific wavelets dedicated for the use in hologram compression.

9.3.2 Content-aware approaches

The works of Onural et al. [104] on the link between the mother wavelet function and depth in the Fresnel transform, was later explored in [105]. This led to the introduction of Fresnelets, which are wavelet-like functions constructed from B-spline wavelets by the use of the Fresnel transform. Which then led to further research on the topic, resulting in the approach of Darakis et al. [106]. In their work, the data, respectively the depth of the scene, is analyzed with specific basis functions and encoded with SPIHT.

In [107] the use of AVC and HEVC codecs is explored on computer-generated holograms. AVC is also used in [108], for the encoding of numerically reconstructed sub-holograms. This method is later extended in [109] with the use of directional scalability and in [110], [111] with Gabor/Morlet wavelet dictionaries and a matching pursuit approach.

As is noted in [93] further research has to be still done in this field for better compression to be achieved. A better understanding of the hologram structure and its features will likely have to be made before the most essential information can be effectively extracted.

9.4 Summary

In this chapter, different compression techniques are mentioned for light field, point cloud and holography. There are some similarities between the approaches, for example, state-of-the-art video codecs such as AVC and HEVC can be used for the compression of all three modalities.

Direct and indirect categories are mentioned, based on the position of compression in the processing timeline. They can further be divided into self-similarity based LF compression methods, in which common compression standards are applied, pseudo-sequence based schemes, which leverage redundancy between individual views of a light field image, and lastly, dictionary-based methods, which use dictionary learning for compression purposes.

Point clouds can be compressed in many ways. Both the positional information and the embedded data must be compressed. Octree-based encoders can encode both geometry and colour information. For efficient compression of colour, a Graph Transform can also be used. So is the case for approaches based on Gaussian Process Transforms or a region-adaptive hierarchical transform. MPEG proposed a Point Cloud Compression technology. Video encoders, still image coding methods and other algorithms were also suggested for point cloud compression.

The compression of holograms is a difficult task. Conventional lossless coding methods, such as LZ77, LZW or Huffman coding, were used. Wavelet-based compression techniques are also proposed. A link between the mother wavelet and depth in the Fresnel transform was found, which led to the introduction of Fresnelets. These are then also used for compression purposes, together with SPIHT. Video coding techniques were also explored. However, further research in this field has to be still done, before an efficient compression method is found.

Chapter 10

Datasets and software

There are several publicly available datasets for each modality. The existence of which is important for implementation, testing and training purposes of the previously discussed conversion and compression methods and future ones. To satisfy a wide range of utilization, each dataset has different properties, be it size, file types, number of captured scenes, type of scenes, and other. Different software can then be used to view or manipulate the data. This chapter gives a brief overview of both the datasets and software.

The datasets can be divided into two categories based on how they were created. That being those that capture real scenes, by the means of cameras or sensors and those that are computer generated. A table containing a few datasets can be seen in Appendix A[1].

10.1 Real scenes

These datasets are acquired by the more traditional means, using digital single-lens reflex cameras (DSLRs), light field cameras or a set of sensors, for example. The scenes they capture render the real world, which provides some advantages and disadvantages.

The lack of ground truth depth data may be a problem for conversion methods that use a depth map as an intermediate step. Having the ground truth available makes it easier to compare the different algorithms. On the other hand the benefit is that reality is being portrayed, instead of a resemblance of it.

The datasets that are based on real scenes include the light field datasets from École polytechnique fédérale de Lausanne[2] and Stanford University[3] and the point cloud dataset

[1] Available also electronically: https://docs.google.com/spreadsheets/d/1OzMUTy9Ll_5M59dbuyTsNs6mkmzXOJdyDXAfBWyCWFM/edit?usp=sharing

[2] Source: https://mmspg.epfl.ch/EPFL-light-field-image-dataset (used 28.1.2019)

[3] Source: http://lightfields.stanford.edu/mvlf/ (used 28.1.2019)

from the University of São Paulo[4]."

10.2 Synthetic scenes

Synthetic scenes, also often called Computer generated scenes, make it possible to better test a specific aspect of the design algorithm. Methods which focus on a property, occlusions, for example, will often use synthetic scenes for benchmarking purposes, rather than their real equivalent.

The most popular software to generate a synthetic scene is Blender[5], with a close second being POV-Ray[6]. Both of them are free to use, with the former being also open source.

10.3 Software

Different programs can be used to view and modify the acquired data. In the case of synthetic scenes, the software used to generate them is the obvious choice. In most cases that means either the above mentioned Blender or POV-Ray. Otherwise, it depends on the file type used. MATLAB, for example, has numerous publicly available toolboxes, which can handle different types of data. A light field toolbox[7], a holographic toolbox (HoloRec3D[8]) and a point cloud toolbox[9], just to name a few. Another open source software is MeshLab[10], which can be used to view point clouds. Tools such as Computer Vision Toolkit (cvkit)[11] and Point Cloud Maker 11 (PCM11)[12] were developed to both view depth/disparity maps and convert them to point clouds, as long as you supply them with specific additional information. To work with digital holograms HoloPy[13] can be used. A larger list of publicly available solutions can be seen in the Appendix C[14].

[4] Source: http://uspaulopc.di.ubi.pt (used 28.1.2019)

[5] Source: https://www.blender.org/ (used 25.11.2018)

[6] Source: http://www.povray.org/ (used 25.11.2018)

[7] Source: http://dgd.vision/Tools/LFToolbox/ (used 25.11.2018)

[8] Source: http://labh-curien.univ-st-etienne.fr/wiki-reconstruction/index.php/Main_Page (used 25.11.2018)

[9] Source: https://github.com/pglira/Point_cloud_tools_for_Matlab (used 25.11.2018)

[10] Source: http://www.meshlab.net (used 25.11.2018)

[11] Source: http://vision.middlebury.edu/stereo/code/ (used 25.11.2018)

[12] Source: https://3dstereophoto.blogspot.com/2016/09/point-cloud-maker-11-pcm11.html (used 25.11.2018)

[13] Source: https://holopy.readthedocs.io/en/latest/ (used 25.11.2018)

[14] Available at: https://docs.google.com/spreadsheets/d/19CEToXuiZf6gn-HTP_ODAoztXHbjgAMQEJ468axwrco/edit?usp=sharing

Chapter 11

Disparity from compressed light fields

For the practical part of this work, a number of procedures described in the previous chapters are offered. Some problems are not sufficiently analyzed in scientific publications. One of which is the effect compression has on disparity estimation.

Raw light field images are mostly bigger than traditional photographs. It is thus expected that some form of compression (see Chapter 9) will have to be applied for storage purposes. If the compression technique chosen is lossy, some amount of distortion is going to occur, which is intern going to have some effect on all the possible subsequent light field applications. It would thus be beneficial to know the effects of lossy compression methods on the applications.

In this chapter, the effects of light field compression on disparity estimation are analysed. Several state-of-the-art image and video compression methods are used in combination with several light field to disparity map conversion techniques. Objective metrics are then used for evaluation. Matlab was used for simulations.

11.1 Overview

The evaluation process is made of several steps, as shown in Figure 11.1. At first, the desired light field data is loaded and compressed with the user-defined compression method (see 11.3). For the middle view of the LF image the Mean Square Error (MSE) is computed as follows:

$$MSE = \frac{1}{xy}\sum_{i=1}^{x}\sum_{j=1}^{y}[I_c(i,j) - I_r(i,j)]^2 \tag{11.1}$$

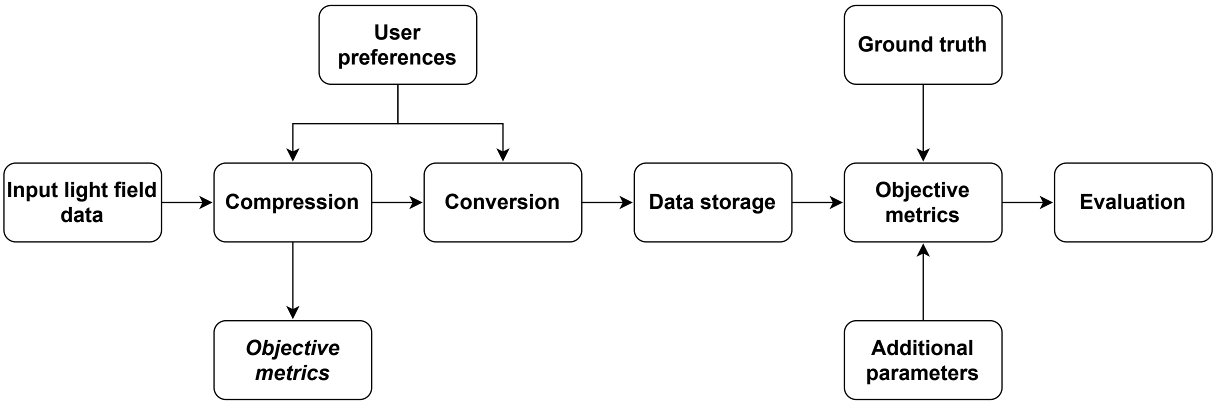


Figure 11.1: Block diagram of the evaluation process.

where (x,y) represent the dimensions of the image, I_c is the compressed image and I_r is the reference image. The PSNR is then computed as:

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad [\text{dB}] \qquad (11.2)$$

where MAX is the maximum value of the image, for example for an 8-bit image that would be 255. These values are stored for later evaluation. The compressed image is loaded with the use of the Light Field Toolbox[1]. A user-chosen algorithm (see 11.3) is then used to convert the data to a disparity map. The output map together with additional information, such as the size of the input images and the run time, are stored for evaluation purposes. This process repeats with a user-defined compression step for a user-defined number of conversion methods. After the procedure is finished, objective metrics are calculated.

11.2 Input data

Images from the synthetic light field dataset (Figure 11.2) from [31] were used as input because they provide both ground truth data and other necessary data for the evaluation. The images were chosen based on their properties, be it noise, depth range, occlusion or overall complexity, to represent many real-world scenarios.

11.3 Compression and conversion

The individual views of the light field images were either encoded with purely image based compression tools or they were organized in a row by row pseudo-sequence and encoded

[1] Source: `http://dgd.vision/Tools/LFToolbox/` (used 7.5.2019)

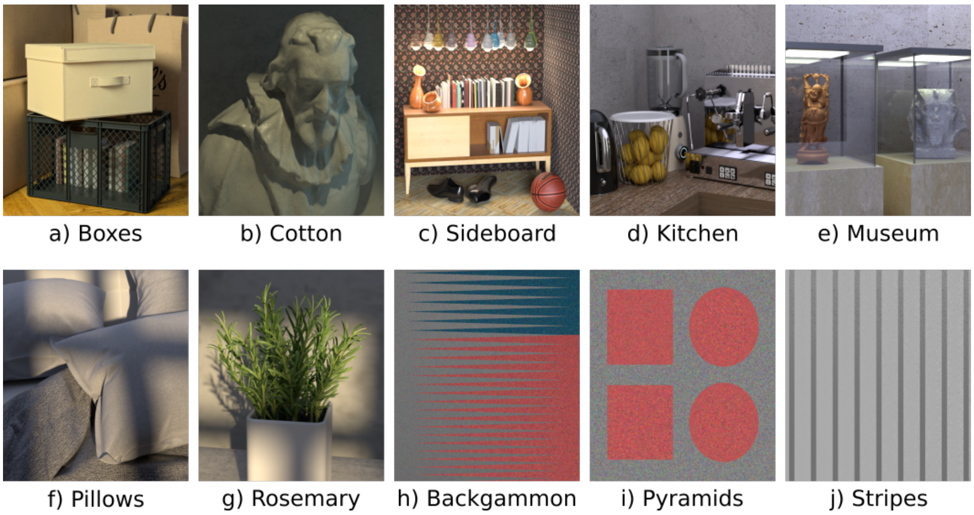


Figure 11.2: A selection of light field images from the synthetic light field dataset from [31]. It is a combination of photorealistic scenes and scenes with specific controlled features.

with video compression algorithms. In the former case, the tools selected are JPEG (based on the DCT transform), JPEG 2000 (based on the DWT transform), and Better Portable Graphics (BPG) (based on a subset of HEVC)[112]. In the latter case, AVC (x264), HEVC (x265), VP9 and AV1, were selected. These specific tools were selected as a representation of both the state-of-the-art and the most widespread. The `imwrite` and `imread` Matlab functions were employed for compression. The former was used to write image data to JPEG and JPEG 2000 files, while the latter was used to load all the decompressed files into Matlab. Otherwise the `system` function from Matlab was used in combination with the BPG executable [112] or the open source FFmpeg tool [2]. The amount of compression in the video codecs is set with Constant Rate Factor (CRF), which is the default encoding mode for FFmpeg. In comparison to the Constant Bit Rate (CBR) mode, CRF aims to encode the sequence of frames so that a constant perceived quality is achieved. The range of CRF is 0-51 for AVC and HEVC and 0-63 for VP9 and AV1. In both cases 0 represents the best possible quality, with 51 or 63 being the worst. The user can however choose quality on a scale of 0 to 100 in the evaluation script as it coincides with the quality parameter in the `imwrite` Matlab function. The values are linearly normalized and then rounded to the nearest integer based on the chosen compression[3]. Furthermore, the Matroska Multimedia Container was used to encompass the data, as it can be used with all the video encoders mentioned above.

The conversion tools selected are LF [113], LF_OCC [114], CAE [27], [115], SPO [21] and EPINET9×9 [19] (for descriptions see Chapter 6). Only the last of them is based

[2] Source: `https://ffmpeg.org` (used 11.5.2019)

[3] The norma

on neural networks, which makes its run time noticeably smaller in comparison with the rest. For this reason and because it ranked towards the very top of the 4D Light Field Benchmark[4], it was chosen as a reference and was evaluated on all the input images. The other algorithms were tested only on a subset of input data. Two things also must be noted. One being that the LF_OCC algorithm did not function properly, as it was able to distinguish between parts with similar disparities, but was not able to assign correct values to them. Because of this fact, the algorithm will be mentioned sporadically. The second being that no evaluation was done on the LF algorithm and thus no results will be mentioned from it, because of its fairly lengthy run time, of approximately 1300 seconds per estimation[5].

11.4 Objective metrics

The PSNR metric (11.2) was used together with metrics from [31] for the objective evaluation. Their names and descriptions can be seen in Table 11.1.

Table 11.1: Overview of the used objective metrics [31].

BadPix	Measures the number of pixels which are above a certain error threshold on masked areas.
Bumpiness	Quantifies the smoothness of planar continuous surfaces.
Bumpiness slanted	Quantifies the smoothness of non-planar continuous surfaces.
Foreground Thinning (FG_Thinning)	Measure how many pixels are closer to the background than the foreground, but should be estimated as foreground instead. It is defined at occlusion boundaries on masked areas.
Foreground Fattening (FG_Fattening)	Measure how many pixels are closer to the foreground than the background, but should be estimated as background instead. It is defined at occlusion boundaries on masked areas.
Thinning	Measure how many pixels are closer to the background than the foreground, while exceeding a threshold value. It is defined at occlusion boundaries on masked areas.
Fattening	Measure how many pixels are closer to the foreground than the background, while exceeding a threshold value. It is defined at occlusion boundaries on masked areas.
Mean Square Error (MSE_M)	Quantifies the mean square difference between a specific image and a reference on masked areas.
Peak Signal-to-Noise Ratio (PSNR)	Ratio between the maximum power of an image and the power of an error image, where the error image is the difference between a specific image and a reference.

However, some adjustments had to be made, as the authors implemented their evaluation algorithm in Python rather than in Matlab. For example, the authors use Scharr

[4] Source: `http://hci-lightfield.iwr.uni-heidelberg.de/benchmark/table?column-type=images&metric=badpix_0070` (used 7.5.2019)

[5] It has to be noted that the CAE algorithm is shown to have a longer average run time per estimation (see Table 11.8) than the LF algorithm; however it performed better in the 4D Light Field Benchmark.

transform for edge detection in the Bumpiness metric, which is not present in Matlab. Thus the Sobel transform was used instead. None of the adjustments deviates from the equations in [31], the differences are only in the implementation of them, which does not necessarily matter in our case as there is no comparison made with the results of [31] and consistency is maintained within our evaluation. The Bjøntegaard metric[6], which is used for the computation of the average distance between two R-D curves, was also added.

11.5 Results

There are four scenes with controlled parameters in [31], which they called stratified. These are designed to test algorithms on specific features, which can occur in complex real-world scenes. Figures 11.3 and 11.4 show artifacts which can arise from compression.

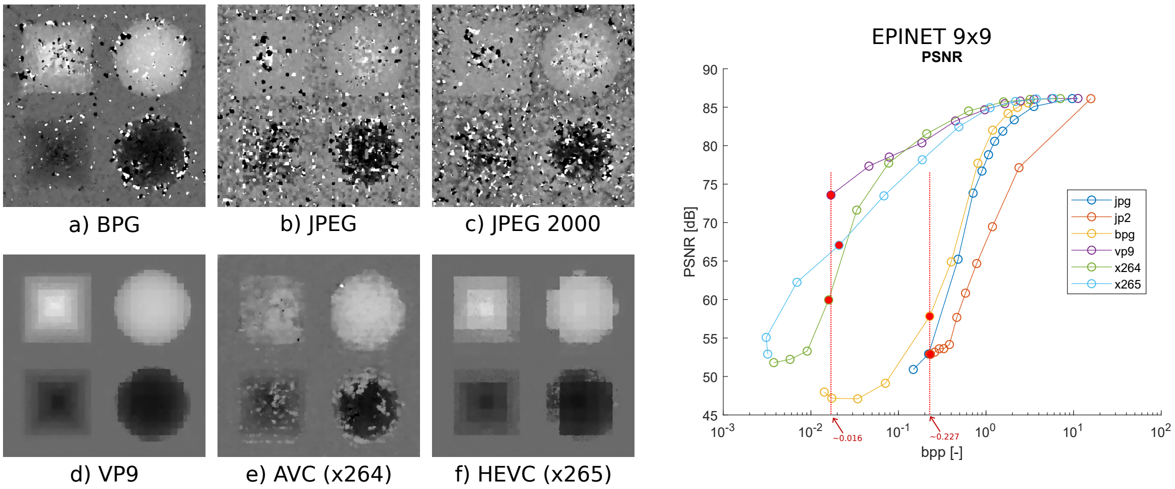
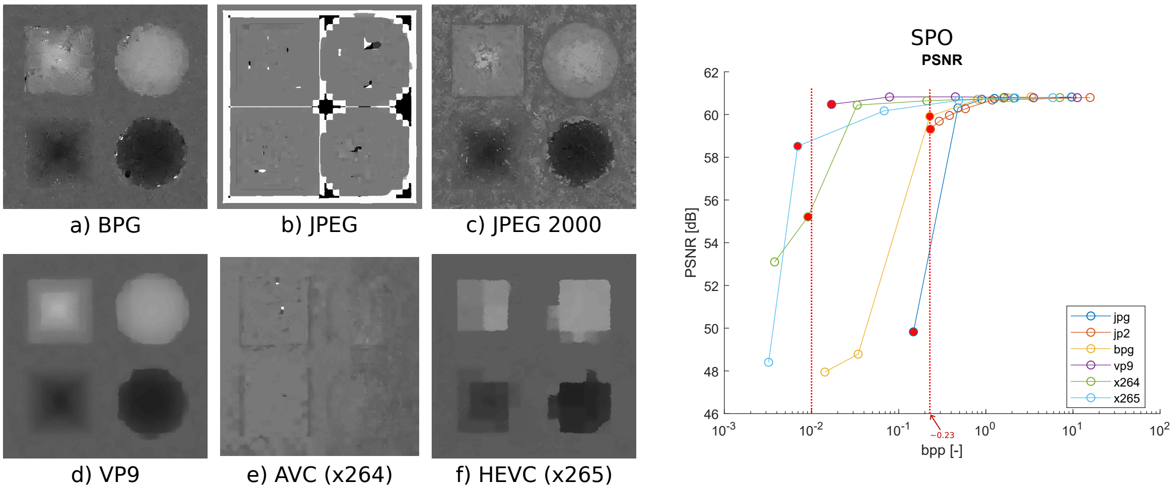

a) BPG
b) JPEG
c) JPEG 2000
d) VP9
e) AVC (x264)
f) HEVC (x265)

Figure 11.3: Effects of compression algorithms on distortion of disparity for the EPINET [19] algorithm. Images of similar bit rate were chosen for the top and bottom row, so that the distortions are visible and comparable. The images chosen are represented by red dots on the PSNR plot, with the dotted lines representing the approximate bit rates.

In both cases, the images were compressed to the approximately same bits-per-pixel value, so that the difference between both compression and conversion algorithms can be seen. In Fig. 11.3, where the EPINET 9×9 algorithm is used, the still image coding techniques exhibit mostly impulse noise, the amount and position of which varies. In contrast, the tested video coding methods display distinctly different cases of distortion, blockiness for VP9 and HEVC, smaller and less consistent bumps for AVC.

[6] Source of the Matlab implementation used: `https://www.mathworks.com/matlabcentral/fileexchange/41749-bjontegaard-metric-calculation-bd-psnr` (5.5.2019)



a) BPG b) JPEG c) JPEG 2000

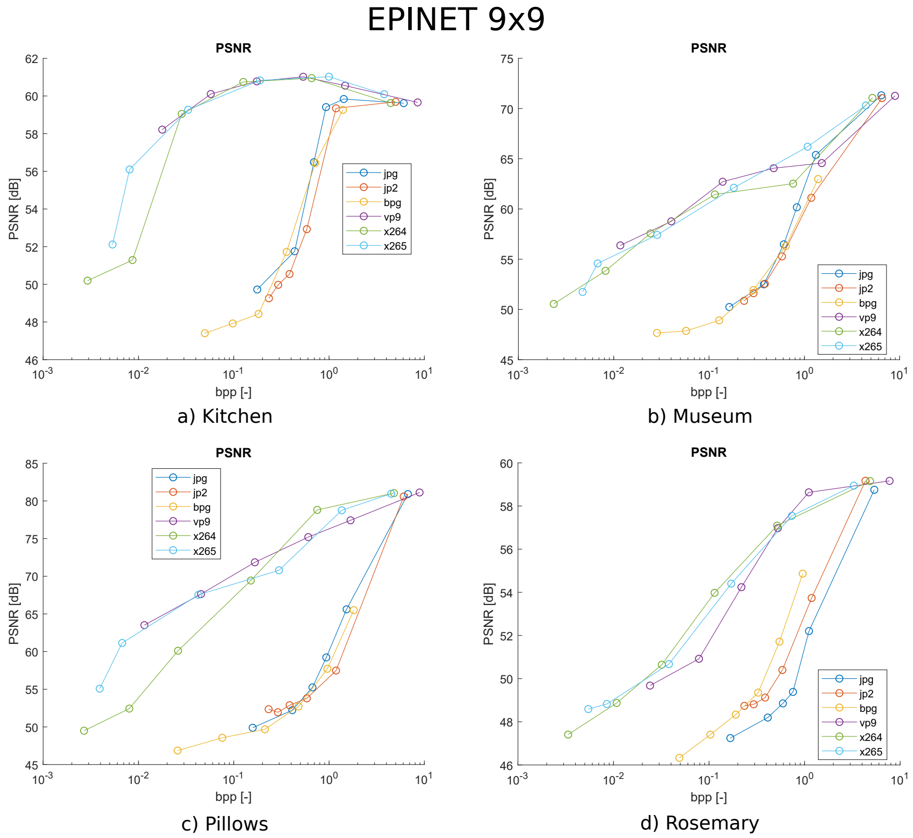
d) VP9 e) AVC (x264) f) HEVC (x265)

Figure 11.4: Effects of compression algorithms on distortion of disparity for the SPO [21] algorithm. Images of similar bit rate were chosen for the top and bottom row, so that the distortions are visible and comparable. The images chosen are represented by red dots on the PSNR plot, with the dotted lines representing the approximate bit rates.

The same behaviour does not, however, apply for the SPO algorithm, as shown in Fig. 11.4. SPO is not based on neural networks, and thus compression has a different effect on it than the EPINET algorithm, which is noticeable both from the disparity maps and from the PSNR plot. Impulse noise is not as prevalent, replaced instead with bumps and general surface roughness. For JPEG compression, one can observe that the algorithm is no longer able to reconstruct almost any depth, picking up only the shapes of the objects in the scene. Similar behaviour can be observed for the AVC codec. The HEVC algorithm displays very similar distortion as in Fig. 11.3. VP9 is the least affected of the tested methods for this specific scene, as the deformations are less periodical, smoother and mostly on the edges of objects. The impact of compression on the SPO algorithm is not as substantial in case of distortion as on EPINET. This is because SPO is not as precise in assigning disparity values as EPINET. The maximum PSNR values are thus smaller.

11.5.1 PSNR metrics

Next observed was the effect compression has on other scenes. Figure 11.5 shows the PSNR for four photorealistic scenes when the EPINET algorithm is used. In each case, trends can be seen. As expected video codecs showcase higher compression ratios in comparison with purely image based compression techniques, as they use both spatial and temporal compression. None can be, however, called superior, as all behave slightly differently, based on the scene. For the kitchen scene, an improvement in PSNR can even be seen when compression is applied. This is due to a reflective surface in the scene, which when compression is not applied complicates the calculation, as it behaves like a mirror.



a) Kitchen

b) Museum

c) Pillows

d) Rosemary

Figure 11.5: PSNR performance of compression techniques for the EPINET conversion algorithm, measured on a set of photorealistic scenes.

However, when the scene is slightly distorted, the conversion algorithm can better pick up depth cues and thus improve the overall PSNR. Otherwise, an almost linear regression can be observed for these scenes.

Purely spatial compression algorithms also exhibit similar, in cases almost the same, behaviour, in terms of PSNR. In both the Museum and Pillows scenes, the overall difference between the techniques is fairly small. However, it can be seen that the generally superior compression methods, BPG and JPEG 2000, are not only not outperforming the older JPEG standard, in some cases, the older JPEG standard reaches higher values of PSNR. Despite this, the last image illustrates, that this behaviour is not consistent. For the Rosemary scene, a clear difference between the generations is presented, with the older JPEG performing significantly worse than both BPG and JPEG 2000. What should be pointed out, is the plot in Fig. 11.3, where it is the JPEG 2000 standard that performs worse.

Table 11.2 further showcases the effects of compression on conversion algorithms using the Bjøntegaard metric (see 11.4). The R-D curve of each compression algorithm was taken, represented in rows, and compared to other ones, represented by the columns. The

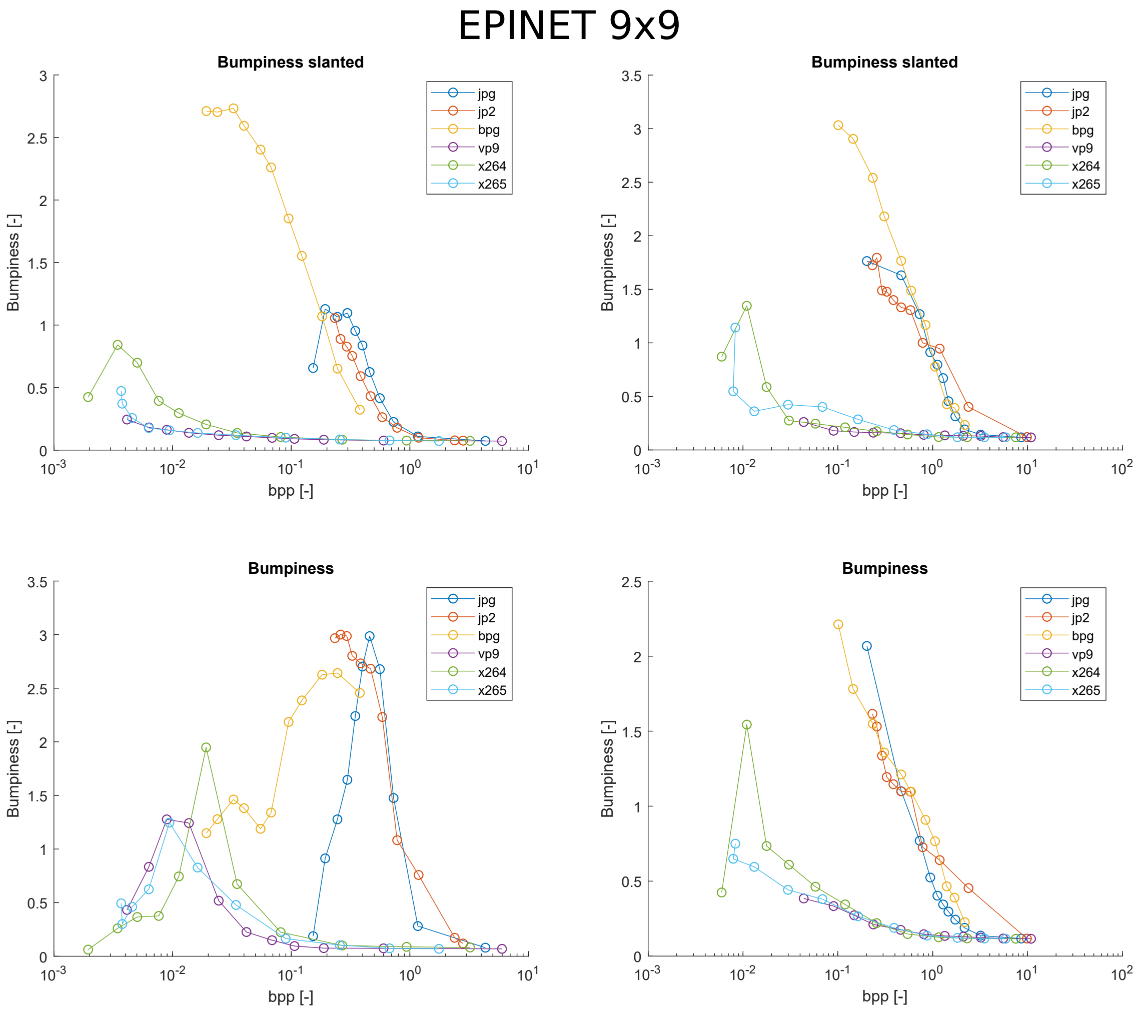
Table 11.2: Comparison of PSNR by the Bjøntegaard metric (BD-PSNR), which computes the average distance between two R-D curves, for three different scenes and conversion algorithms. Higher values are better.

		BD-PSNR [dB]																	
		EPINET						SPO						CAE					
		JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265
BOXES	JPG	X	3.06	0.85	-3.77	-4.03	-3.76	X	-0.44	-0.90	-1.54	-1.52	-1.48	X	-0.12	-0.97	-0.95	-0.74	-0.78
	JP2	-3.06	X	-0.89	-6.19	-6.50	-6.40	0.44	X	0.52	-0.60	-0.59	-0.52	0.12	X	-0.28	-0.56	-0.35	-0.41
	BPG	-0.85	0.89	X	-9.35	-9.50	-9.36	0.90	-0.52	X	-3.58	-3.52	-3.67	0.97	0.28	X	-1.77	-1.73	-1.87
	VP9	3.77	6.19	9.35	X	-0.04	0.00	1.54	0.60	3.58	X	0.11	-0.10	0.95	0.56	1.77	X	0.12	-0.03
	x264	4.03	6.50	9.50	0.04	X	-0.22	1.52	0.59	3.52	-0.11	X	-0.46	0.74	0.35	1.73	-0.12	X	-0.26
	x265	3.76	6.40	9.36	0.00	0.22	X	1.48	0.52	3.67	0.10	0.46	X	0.78	0.41	1.87	0.03	0.26	X
SIDEBOARD	JPG	X	3.31	1.28	-6.42	-7.05	-7.47	X	-1.04	-0.37	-2.27	-2.34	-2.32	X	-0.31	-1.33	X	X	X
	JP2	-3.31	X	-0.30	-9.08	-10.10	-10.46	1.04	X	1.34	-0.91	-0.97	-0.85	0.31	X	-0.67	X	X	X
	BPG	-1.28	0.30	X	-12.11	-12.53	-12.43	0.37	-1.34	X	-4.19	-4.16	-4.08	1.33	0.67	X	X	X	X
	VP9	6.42	9.08	12.11	X	-0.24	-0.22	2.27	0.91	4.19	X	0.16	0.07	X	X	X	X	X	X
	x264	7.05	10.10	12.53	0.24	X	-0.38	2.34	0.97	4.16	-0.16	X	-0.56	X	X	X	X	X	X
	x265	7.47	10.46	12.43	0.22	0.38	X	2.32	0.85	4.08	-0.07	0.56	X	X	X	X	X	X	X
PYRAMIDS	JPG	X	6.47	-2.22	-9.27	-10.83	-9.82	X	-0.59	-1.44	-1.66	-1.79	-1.36	X	-0.30	-0.79	-1.26	-1.65	-1.29
	JP2	-6.47	X	-10.16	-12.87	-15.74	-15.05	0.59	X	-0.43	-0.29	-0.34	0.25	0.30	X	-1.40	-0.40	-1.00	-0.63
	BPG	2.22	10.16	X	-18.57	-16.83	-16.04	1.44	0.43	X	-3.92	-3.82	-4.20	0.79	1.40	X	-5.07	-4.94	-5.49
	VP9	9.27	12.87	18.57	X	1.43	2.28	1.66	0.29	3.92	X	0.28	-0.01	1.26	0.40	5.07	X	0.29	-0.10
	x264	10.83	15.74	16.83	-1.43	X	-0.39	1.79	0.34	3.82	-0.28	X	-0.38	1.65	1.00	4.94	-0.29	X	-0.85
	x265	9.82	15.05	16.04	-2.28	0.39	X	1.36	-0.25	4.20	0.01	0.38	X	1.29	0.63	5.49	0.10	0.85	X

values in the table specifically express the average distance and thus the average gain between the two compared R-D curves. In this case, the higher the values, the bigger the average PSNR difference. For each conversion technique in the table, certain trends can be seen. Be it the effect of temporal coding, which creates a performance gap between coding algorithms, or the similarity between purely image-based compression schemes and video codecs. Furthermore, it may seem BPG in comparison to video compression algorithm performs worse than both JPEG standards. However, when looking at both Tab. 11.2 and Fig. 11.5, it is clear that BPG is rated worse, because of its ability to compress the input image further than both JPEG standards, thus increasing the surface between the measured R-D curves.

11.5.2 Bumpiness metrics

The tested algorithms may exhibit roughness on surfaces that should be smooth. This can be measured with the Bumpiness metric, both for planar and non-planar continuous surfaces. Figure 11.6 details the case for two photorealistic scenes, Cotton and Sideboard. In the former case, a sudden decrease in bumpiness can be seen for lower bit rates. This is due to the design of the Bumpiness metric, in which a transform is used for detection of edges and thus roughness in the image. However, when the bit rate of the compressed image is lower then a certain threshold, the bumps start to form larger distorted patches. The metric therefore detects that the amount of discontinuities lowers, even though the image might no longer be recognizable. Otherwise, a similar behaviour to the PSNR



a) Cotton

b) Sideboard

Figure 11.6: Bumpiness performance on planar and non-planar surfaces of compression techniques for the EPINET conversion algorithm, displayed for the Cotton and Sideboard scenes.

metric can be seen for both images on both planar and non-planar (slanted) surfaces.

The Bjontengaard metric was again used to measure the differences between R-D curves, as can be seen in Tables 11.3 and 11.4. In the former case, there are again some noticeable trends, as video codecs outperform image compression techniques. However, no method can be named truly superior in either group, as the distortion is highly input dependent.

The Pyramids scene deviates in Bumpiness in the latter case. This is mainly due to the sheer amount of slanted continuous surfaces compared to the two other scenes. The slope of the surfaces is also different, which gives the potential to produce more discontinuities. Lastly, there is noise in each view, which is another difficulty the conversion algorithms have to deal with.

Table 11.3: Comparison of Bumpiness on planar surfaces by the Bjøntegaard metric (BD-Bumpiness), which computes the average distance between two R-D curves, for three different scenes and conversion algorithms. Lower values are better.

		BD-Bumpiness [-]																	
		EPINET						SPO						CAE					
		JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265
BOXES	JPG	X	-0.90	-1.25	0.19	-1.04	0.19	X	-0.19	-0.43	0.30	0.30	0.29	X	-0.07	0.05	0.01	-0.04	0.02
	JP2	0.90	X	0.78	1.08	-0.10	1.16	0.19	X	0.12	0.43	0.44	0.45	0.07	X	0.02	0.06	0.01	0.07
	BPG	1.25	-0.78	X	1.72	0.42	1.82	0.43	-0.12	X	1.33	1.30	1.36	-0.05	-0.02	X	0.12	0.09	0.13
	VP9	-0.19	-1.08	-1.72	X	-1.10	0.10	-0.30	-0.43	-1.33	X	-0.02	0.03	-0.01	-0.06	-0.12	X	-0.02	0.01
	x264	1.04	0.10	-0.42	1.10	X	1.06	-0.30	-0.44	-1.30	0.02	X	0.08	0.04	-0.01	-0.09	0.02	X	0.03
	x265	-0.19	-1.16	-1.82	-0.10	-1.06	X	-0.29	-0.45	-1.36	-0.03	-0.08	X	-0.02	-0.07	-0.13	-0.01	-0.03	X
SIDEBOARD	JPG	X	-0.13	-0.08	0.46	0.52	0.51	X	0.01	-0.14	0.19	0.19	0.21	X	0.02	-0.01	X	X	X
	JP2	0.13	X	-0.03	0.52	0.60	0.59	-0.01	X	-0.16	0.15	0.16	0.18	-0.02	X	-0.03	X	X	X
	BPG	0.08	0.03	X	0.99	0.99	0.99	0.14	0.16	X	0.53	0.50	0.53	0.01	0.03	X	X	X	X
	VP9	-0.46	-0.52	-0.99	X	-0.02	0.00	-0.19	-0.15	-0.53	X	-0.03	0.00	X	X	X	X	X	X
	x264	-0.52	-0.60	-0.99	0.02	X	0.07	-0.19	-0.16	-0.50	0.03	X	0.05	X	X	X	X	X	X
	x265	-0.51	-0.59	-0.99	0.00	-0.07	X	-0.21	-0.18	-0.53	0.00	-0.05	X	X	X	X	X	X	X
PYRAMIDS	JPG	X	-0.25	0.12	0.24	0.27	0.27	X	-0.10	0.07	0.22	0.14	0.11	X	-0.05	-0.03	0.03	0.04	0.03
	JP2	0.25	X	0.66	0.40	0.50	0.52	0.10	X	0.18	0.23	0.15	0.09	0.05	X	0.06	0.08	0.08	0.08
	BPG	-0.12	-0.66	X	1.21	1.21	1.24	-0.07	-0.18	X	0.99	0.94	0.98	0.03	-0.06	X	0.37	0.37	0.38
	VP9	-0.24	-0.40	-1.21	X	-0.03	-0.01	-0.22	-0.23	-0.99	X	-0.10	-0.07	-0.03	-0.08	-0.37	X	-0.01	0.00
	x264	-0.27	-0.50	-1.21	0.03	X	0.04	-0.14	-0.15	-0.94	0.10	X	-0.01	-0.04	-0.08	-0.37	0.01	X	0.01
	x265	-0.27	-0.52	-1.24	0.01	-0.04	X	-0.11	-0.09	-0.98	0.07	0.01	X	-0.03	-0.08	-0.38	0.00	-0.01	X

Table 11.4: Comparison of Bumpiness on slanted surfaces by the Bjøntegaard metric (BD-Bumpiness slanted), which computes the average distance between two R-D curves, for three different scenes and conversion algorithms. Lower values are better.

		BD-Bumpiness slanted [-]																	
		EPINET						SPO						CAE					
		JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265
COTTON	JPG	X	-0.03	0.21	0.22	0.31	0.41	X	-0.01	0.28	0.08	0.11	0.15	X	X	X	X	X	X
	JP2	0.03	X	0.29	0.22	0.27	0.26	0.01	X	0.05	0.04	0.05	0.05	X	X	X	X	X	X
	BPG	-0.21	-0.29	X	1.71	1.68	1.72	-0.28	-0.05	X	0.75	0.75	0.76	X	X	X	X	X	X
	VP9	-0.22	-0.22	-1.71	X	-0.06	-0.01	-0.08	-0.04	-0.75	X	-0.01	0.00	X	X	X	X	X	X
	x264	-0.31	-0.27	-1.68	0.06	X	0.06	-0.11	-0.05	-0.75	0.01	X	0.02	X	X	X	X	X	X
	x265	-0.41	-0.26	-1.72	0.01	-0.06	X	-0.15	-0.05	-0.76	0.00	-0.02	X	X	X	X	X	X	X
SIDEBOARD	JPG	X	-0.08	-0.18	0.61	0.65	0.67	X	0.03	0.11	0.16	0.16	0.17	X	0.05	0.02	X	X	X
	JP2	0.08	X	-0.18	0.63	0.70	0.71	-0.03	X	-0.02	0.10	0.11	0.10	-0.05	X	-0.06	X	X	X
	BPG	0.18	0.18	X	1.56	1.58	1.51	-0.11	0.02	X	0.26	0.27	0.27	-0.02	0.06	X	X	X	X
	VP9	-0.61	-0.63	-1.56	X	0.00	-0.05	-0.16	-0.10	-0.26	X	0.00	0.01	X	X	X	X	X	X
	x264	-0.65	-0.70	-1.58	0.00	X	0.00	-0.16	-0.11	-0.27	0.00	X	0.01	X	X	X	X	X	X
	x265	-0.67	-0.71	-1.51	0.05	0.00	X	-0.17	-0.10	-0.27	-0.01	-0.01	X	X	X	X	X	X	X
PYRAMIDS	JPG	X	-33.24	10.04	73.43	89.41	74.73	X	-33.83	-75.25	-30.19	-25.91	-31.28	X	108.34	-24.33	94.28	77.79	101.93
	JP2	33.24	X	24.39	106.38	124.12	107.04	33.83	X	-11.65	18.74	30.56	21.04	-108.34	X	-107.55	-1.94	-14.61	13.78
	BPG	-10.04	-24.39	X	43.13	32.62	23.63	75.25	11.65	X	36.09	36.37	59.38	24.33	107.55	X	34.87	8.40	13.62
	VP9	-73.43	-106.38	-43.13	X	-4.50	-14.10	30.19	-18.74	-36.09	X	1.04	20.35	-94.28	1.94	-34.87	X	-20.06	-14.11
	x264	-89.41	-124.12	-32.62	4.50	X	-1.96	25.91	-30.56	-36.37	-1.04	X	7.78	-77.79	14.61	-8.40	20.06	X	-3.22
	x265	-74.73	-107.04	-23.63	14.10	1.96	X	31.28	-21.04	-59.38	-20.35	-7.78	X	-101.93	-13.78	-13.62	14.11	3.22	X

Both the EPINET and SPO algorithms behave differently, as was shown in Figures 11.3 and 11.4. This is further confirmed by the Bjøntegaard metric, with SPO being the one developing fewer bumps than its counterpart. All compression techniques exhibit similar behaviour to a certain bit rate. Video codecs gain an advantage after it thanks to temporal coding.

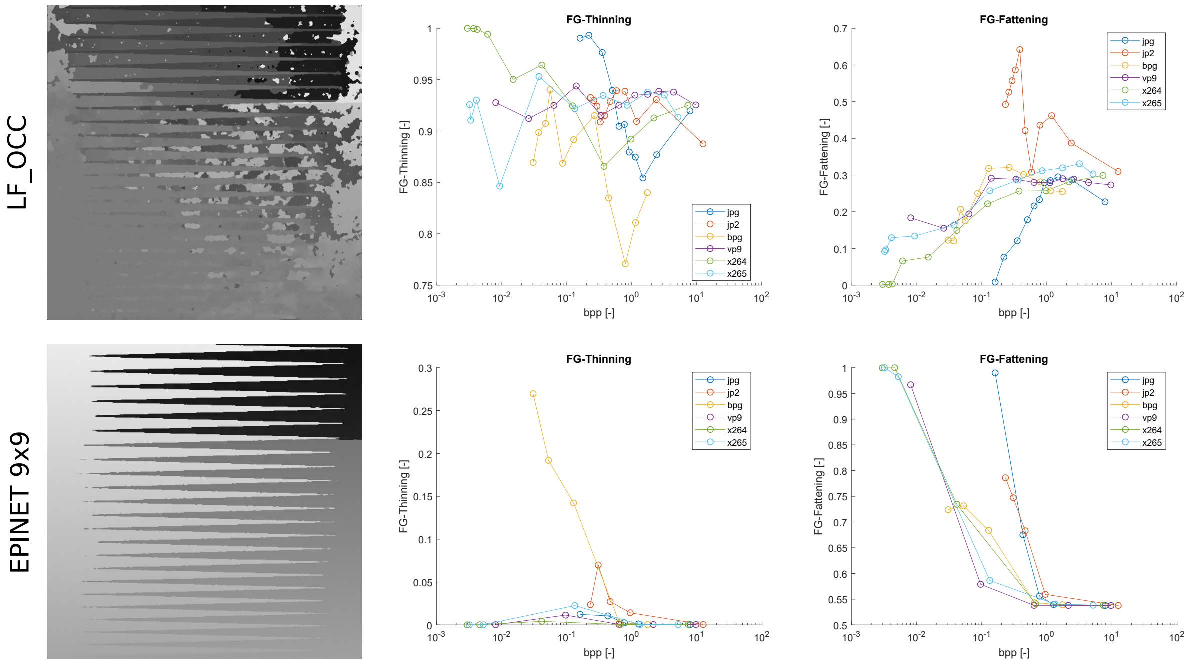


Figure 11.7: Foreground Thinning and Fattening performance of compression techniques for the LF_OCC and EPINET conversion algorithms. Displayed for the Backgammon scene, with examples (left column).

11.5.3 Foreground Thinning and Fattening metrics

In Figure 11.7 some results from the FG_Thinning and FG_Fattening measurement are displayed. The goal of these metrics is to quantify the amount of incorrectly assigned disparity values at occlusion boundaries for thin structures [31]. In the case of the EPINET algorithms, not much FG_Thinning occurred when compression was applied. The exception being BPG, which suffered greatly from impulse distortion.

What can, however, be seen from the figure, is that compression affects FG_Fattening. This can be credited to the occurrence of other types of artifacts, mainly blockiness, blur and inaccurate colour reproduction. Both JPEG standards show similar performance in that regard, even though they use different transform coding techniques. In the case of the legacy JPEG standard, that is probably mainly due to the 8×8 block size. Blur, on the other hand, is probably the cause when JPEG 2000 is applied. In contrast, the behaviour of BPG is comparable to AVC, at least in terms of the R-D curves. When looked at the distortion, differences can be seen.

VP9 outperforms the rest, as it is able to reach the lowest bit rate with the slits being still recognizable. Otherwise, the behaviour between VP9 and HEVC is very similar. The LF_OCC algorithm and its incorrect behaviour are also shown.

11.5.4 Thinning and Fattening metrics

The difference between the Thinning and Fattening metrics and their Foreground variants is in that for the former a threshold is first set, then the difference between the ground truth and the final disparity map is computed. Only values that exceed the threshold, which in our case was ± 0.15, are measured. This way, if a pixel is only slightly in the foreground or background, it is not counted as completely incorrect.

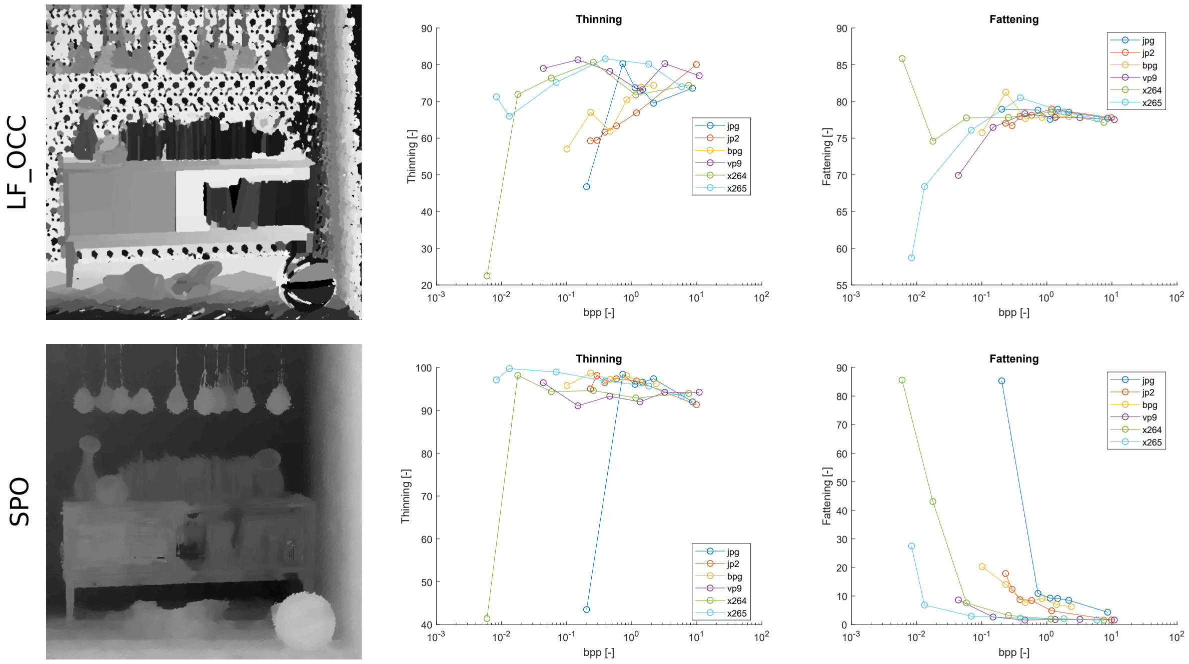
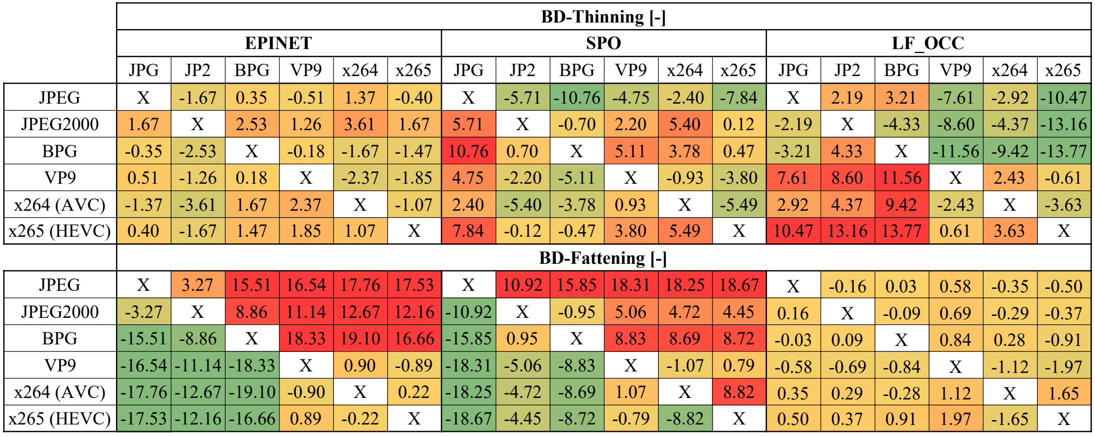


Figure 11.8: Thinning and Fattening performance of compression techniques for the LF_OCC and SPO conversion algorithms. Displayed for the Sideboard scene, with examples (left column).

Figure 11.8 shows the results for the SPO and LF_OCC algorithms. As can be observed, the former one displays high values of Thinning, no matter the compression. Fattening, on the contrary, is low. The two metrics present almost the opposite behaviour. However, in the case of Fattening, the compression algorithms can once again be split into two groups, with and without temporal coding. Clear conclusions cannot be drawn for the LF_OCC algorithm, because of its incorrect functioning, even though certain trends are noticeable.

The Bjøntegaard metric offers a clearer comparison, as can be seen in Table 11.5. In the case of Thinning, the differences between R-D curves for both EPINET and SPO are highly inconsistent. VP9 is, however, the most consistent performer for the latter. AVC shows very similar behaviour to VP9 for both algorithms.

Table 11.5: Comparison of Thinning and Fattening by the Bjøntegaard metric (BD-Thinning and BD-Fattening), which computes the average distance between two R-D curves, for the Sideboard scene and three conversion algorithms. Lower values are better.



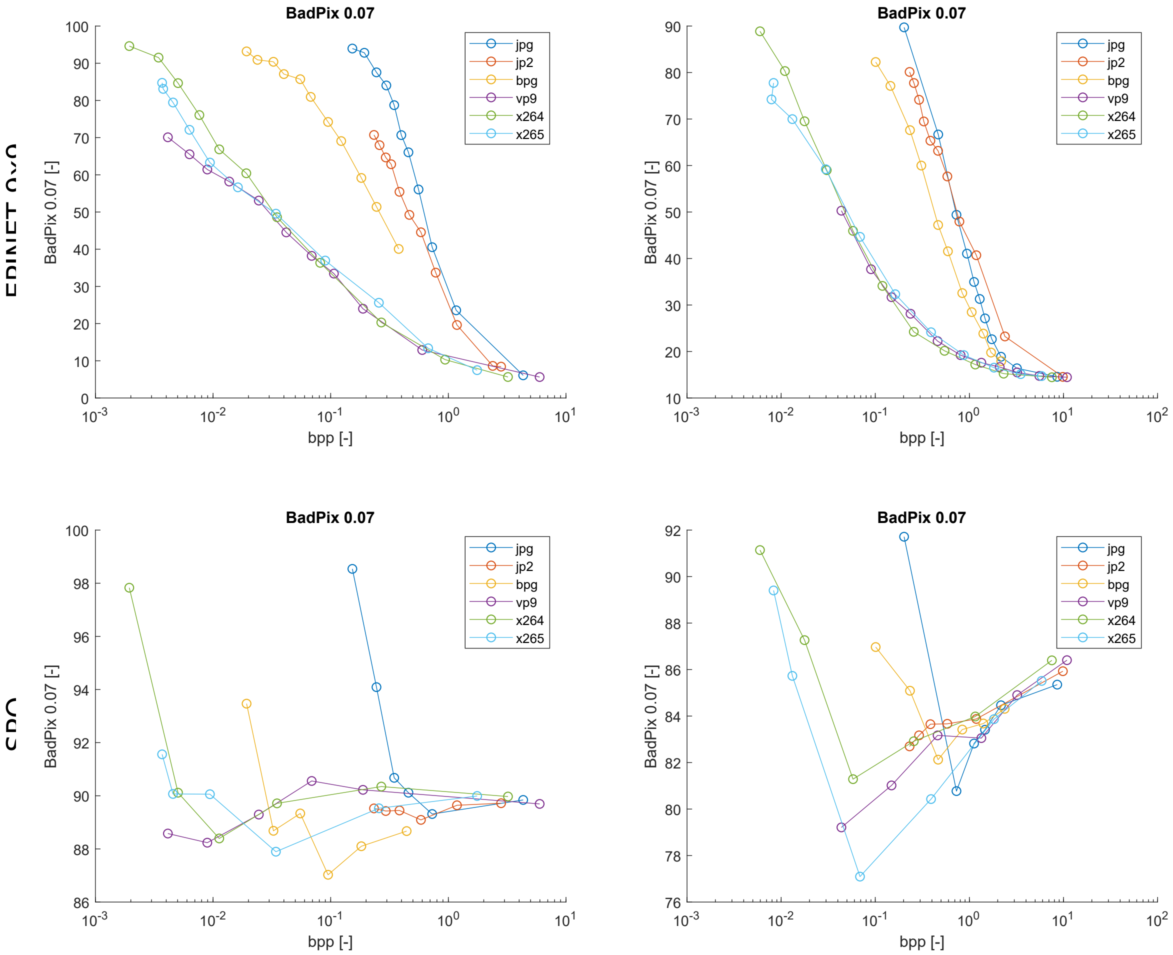
	BD-Thinning [-]																	
	EPINET						**SPO**						**LF_OCC**					
	JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265
JPEG	X	-1.67	0.35	-0.51	1.37	-0.40	X	-5.71	-10.76	-4.75	-2.40	-7.84	X	2.19	3.21	-7.61	-2.92	-10.47
JPEG2000	1.67	X	2.53	1.26	3.61	1.67	5.71	X	-0.70	2.20	5.40	0.12	-2.19	X	-4.33	-8.60	-4.37	-13.16
BPG	-0.35	-2.53	X	-0.18	-1.67	-1.47	10.76	0.70	X	5.11	3.78	0.47	-3.21	4.33	X	-11.56	-9.42	-13.77
VP9	0.51	-1.26	0.18	X	-2.37	-1.85	4.75	-2.20	-5.11	X	-0.93	-3.80	7.61	8.60	11.56	X	2.43	-0.61
x264 (AVC)	-1.37	-3.61	1.67	2.37	X	-1.07	2.40	-5.40	-3.78	0.93	X	-5.49	2.92	4.37	9.42	-2.43	X	-3.63
x265 (HEVC)	0.40	-1.67	1.47	1.85	1.07	X	7.84	-0.12	-0.47	3.80	5.49	X	10.47	13.16	13.77	0.61	3.63	X
	BD-Fattening [-]																	
JPEG	X	3.27	15.51	16.54	17.76	17.53	X	10.92	15.85	18.31	18.25	18.67	X	-0.16	0.03	0.58	-0.35	-0.50
JPEG2000	-3.27	X	8.86	11.14	12.67	12.16	-10.92	X	-0.95	5.06	4.72	4.45	0.16	X	-0.09	0.69	-0.29	-0.37
BPG	-15.51	-8.86	X	18.33	19.10	16.66	-15.85	0.95	X	8.83	8.69	8.72	-0.03	0.09	X	0.84	0.28	-0.91
VP9	-16.54	-11.14	-18.33	X	0.90	-0.89	-18.31	-5.06	-8.83	X	-1.07	0.79	-0.58	-0.69	-0.84	X	-1.12	-1.97
x264 (AVC)	-17.76	-12.67	-19.10	-0.90	X	0.22	-18.25	-4.72	-8.69	1.07	X	8.82	0.35	0.29	-0.28	1.12	X	1.65
x265 (HEVC)	-17.53	-12.16	-16.66	0.89	-0.22	X	-18.67	-4.45	-8.72	-0.79	-8.82	X	0.50	0.37	0.91	1.97	-1.65	X

More profound differences can be seen in the case of Fattening, where video codecs show similar behaviour and outperform the rest. HEVC outperforms the other techniques when the SPO algorithm is used. The same conclusion, however, cannot be drawn for the EPINET algorithm, as the performance of each codec varies based on bit rate to a greater extent. In case of image compression techniques, JPEG 2000 end up being the top performer for the SPO algorithm, BPG being the second best and legacy JPEG being last. For EPINET, the first two places are swapped, as BPG tops JPEG 2000.

11.5.5 BadPix metrics

Similar to the Fattening and Thinning metrics is the BadPix metric, which is computed at occlusion areas. A threshold is set, 0.07 px in our case, to which the absolute difference between the ground truth and the final disparity map is compared. The performance of the EPINET and SPO algorithms for two photorealistic scenes can be seen in Figure 11.9. In the case of the EPINET algorithm, some clear trends can be seen for both image and video compression techniques. There is clear performance separation between them. In case of image compression methods, BPG outperforms the rest for both scenes, with the JPEG standards being much closer to each other. The difference between video codec changes based on the scene. The best performer thus cannot be named, as is further displayed in Table 11.6.

The SPO algorithm varies much more in terms of the BadPix measurement then EPINET. Furthermore, there can be seen an improvement in the BadPix value in case of the Sideboard scene. This is due to fuzziness at occlusion areas, which is partly eliminated



a) Cotton

b) Sideboard

Figure 11.9: BadPix 0.07 px performance of compression techniques for the EPINET and SPO conversion algorithms. Displayed for the Cotton and Sideboard scenes.

with compression. However, upon detailed inspection, one can notice, that the decrease in value is relatively small.

The dependence on the scene is highlighted in Table 11.6. There, some clear patterns can be seen for the EPINET algorithm, but not for SPO.

11.5.6 AV1 performance

Lastly, the performance of the experimental implementation of AV1 was tested with the FFmpeg tool. This was done only for the Boxes scene and only for the EPINET conversion algorithm. When looked at both Figure 11.10 and Table 11.7, it can be seen that the AV1 codec outperforms the rest in most calculated metrics. The Bumpiness metric being the outlier. There both VP9 and HEVC perform better.

Table 11.6: Comparison of BadPix 0.07px by the Bjøntegaard metric (BD-BadPix), which computes the average distance between two R-D curves, for three scenes and conversion algorithms. Lower values are better.

		BD-BadPix 0.07 [-]											
		EPINET						SPO					
		JPG	JP2	BPG	VP9	x264	x265	JPG	JP2	BPG	VP9	x264	x265
COTTON	JPG	X	6.06	35.11	29.94	33.01	38.33	X	0.62	5.15	0.16	-0.38	1.51
	JP2	-6.06	X	17.82	23.11	23.08	23.84	-0.62	X	0.82	-1.32	-2.04	-0.38
	BPG	-35.11	-17.82	X	37.54	36.51	36.02	-5.15	-0.82	X	-1.41	-0.82	-0.11
	VP9	-29.94	-23.11	-37.54	X	-2.99	-2.83	-0.16	1.32	1.41	X	-0.14	0.68
	x264	-33.01	-23.08	-36.51	2.99	X	0.40	0.38	2.04	0.82	0.14	X	0.83
	x265	-38.33	-23.84	-36.02	2.83	-0.40	X	-1.51	0.38	0.11	-0.68	-0.83	X
SIDEBOARD	JPG	X	-2.25	12.57	19.48	21.67	21.88	X	0.52	0.18	0.95	0.45	1.93
	JP2	2.25	X	12.93	19.44	22.55	22.78	-0.52	X	0.20	0.12	-0.37	1.08
	BPG	-12.57	-12.93	X	25.22	26.15	24.22	-0.18	-0.20	X	1.43	0.90	3.31
	VP9	-19.48	-19.44	-25.22	X	0.62	-0.95	-0.95	-0.12	-1.43	X	-0.95	1.50
	x264	-21.67	-22.55	-26.15	-0.62	X	-0.35	-0.45	0.37	-0.90	0.95	X	2.59
	x265	-21.88	-22.78	-24.22	0.95	0.35	X	-1.93	-1.08	-3.31	-1.50	-2.59	X
PYRAMIDS	JPG	X	-5.10	15.19	17.66	20.18	18.99	X	14.61	0.42	-0.13	-3.63	0.74
	JP2	5.10	X	22.85	13.67	17.15	16.83	-14.61	X	-8.56	-15.01	-17.80	-10.93
	BPG	-15.19	-22.85	X	39.60	39.34	42.79	-0.42	8.56	X	-4.74	-1.03	-6.05
	VP9	-17.66	-13.67	-39.60	X	-1.20	1.24	0.13	15.01	4.74	X	2.75	-1.02
	x264	-20.18	-17.15	-39.34	1.20	X	12.90	3.63	17.80	1.03	-2.75	X	-5.39
	x265	-18.99	-16.83	-42.79	-1.24	-12.90	X	-0.74	10.93	6.05	1.02	5.39	X

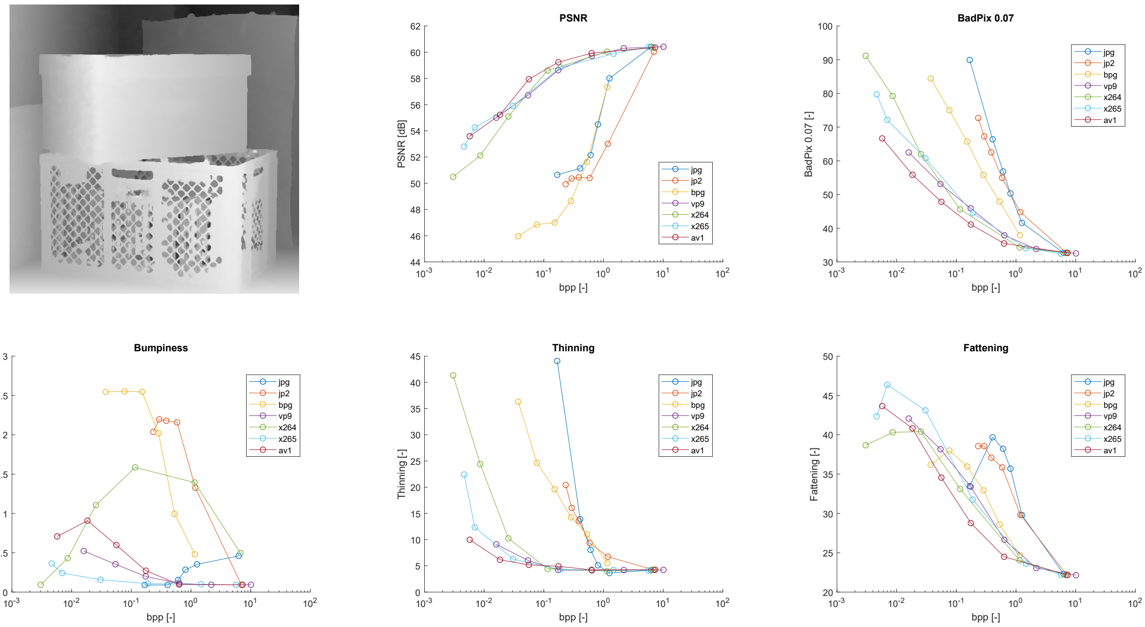
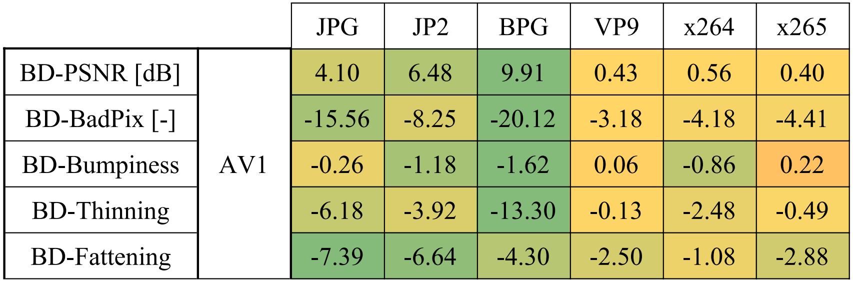


Figure 11.10: Performance of the experimental AV1 compression algorithm in comparison to the other tested compression methods. Computed by EPINET for the Boxes scene.

11.5.7 Computational demands

The evaluation was implemented in Matlab version 2017b and run on a computer equipped with a Ryzen 5 1600 CPU running at 3.70 GHz, an NVIDIA GTX 1050Ti and 16 GB of RAM. The run times of the tested algorithms can be seen in Table 11.8. It has to be noted that the output of the EPINET algorithm is a PFM file, which has to be converted to a

Table 11.7: Performance of the experimental AV1 compression algorithm compared to the other tested compression methods with the Bjøntegaard metric, which computes the average distance between two R-D curves,. Computed by EPINET for the Boxes scene. Lower values are better, except for BD-PSNR.



		JPG	JP2	BPG	VP9	x264	x265
BD-PSNR [dB]	AV1	4.10	6.48	9.91	0.43	0.56	0.40
BD-BadPix [-]		-15.56	-8.25	-20.12	-3.18	-4.18	-4.41
BD-Bumpiness		-0.26	-1.18	-1.62	0.06	-0.86	0.22
BD-Thinning		-6.18	-3.92	-13.30	-0.13	-2.48	-0.49
BD-Fattening		-7.39	-6.64	-4.30	-2.50	-1.08	-2.88

PNG before it can be loaded to Matlab. The run time thus includes both the estimation and the conversion to PNG. Still it took far less time than the other algorithms.

Table 11.8: Average run times for the disparity map estimation for the Boxes scene.

Method	LF_OCC	CAE	SPO	EPINET 9×9
Time	446s	1742s	1058s	43s

In terms of compression, the run times were mostly seconds long. The only exception was AV1, which took anywhere from seconds to about a minute. This is because FFmpeg version 4.1.1. has AV1 only as an experimental implementation.

11.6 Summary

Several compression algorithms were tested in combination with several light field to disparity estimation methods on eight scenes. The tested compression algorithms exhibit different cases of distortion, which has, in turn, different effects on the disparity estimation (see Figures 11.3 and 11.4). The effects are not however consistent for all the conversion methods. They are rather highly scene and method dependent.

Several metrics were computed (see 11.4), and their values were evaluated. The results imply that the inclusion or absence of temporal coding makes a significant difference in performance for most metrics. The methods can thus be divided into two groups. Further testing would, however, need to be done, before a definitive best performer for each group could be named.

The experimental implementation of the AV1 codec outperformed the rest for the Boxes scene but was not tested for other scenes. VP9 and HEVC (x265) perform better than the older AVC (x264) on several occasion, however, as the tables and figures in this chapter show, there are many scenes and metrics where the differences between them are negligible. On some occasions, the AVC encoder performs better.

In terms of image-based compression techniques, the results are even more scene and metric dependent, as can be seen for example in Figure 11.5 and Table 11.2. The absence of temporal coding, however, puts these methods behind video compression methods in terms of performance.

Chapter 12

Conclusions and future work

The project introduces three imaging modalities, light field, point cloud and holography, and explores currently available techniques of conversion between them. At first, an introduction to plenoptic imaging is made with the basis being the plenoptic function. The different representations of the function are then shown in Chapters 2 to 5. Each of those can be split into three main steps, depicting gradually the main principle, data acquisition and data representation. One should be able to get a clear distinction between the modalities in terms of their usage, benefits and drawbacks.

Chapters 6 to 8 summarize the different techniques of conversion between the modalities[1]. The character and amount of data between the representations can differ greatly. It is thus not always possible or easy to convert between them. Assumptions have to be often made about the information at hand for the procedure to work. Even so, there are still directions for which the conversions are either not described or an intermediary step through a different representation has to be made. This is mainly the case for the conversion from point cloud, as it usually provides fewer data and thus a worse representation of the specific scene then both light field and holography. With the introduction of JPEG Pleno and the hardware and software available to capture and process the data, an increase in research can be seen nowadays.

Chapter 9 focuses on compression of the three modalities. Many different methods can be applied; however, certain algorithms appear for each. This is the case for example for JPEG, JPEG 2000, AVC and HEVC. Wavelets show promising results for compression of raw light fields. However, as pseudo-sequence methods show, efficiently exploring the correlation between views of the light field image seems to be the key. From the three modalities, most research by far was done on point cloud compression, as there are many

[1] Further details are included in Appendix B or its electronic version at `https://docs.google.com/spreadsheets/d/1HgcG6RxWGCKbkmWplLjZq0qN3FWMQe6dCPxWFNcP6U8/edit?usp=sharing`

proposed techniques. This is partly due to research in autonomous driving. Certain methods can be used for both geometry and colour compression, as is the case, for example, for octrees. Others utilize conventional compression algorithms as part of their compression scheme. This is the case for example for JPEG, JPEG 2000, AVC and HEVC. The same encoders can be used in holography as well. Holographic compression is, however, in comparison to the other two modalities still in fairly early stages of research. The information stored in digital holograms is completely scrambled, which makes it problematic to compress. Wavelet-based approaches, together with the content-aware techniques, seem to show most promise so far. A better understanding of the hologram structure and its features has to be made before truly efficient compression schemes can be designed.

Chapter 10 is dedicated to the datasets and software that can be used for testing[2]. Two ways of dataset creation are depicted based on the method of acquisition. The benefits of both real and synthetic scenes are also mentioned so are the various types of software for both displaying and processing. For the development of conversion algorithms, the choice of datasets can be crucial. This is true primarily in the case of neural networks as they have to be properly trained.

In Chapter 11, the effects of compression of light field images on disparity estimation are analysed. The results show that the experimental implementation of the AV1 codec outperformed the other tested codec for one specific scene. However, as it was not tested on other, it cannot be conclusively named the best performer overall. Video encoders are shown to have better performance than still image coding approaches, as they exploit the correlation between the views of light field images. VP9 and HEVC (x265) perform better than the older AVC (x264) on several occasion. Their behaviour is, however, highly scene dependent, and thus inconsistent. This can be seen on some occasions, where the AVC encoder performs better than both VP9 and HEVC. In terms of image-based compression techniques, the results are even more scene and metric dependent, thus more inconsistent. However, the absence of temporal coding puts these methods behind video compression methods in terms of performance. Other compression algorithms, together with other conversion methods, would need to be tested before a definitive conclusion could be drawn.

The computational together with memory or storage demands for the conversions are often quite large. A closer look at the effects of compression algorithms at the different conversions could be made in future work. Subjective evaluation metrics could also be

[2] A table of publicly available datasets can be found in Appendix A and its electronic version at https://docs.google.com/spreadsheets/d/1OzMUTy9Ll_5M59dbuyTsNs6mkmzXOJdyDXAfBWyCWFM/edit?usp=sharing

added as precision is not always the most desired aspect. That is the case, for example, in portrait photography in smartphones. It could also be beneficial to explore what could be done about the missing information in the different representations or what effects does the conversion between two modalities have on the data. Different quality assessment techniques could be tested. Lastly, a comparison to JPEG Pleno could be made after the standard is finalized.

Appendix A

Datasets

Table A.1: Publicly available holographic datasets.

name	origin/hosted by	modality	consists of	size overall	size of the main data	size of additional data	filetype main data	filetype additional data	scene type	acquisition
ERC Interfere Holographicdata set	Vrije Universiteit Brussel	holography	holograms	1,07 GB	1,07 GB	x	mat;	x	synthetic	CGH - reconstructed from point cloud data
EmergImg-HoloGrail	Universidade da Beira Interior	holography	holograms	999 MB	999 MB	x	mat;	x	synthetic; real	CGH and acquired with an optical recording setup
Holographic database	Advanced Media Coding Lab at IRT b-com	holography	holograms	~276,34 GB	~276,34 GB	x	bmp; exr;	x	synthetic; real	computation from a synthetic scene or Multiview-plus-Depth projections

Table A.2: Publicly available point cloud datasets.

name	origin/hosted by	modality	consists of	size overall	size of the main data	size of additional data	filetype main data	filetype additional data	scene type	acquisition
UNIVERSITY OF SÃO PAULO POINT CLOUD DATASET	University of São Paulo	point cloud	point clouds	1,57 GB	1,57 GB	x	ply;	x	real	?
GTI-UPM Point-cloud dataset	Universidad Politécnica de Madrid	point cloud	point clouds	192 MB	192 MB	x	obj; jpg; wrl;	x	real	?
8iVFB v2	8i Labs	point cloud	point clouds	5,5 GB	5,5 GB	x	ply; png; mp4; txt; pdf; docx;	x	real	RGB cameras
Microsoft Voxelized UpperBodies - A Voxelized Point Cloud Dataset	Microsoft	point cloud	point clouds	7,9 GB	7,9 GB	x	ply; png; avi; pdf;	x	real	RGBD cameras

Table A.3: Publicly available light field datasets.

name	**origin/hosted by**	**modality**	**consists of**	**size overall**	**size of the main data**	**size of additional data**	**filetype main data**	**filetype additional data**	**scene type**	**acquisition**
(OLD) 4D Light Field Benchmark	University of Konstanz and the HCI at Heidelberg University	light field	light field images; depth maps	4,02 GB	4,02 GB	x	png; h5; gif	x	synthetic; real	blender; gantry; Raytrix
(NEW) 4D Light Field Benchmark	University of Konstanz and the HCI at Heidelberg University	light field	light field images; depth maps	5,68 GB	3,16 GB	1,52 GB	png; pfm; cfg	pfm; cfg	synthetic	blender
DDF12	Caner Hazirbas - TU Munich	light field	light field images; depth maps	93,76 GB	80,4 GB	13,36 GB	npy; mat; png; raw; h5; ?	png; lfr; json; gct; hst; raw; txt;	real	Lytro ILLUM
Lytro Dataset	Irisa	light field	light field images;	1,33 GB	1,33 GB	x	lfp; gct; jpg	x	real	Lytro F01
Material recognition dataset	University of California San Diego	light field	light field images; labels	30 GB	30GB	x	png;	x	real	Lytro ILLUM
Synthetic Light Field Archive	Massachusetts Institute of Technology	light field	light field images	348 MB	349 MB	x	png;	x	synthetic	POV-Ray
Occlusion-aware depth estimation using light-field cameras	University of California, Berkley	light field	light field images	3,03 GB	3,03 GB	x	h5;	x	real	Lytro ILLUM
Scene Reconstruction from High Spatio-Angular Resolution Light Fields	Disney research	light field	light field images; depth maps	7,18 GB	7,18 GB	x	png; jpg; dmap;	x	real	Multi-Camera array
Stanford Lytro Light Field Archive	Stanford University	light field	light field images; depth maps	60,7 GB	60,7 GB	x	png; json; jsn;	x	real	Lytro ILLUM
Multiview Dataset	Stanford University	light field	light field images; depth maps	212,1 GB / 223,1 GB	207 GB / 218 GB	5,1 GB	lfr; decoded eslf;	?	real	Lytro
Three-View Dataset	Stanford University	light field	?	155 GB	150 GB	5,1 GB	lfr; decoded eslf;	?	real	Lytro ILLUM
The (New) Stanford Light Field Archive	Stanford University	light field	light field images;	18,95	18,95	x	png; jpg; hom;	x	real	Stanford Multi-Camera Array; gantry; Lego Mindstorms gantry; light field microscope
The (Old) Stanford Light Field Archive	Stanford University	light field	light field images;	140,3 MB	140,3 MB	x	lif; rgb; lid;	?	synthetic; real	gantry
Light-Field Image Dataset	École polytechnique fédérale de Lausanne	light field	light field images; depth maps	56,6 GB	55 GB	1,6 GB	lfr; mat; jpg;	?	real	Lytro
EPFL Light-field lensletdata set	École polytechnique fédérale de Lausanne	light field	light field images;	3,42 GB	3,42 GB	x	ppm;	x	real	?
LCAV-31	École polytechnique fédérale de Lausanne	light field	light field images;	660 MB	660 MB	x	jpg;	x	real	Lytro
Light Field Dataset	Universidad de Zaragoza	light field	light field images; depth maps	384 MB	384 MB	x	jpg; float;	x	synthetic; real	Raytrix; gantry; Lytro; PBRT (the physically-based renderer)
Depth from Combining Defocus and Correspondence Using light-Field Cameras dataset	University of California, Berkley	light field	light field images;	80 MB	80 MB	x	jpg; lfp;	x	real	Lytro

Appendix B

Conversion methods

Table B.1: Methods, which use neural networks for the conversion between light field and point cloud/depth map.

paper	**name in the 4D benchmark**	**year**	**method**	**code publicly available**	**programing language used**	**hardware used to run the code**	**training dataset(s)**	**number of scenes**
Convolutional networks for shape from light field	-	2016	CNN (neural networks)	No	-	-	LF dataset generated by POV-Ray	8,00E+06
EPI-Patch Based Convolutional Neural Network for Depth Estimation on 4D Light Field	EPN+OS+GC	2017	CNN (neural networks)	No	Python + Matlab	i7-4720HQ @ 2.60GHz, 128 GB RAM, TITAN X	4D light field benchmark dataset	16
EPI-Patch Based Convolutional Neural Network for Depth Estimation on 4D Light Field	SOA-EPN	2017	CNN (neural networks)	No	Python + Matlab	i7-4720HQ @ 2.60GHz, 128 GB RAM, TITAN X	4D light field benchmark dataset	33
Neural epi-volume networks for shape from light field	-	2017	CNN (neural networks)	No	-	-	LF dataset generated by POV-Ray	900
EPINET: A Fully-Convolutional Neural Network Using Epipolar Geometry for Depth from Light Field Images	Epinet-fcn	2018	CNN (neural networks)	Yes	Python	i7-7770 @ 3.6GHz, 32GB RAM, 1080ti	LF dataset generated by POV-Ray (a combination of those used in Convolutional networks for shape from light field and Neural epi-volume networks for shape from ligh tfield	250
EPINET: A Fully-Convolutional Neural Network Using Epipolar Geometry for Depth from Light Field Images	Epinet-fcn-m	2018	CNN (neural networks)	Yes	Python	i7-7770 @ 3.6GHz, 32GB RAM, 1080ti	LF dataset generated by POV-Ray (a combination of those used in Convolutional networks for shape from light field and Neural epi-volume networks for shape from ligh tfield	250
EPINET: A Fully-Convolutional Neural Network Using Epipolar Geometry for Depth from Light Field Images	Epinet-fcn9X9	2018	CNN (neural networks)	Yes	Python	i7-7770 @ 3.6GHz, 32GB RAM, 1080ti	LF dataset generated by POV-Ray (a combination of those used in Convolutional networks for shape from light field and Neural epi-volume networks for shape from ligh tfield	250

Table B.2: Other methods for the conversion between light field and point cloud/depth map.

paper	**name in the 4D benchmark**	**year**	**method**	**occlusion aware**	**code publicly available**	**programing language used**	**hardware used to run the code**
Epipolarplane image analysis: An approach to determining structure from motion	-	1987	line fitting	-	No	-	-
Accurate correspondences from epipolar plane images	-	2001	line fitting (cost function)	-	No	-	-
Handling occlusions indense multi-view stereo	-	2001	shiftable window + multi-view (cost function, data cost)	Yes	No	-	-
Multi-camera scene reconstruction via graph cuts	-	2002	graph cut	-	Yes	C++	-
Stereo matching using belief propagation	-	2003	multi-view	-	No	-	-
Asymmetrical occlusion handling using graph cut for multi-view stereo	-	2005	graph cut (occlusion)	Yes	No	-	-
Reconstructing occluded surfaces using syntheticapertures: Stereo, focus and robust measures	-	2006	synthetic apertures (cost fuction, data cost)	Yes	No	-	-
Plenoptic depth estimation from multiple aliased views	-	2009	multi-view	-	No	-	-
Reducing plenoptic camera artifacts	-	2010	cross correlation	-	No	-	-
Surface stereo with softsegmentation	-	2010	stereo algorithm (surface based)	-	No	-	-
Globally Consistent Depth Labeling of 4D Light Fields	EPI2	2012	structure tensor	-	No	-	-
Structure and motion from scene registration	-	2012	convertion into a volumetric space	-	No	-	-
The light field camera: Extended depth of field, aliasing, and superresolution	-	2012	multi-view	-	No	-	-
Depth from combining defocus and correspondence using lightfield cameras	-	2013	defocus and correspondence (data cost)	Yes	Yes	Matlab	-
Line assisted light field triangulation and stereo matching	-	2013	line fitting	-	No	-	-
Reconstructing reflective and transparent surfaces from epipolar plane images	-	2013	structure tensor	-	No	-	-
Scene reconstruction from high spatio-angular resolution light fields	-	2013	line fitting	-	Yes	Python	-
Variational shape from light field	-	2013	active wavefront sampling	-	No	-	-
Depth estimation for glossy surfaces with light-field cameras	-	2014	glossy	-	No	-	-
Light field scale-depth space transform for dense depth estimation	-	2014	convolution + Ray Gaussian kernel	-	No	-	-
Light field stereo matching using bilateral statistics of surface cameras	-	2014	mix of defocus and correspondence and multi-view	Yes	No	-	-
Shape from light field meets robust PCA	-	2014	multi-view - low rank structure regularization	-	No	-	-
Variational light field analysis for disparity estimation and super-resolution	-	2014	structure tensor	-	Yes	C++	-
A multi-resolution approach to depth field estimation in dense image arrays	RM3DE	2015	multi-resolution	-	No	-	-
Accurate Depth Map Estimation from a Lenslet Light Field Camera	LF	2015	cost volume, phase shift	-	Yes	Matlab	i7-4770 @3.40GHz
Depth from shading, defocus, and correspondence using light-field angular coherence	-	2015	defocus and correspondence (data cost)	-	Yes	Matlab	-
Depth recovery from light field using focal stack symmetry	-	2015	defocus and correspondence	-	No	-	-
Occlusion-aware depth estimation using light-field cameras	LF_OCC	2015	defocus and correspondence (data cost)	Yes	Yes	Matlab	i7-4770 @3.40GHz
Shape estimation from shading, defocus, and correspondence using light-field angular coherence	-	2015	defocus and correspondence	-	No	-	-
Variational separation of light field layers	-	2015	layer separation	-	No	-	-
Dense Depth-map Estimation and Geometry Inference from Light Fields via Global Optimization	SC_GC	2016	cost volume (optimization)	-	No	-	-
Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras	-	2016	glossy	-	No	-	-
Depth estimation with occlusion modeling using light-field cameras	-	2016	defocus and correspondence (occlusion)	Yes	No	-	-
Depth from gradients in dense light fields for object reconstruction	-	2016	patch-based local gradient	-	No	-	-
Occlusion-aware depth estimation using sparse light field coding	-	2016	sparse light field coding (occlusion)	Yes	No	-	-
3D point cloud reconstruction from single plenoptic image	-	2016	histogram stretching, edge detection, image fusion	-	No	-	-
Robust Depth Estimation for Light Field via Spinning Parallelogram Operator	SPO	2016	spinning parallelogram	Yes	Yes	Matlab	i7-4790 CPU @ 3.60GHz
Robust Light Field Depth Estimation for Noisy Scene with Occlusion	-	2016	defocus and correspondence (data cost)	Yes	Yes	Matlab	-
SVBRDF-invariant shape and reflectance estimation from light-field cameras	-	2016	glossy	-	No	-	-
What sparse light field coding reveals about scene structure	EPI1	2016	light field dictionary	-	No	-	-
Accurate depth and normal maps from occlusion-aware focal stack symmetry	OFSY330DNR	2017	defocus and correspondence	Yes	Yes	Matlab	i7-3770 @ 3.40GHz, Titan X
Depth estimation from light field by accumulating binary maps based on foreground-background separation.	FBS	2017	binary maps	-	No	-	-
Fast and Efficient Depth Map Estimation from Light Fields	BSL	2017	line fitting (matching cost)	-	No	-	-
	BSL_I						
Geometric Occlusion Analysis in Depth Estimation using Integral Guided Filter for Light-Field Image	-	2017	multi-view	Yes	No	-	-
Robust and dense depth estimation for light field images	-	2017	multi-view (stereo non-dense methods)	-	No	-	-
Robust Pseudo Random Fields for Light-Field Stereo Matching	RPRF-5view	2017	statistical method (Bayesian framework)	-	Yes	Matlab and C++	i7-6700 @ 3.40GHz
Depth from a Light Field Image with Learning-based Matching Costs	PS_RF	2018	cascade random forest (cost volume)	-	No	-	-
Occlusion-Aware Depth Estimation for Light Field Using Multi-Orientation EPIs	SPO-MO	2018	spinning parallelogram (optimization)	Yes	No	-	-
Robust Light Field Depth Estimation using Occlusion-Noise Aware Data Costs	CAE	2018	defocus and correspondence (data cost)	Yes	Yes	Matlab	i7-7770 @ 3.6GHz

Appendix C

Software

Table C.1: Publicly available software.

name	toolbox for	modality
Point cloud tools	Matlab	point cloud
PovRay	x	point cloud
Point Cloud Maker 11	x	point cloud
MeshLab	x	point cloud
CloudCompare	x	point cloud
Computer Vision Toolkit	x	point cloud
Blender	x	point cloud
OctoMap	x	point cloud
OpenCTM	x	point cloud
Corto	x	point cloud
Draco	x	point cloud
LASzip	x	point cloud
LEPCC	x	point cloud
PCL	x	point cloud
cwi-pcl-codec	x	point cloud
pointzip	x	point cloud
PDAL	x	point cloud
Pointfuse	x	point cloud
Light Field Toolbox	Matlab	light field
Light Field Imaging Toolkit (LFIT)	Matlab	light field
Plenoptic Toolbox 2.0	Python	light field
Plenoptic-Simulation	Blender	light field
cocolib	x	light field
Geometric Calibration for Light Field	Matlab	light field
Blender Addon	Blender	light field
Python tools	Python	light field
lfptools	x	light field
Light field WebGL viewer	x	light field
light-field-graph-codec	Matlab	light field
Lightfield-processing	x	light field
WaSP light field compression	x	light field
LYTRO meltdown	x	light field
python-lfp-reader	Python	light field
HoloPy	Python	holography
HoloRec3D	Matlab	holography
CGDH Tools	Matlab	holography
x265 HEVC Encoder	x	
JEM software	x	
Kakadu software	x	
ICME 2018 Grand Challenge	x	

Appendix D

Structure of appendix archive

The appendix archive comprises of 2 folders:

- **/Implementation/...** - Folder containing all the scripts necessary for the evaluation described in Chapter 11. The folder contains a README file, which further explains the structure of underlining sub-folders and their contents.
- **/Results/...** - Folder containing results in the form of tables, figures and computed disparity maps. The folder also contains a README file, which further explains the structure of the underlining sub-folders and their contents.

Bibliography

[1] U. Schnars and W. Jueptner, *Digital holography, Digital hologram recording, numerical reconstruction, and related techniques.* Berlin: Springer - Verlag, 2005, ISBN: 3-540-21934-x.

[2] 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.

[3] F. Pereira and E. A. B. da Silva, "Efficient plenoptic imaging representation, Why do we need it?", in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2016, pp. 1–6, ISBN: 978-1-4673-7258-9. DOI: `10.1109/ICME.2016.7552939`. [Online]. Available: `http://ieeexplore.ieee.org/document/7552939/`.

[4] 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. DOI: `10.1109/MMUL.2016.64`. [Online]. Available: `http://ieeexplore.ieee.org/document/7742781/`.

[5] J. Švihálek, *Metody zpracování plenoptických obrazových dat*, Magisterská práce, Praha, 2018. [Online]. Available: `https://dspace.cvut.cz/bitstream/handle/10467/73930/F3-DP-2018-Svihalek-Jan-DP_svihajan_plenoptic.pdf`.

[6] E. Adelson and J. Wang, "Single lens stereo with a plenoptic camera", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 99–106, 1992, ISSN: 0162-8828. DOI: `10.1109/34.121783`. [Online]. Available: `http://ieeexplore.ieee.org/document/121783/`.

[7] B. Goldlücke, O. Klehm, S. Wanner, and E. Eisemann, "Plenoptic cameras", in *Digital representations of the real world: How to capture, model, and render visual reality*, Boca Raton: CRC Press, 2015, pp. 65–77, ISBN: 978-1-4822-4381-9.

[8] R. Ng, *Digital light field photography*, PhD thesis, AAI3219345, Stanford, 2006. [Online]. Available: `https://stanford.edu/class/ee367/reading/Ren%20Ng-thesis%20Lytro.pdf`.

[9] M. Yamaguchi, "Full-parallax holographic light-field 3-D displays and interactive 3-D touch", *Proceedings of the IEEE*, vol. 105, no. 5, pp. 947–959, 2017, ISSN: 0018-9219. DOI: `10.1109/JPROC.2017.2648118`. [Online]. Available: `http://ieeexplore.ieee.org/document/7831483/`.

[10] G. Lippmann, "Épreuves réversibles donnant la sensation du relief", *Journal de Physique Théorique et Appliquée*, vol. 7, no. 1, pp. 821–825, 1908, ISSN: 0368-3893. DOI: `10.1051/jphystap:019080070082100`. [Online]. Available: `http://www.edpsciences.org/10.1051/jphystap:019080070082100`.

[11] H. Houshiar, *Documentation and mapping with 3D point cloud processing*, PhD thesis, Würzburg, 2017. [Online]. Available: https://opus.bibliothek.uni-wuerzburg.de/opus4-wuerzburg/frontdoor/deliver/index/docId/14449/file/Houshiar_Thesis_WFRT12.pdf.

[12] K. Sugimoto, R. A. Cohen, D. Tian, and A. Vetro, "Trends in efficient representation of 3d point clouds", in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2017, pp. 364–369, ISBN: 978-1-5386-1542-3. DOI: 10.1109/APSIPA.2017.8282059. [Online]. Available: http://ieeexplore.ieee.org/document/8282059/.

[13] M. Weinmann, "Preliminaries of 3D point cloud processing", in *Reconstruction and Analysis of 3D Scenes*, Cham: Springer International Publishing, 2016, pp. 17–38, ISBN: 978-3-319-29244-1. DOI: 10.1007/978-3-319-29246-5_2. [Online]. Available: http://link.springer.com/10.1007/978-3-319-29246-5_2.

[14] L. Linsen, *Point cloud representation.* Univ., Fak. für Informatik, Bibliothek Technical Report, Faculty of Computer Science, University of Karlsruhe, 2001.

[15] T.-C. Poon and J.-P. Liu, *Introduction to modern digital holography.* Cambridge: Cambridge University Press, 2014, ISBN: 978-1-107-01670-5.

[16] H. Halaq, N. Demoli, I. Sović, K. Šariri, M. Torzynski, and D. Vukičević, "Undersampled digital holographic interferometry", in *Proceedings of SPIE*, vol. 6995, 2008, pp. 699 504–1–699504–5. DOI: 10.1117/12.781365. [Online]. Available: https://doi.org/10.1117/12.781365.

[17] K.-J. Oh, H.-G. Choo, and J. Kim, "Analysis on digital holographic data representation and compression", in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, IEEE, 2016, pp. 1–4, ISBN: 978-9-8814-7682-1. DOI: 10.1109/APSIPA.2016.7820789. [Online]. Available: http://ieeexplore.ieee.org/document/7820789/.

[18] J. Peixeiro, C. Brites, J. Ascenso, and F. Pereira, "Digital holography, benchmarking coding standards and representation formats", in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2016, pp. 1–6, ISBN: 978-1-4673-7258-9. DOI: 10.1109/ICME.2016.7552940. [Online]. Available: http://ieeexplore.ieee.org/document/7552940/.

[19] C. Shin, H. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images", *CoRR*, vol. abs/1804.02379, 2018. arXiv: 1804.02379. [Online]. Available: http://arxiv.org/abs/1804.02379.

[20] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 41–48, ISBN: 978-1-4673-1228-8. DOI: 10.1109/CVPR.2012.6247656. [Online]. Available: http://ieeexplore.ieee.org/document/6247656/.

[21] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator", *Computer Vision and Image Understanding*, vol. 145, pp. 148–159, 2016, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2015.12.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314215002714.

[22] H. Sheng, P. Zhao, S. Zhang, J. Zhang, and D. Yang, “Occlusion-aware depth estimation for light field using multi-orientation EPIs”, *Pattern Recognition*, vol. 74, pp. 587–599, 2018, ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2017.09.010`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0031320317303539`.

[23] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross, “Scene reconstruction from high spatio-angular resolution light fields”, *ACM Transactions on Graphics*, vol. 32, no. 4, article no. 73, 2013. [Online]. Available: `http://people.csail.mit.edu/changil/assets/scene-reconstruction-from-high-spatio-angular-resolution-light-fields-siggraph-2013-compressed-kim-et-al.pdf`.

[24] Y. Anisimov and D. Stricker, “Fast and efficient depth map estimation from light fields”, in *2017 International Conference on 3D Vision (3DV)*, IEEE, 2017, pp. 337–346, ISBN: 978-1-5386-2610-8. DOI: `10.1109/3DV.2017.00046`. [Online]. Available: `https://ieeexplore.ieee.org/document/8374587/`.

[25] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, “Depth from combining defocus and correspondence using light-field cameras”, in *2013 IEEE International Conference on Computer Vision*, IEEE, 2013, pp. 673–680, ISBN: 978-1-4799-2840-8. DOI: `10.1109/ICCV.2013.89`. [Online]. Available: `http://ieeexplore.ieee.org/document/6751193/`.

[26] M. Strecke, A. Alperovich, and B. Goldluecke, “Accurate depth and normal maps from occlusion-aware focal stack symmetry”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 2529–2537, ISBN: 978-1-5386-0457-1. DOI: `10.1109/CVPR.2017.271`. [Online]. Available: `http://ieeexplore.ieee.org/document/8099754/`.

[27] Williem, I. K. Park, and K. M. Lee, “Robust light field depth estimation using occlusion-noise aware data costs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2484–2497, 2018, ISSN: 0162-8828. DOI: `10.1109/TPAMI.2017.2746858`. [Online]. Available: `https://ieeexplore.ieee.org/document/8022875/`.

[28] S. B. Kang, R. Szeliski, and J. Chai, “Handling occlusions in dense multi-view stereo”, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, 2001, pp. I–103–I–110, ISBN: 0-7695-1272-0. DOI: `10.1109/CVPR.2001.990462`. [Online]. Available: `http://ieeexplore.ieee.org/document/990462/`.

[29] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003, ISSN: 0162-8828. DOI: `10.1109/TPAMI.2003.1206509`. [Online]. Available: `http://ieeexplore.ieee.org/document/1206509/`.

[30] J. Navarro and A. Buades, “Robust and dense depth estimation for light field images”, *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1873–1886, 2017, ISSN: 1057-7149. DOI: `10.1109/TIP.2017.2666041`. [Online]. Available: `http://ieeexplore.ieee.org/document/7847308/`.

[31] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields", in *Computer Vision – ACCV 2016*, Cham: Springer International Publishing, 2017, pp. 19–34, ISBN: 978-3-319-54186-0. DOI: `10.1007/978-3-319-54187-7_2`. [Online]. Available: `http://link.springer.com/10.1007/978-3-319-54187-7_2`.

[32] J. Kramer, N. Burrus, F. Echtler, H. C. Daniel, and M. Parker, "Point clouds, part 1", in *Hacking the Kinect*, Berkeley: Apress, 2012, pp. 127–150, ISBN: 978-1-4302-3867-6. DOI: `10.1007/978-1-4302-3868-3_7`. [Online]. Available: `http://link.springer.com/10.1007/978-1-4302-3868-3_7`.

[33] R. Ziegler, S. Bucheli, L. Ahrenberg, M. Magnor, and M. Gross, "A bidirectional light field - hologram transform", *Computer Graphics Forum*, vol. 26, no. 3, pp. 435–446, DOI: `10.1111/j.1467-8659.2007.01066.x`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01066.x`.

[34] J. Mäkinen, *From light fields to wavefields: Hologram generation using multiperspective images*, Master's thesis, Tampere, 2017. [Online]. Available: `http://URN.fi/URN:NBN:fi:tty-201705261552`.

[35] M. Breinig, *Coherence*, Knoxville, 2015. [Online]. Available: `http://electron6.phys.utk.edu/optics421/modules/m5/Coherence.htm`.

[36] A. Gilles, *Fast hologram synthesis methods for realistic 3D visualization*, PhD thesis, Rennes, 2016. [Online]. Available: `https://tel.archives-ouvertes.fr/tel-01392677`.

[37] D. Leseberg and C. Frère, "Computer-generated holograms of 3-d objects composed of tilted planar segments", *Applied Optics*, vol. 27, no. 14, pp. 3020–3024, 1988, ISSN: 0003-6935. DOI: `10.1364/AO.27.003020`. [Online]. Available: `https://www.osapublishing.org/abstract.cfm?URI=ao-27-14-3020`.

[38] A. Asundi and C. Zuo, "Digital holography to light field", in *Proceedings of SPIE*, vol. 9132, 2014, 91320U–1–91320U–8. DOI: `10.1117/12.2059775`. [Online]. Available: `https://doi.org/10.1117/12.2059775`.

[39] S. Hamann, L. Shi, O. Solgaard, and G. Wetzstein, "Time-multiplexed light field synthesis via factored wigner distribution function", *Optics Letters*, vol. 43, no. 3, pp. 599–602, 2018, ISSN: 0146-9592. DOI: `10.1364/OL.43.000599`. [Online]. Available: `https://www.osapublishing.org/abstract.cfm?URI=ol-43-3-599`.

[40] P. W. M. Tsang, T.-C. Poon, and Y. M. Wu, "Review of fast methods for point-based computer-generated holography [invited]", *Photonics Research*, vol. 6, no. 9, pp. 837–846, 2018, ISSN: 2327-9125. DOI: `10.1364/PRJ.6.000837`. [Online]. Available: `https://www.osapublishing.org/abstract.cfm?URI=prj-6-9-837`.

[41] Z. Zeng, H. Zheng, Y. Yu, and A. K. Asundi, "Off-axis phase-only holograms of 3D objects using accelerated point-based fresnel diffraction algorithm", *Optics and Lasers in Engineering*, vol. 93, pp. 47–54, 2017, ISSN: 0143-8166. DOI: `10.1016/j.optlaseng.2017.01.006`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0143816616302445`.

[42] T. Shimobaba, T. Kakue, and T. Ito, “Point cloud-based hologram calculation, WASABI and WRP methods”, in *Digital Holography and Three-Dimensional Imaging*, Washington, D.C: OSA, 2017, M3B.2, ISBN: 978-1-943580-28-6. DOI: 10.1364/DH.2017.M3B.2. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=DH-2017-M3B.2.

[43] D. Arai, T. Shimobaba, T. Nishitsuji, T. Kakue, N. Masuda, and T. Ito, “An accelerated hologram calculation using the wavefront recording plane method and wavelet transform”, *Optics Communications*, vol. 393, pp. 107–112, 2017, ISSN: 0030-4018. DOI: 10.1016/j.optcom.2017.02.038. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0030401817301256.

[44] F. Murgia and D. Giusto, “A comparison of raw light field lossless data compression algorithms”, in *2016 24th Telecommunications Forum (TELFOR)*, IEEE, 2016, pp. 1–4, ISBN: 978-1-5090-4086-5. DOI: 10.1109/TELFOR.2016.7818796. [Online]. Available: http://ieeexplore.ieee.org/document/7818796/.

[45] C. Conti, L. D. Soares, P. Nunes, C. Perra, P. A. Assunção, M. Sjöström, Y. Li, R. Olsson, and U. Jennehag, “Light field image compression”, in *3D Visual Content Creation, Coding and Delivery*, Cham: Springer International Publishing, 2019, pp. 143–176, ISBN: 978-3-319-77841-9. DOI: 10.1007/978-3-319-77842-6_6. [Online]. Available: http://link.springer.com/10.1007/978-3-319-77842-6_6.

[46] I. Viola, M. Rerabek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi, “Objective and subjective evaluation of light field image compression algorithms”, in *2016 Picture Coding Symposium (PCS)*, IEEE, 2016, pp. 1–5, ISBN: 978-1-5090-5966-9. DOI: 10.1109/PCS.2016.7906379. [Online]. Available: http://ieeexplore.ieee.org/document/7906379/.

[47] H. Amirpour, A. M. G. Pinheiro, M. Pereira, and M. Ghanbari, “Fast and efficient lenslet image compression”, *CoRR*, vol. abs/1901.11396, 2019. [Online]. Available: http://arxiv.org/abs/1901.11396.

[48] L. Li and Z. Li, “Light field image compression”, in *Handbook of Dynamic Data Driven Applications Systems*, Cham: Springer International Publishing, 2018, pp. 547–570, ISBN: 978-3-319-95503-2. DOI: 10.1007/978-3-319-95504-9_24. [Online]. Available: http://link.springer.com/10.1007/978-3-319-95504-9_24.

[49] C. Conti, L. D. Soares, and P. Nunes, “HEVC-based 3D holoscopic video coding using self-similarity compensated prediction”, *Signal Processing: Image Communication*, vol. 42, pp. 59–78, 2016, ISSN: 0923-5965. DOI: 10.1016/j.image.2016.01.008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0923596516000114.

[50] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, “Compressive light field photography using overcomplete dictionaries and optimized projections”, *ACM Transactions on Graphics*, vol. 32, no. 4, article no. 46, 2013, ISSN: 0730-0301. DOI: 10.1145/2461912.2461914. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2461912.2461914.

[51] 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*, vol. 11, no. 7, pp. 926–954, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2747126. [Online]. Available: http://ieeexplore.ieee.org/document/8022901/.

[52] R. S. Higa, R. F. L. Chavez, R. B. Leite, R. Arthur, and Y. Iano, "Plenoptic image compression comparison between JPEG, JPEG2000 and SPITH", *Cyber Journals: JSAT*, vol. 3, no. 6, pp. 1–6, 2013. [Online]. Available: http://www.cyberjournals.com/Papers/Jun2013/04.pdf.

[53] 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)*, IEEE, 2017, pp. 27–28, ISBN: 978-1-5090-5544-9. DOI: 10.1109/ICCE.2017.7889217. [Online]. Available: http://ieeexplore.ieee.org/document/7889217/.

[54] 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, pp. 1231–1234, ISBN: 978-1-4673-6997-8. DOI: 10.1109/ICASSP.2015.7178166. [Online]. Available: http://ieeexplore.ieee.org/document/7178166/.

[55] X. Jin, H. Han, and Q. Dai, "Plenoptic image coding using macropixel-based intra prediction", *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3954–3968, 2018, ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2832449. [Online]. Available: https://ieeexplore.ieee.org/document/8353484/.

[56] A. Gelman, P. L. Dragotti, and V. Velisavljević, "Centralized and interactive compression of multiview images", in *Proceedings of SPIE*, vol. 8135, 2011, 81350J–1–81350J–15. DOI: 10.1117/12.895982. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.895982.

[57] W. Ahmad, R. Olsson, and M. Sjostrom, "Interpreting plenoptic images as multiview sequences for improved compression", in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4557–4561, ISBN: 978-1-5090-2175-8. DOI: 10.1109/ICIP.2017.8297145. [Online]. Available: http://ieeexplore.ieee.org/document/8297145/.

[58] X. Huang, P. An, L. Shen, and K. Li, "Light field image compression scheme based on MVD coding standard", in *Advances in Multimedia Information Processing – PCM 2017*, Cham: Springer International Publishing, 2018, pp. 79–88, ISBN: 978-3-319-77379-7. DOI: 10.1007/978-3-319-77380-3_8. [Online]. Available: http://link.springer.com/10.1007/978-3-319-77380-3_8.

[59] C. Perra, "Light field image compression based on preprocessing and high efficiency coding", in *2016 24th Telecommunications Forum (TELFOR)*, IEEE, 2016, pp. 1–4, ISBN: 978-1-5090-4086-5. DOI: 10.1109/TELFOR.2016.7818930. [Online]. Available: http://ieeexplore.ieee.org/document/7818930/.

[60] 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)*, IEEE, 2017, pp. 1–5, ISBN: 978-1-5090-4937-0. DOI: 10.1109/BMSB.2017.7986144. [Online]. Available: http://ieeexplore.ieee.org/document/7986144/.

[61] J. M. Santos, P. A. A. Assuncao, L. A. da Silva Cruz, L. Távora, R. Fonseca-Pinto, and S. M. M. Faria, “Performance evaluation of light field pre-processing methods for lossless standard coding”, *MMTC Communications-Frontiers*, vol. 12, no. 4, pp. 44–49, 2017. [Online]. Available: https://web.njit.edu/~xs47/Publication/MMTC_Communication_Frontier_July_2017_singlecolumn_v1.pdf\#page=44.

[62] J. Hou, J. Chen, and L.-P. Chau, “Light field image compression based on bi-level view compensation with rate-distortion optimization”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 517–530, 2019, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2018.2802943. [Online]. Available: https://ieeexplore.ieee.org/document/8283506/.

[63] C. Jia, X. Zhang, S. Wang, S. Wang, and S. Ma, “Light field image compression using generative adversarial network-based view synthesis”, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 177–189, 2019, ISSN: 2156-3357. DOI: 10.1109/JETCAS.2018.2886642. [Online]. Available: https://ieeexplore.ieee.org/document/8574895/.

[64] Y. Y. Liu, C. Zhu, and M. Mao, “Light field image compression based on quality aware pseudo-temporal sequence”, *Electronics Letters*, vol. 54, no. 8, pp. 500–501, 2018, ISSN: 0013-5194. DOI: 10.1049/el.2017.4560. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/el.2017.4560.

[65] 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 & Expo Workshops (ICMEW)*, IEEE, 2016, pp. 1–4, ISBN: 978-1-5090-1552-8. DOI: 10.1109/ICMEW.2016.7574671. [Online]. Available: http://ieeexplore.ieee.org/document/7574671/.

[66] S. Shi, P. Gioia, and G. Madec, “Efficient compression method for integral images using multi-view video coding”, in *2011 18th IEEE International Conference on Image Processing*, IEEE, 2011, pp. 137–140, ISBN: 978-1-4577-1303-3. DOI: 10.1109/ICIP.2011.6115695. [Online]. Available: http://ieeexplore.ieee.org/document/6115695/.

[67] R. Verhack, T. Sikora, L. Lange, R. Jongebloed, G. V. Wallendael, and P. Lambert, “Steered mixture-of-experts for light field coding, depth estimation, and processing”, in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2017, pp. 1183–1188, ISBN: 978-1-5090-6067-2. DOI: 10.1109/ICME.2017.8019442. [Online]. Available: http://ieeexplore.ieee.org/document/8019442/.

[68] C. Jia, Y. Yang, X. Zhang, S. Wang, S. Wang, and S. Ma, “Light field image compression with sub-apertures reordering and adaptive reconstruction”, in *Advances in Multimedia Information Processing – PCM 2017*, Cham: Springer International Publishing, 2018, pp. 47–55, ISBN: 978-3-319-77379-7. DOI: 10.1007/978-3-319-77380-3_5. [Online]. Available: http://link.springer.com/10.1007/978-3-319-77380-3_5.

[69] C. Jia, Y. Yang, X. Zhang, X. Zhang, S. Wang, S. Wang, and S. Ma, “Optimized inter-view prediction based light field image compression with adaptive reconstruction”, in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4572–4576, ISBN: 978-1-5090-2175-8. DOI: 10.1109/ICIP.2017.8297148. [Online]. Available: http://ieeexplore.ieee.org/document/8297148/.

[70] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression", in *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2016, pp. 1–4, ISBN: 978-1-5090-1552-8. DOI: 10.1109/ICMEW.2016.7574674. [Online]. Available: http://ieeexplore.ieee.org/document/7574674/.

[71] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-d hierarchical coding structure for light-field image compression", *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1107–1119, 2017, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2017.2725198. [Online]. Available: http://ieeexplore.ieee.org/document/7972959/.

[72] A. Miyazawa, Y. Kameda, T. Ishikawa, I. Matsuda, and S. Itoh, "Lossless coding of light field camera data captured with a micro-lens array and a color filter", in *2018 International Workshop on Advanced Image Technology (IWAIT)*, IEEE, 2018, pp. 1–4, ISBN: 978-1-5386-2615-3. DOI: 10.1109/IWAIT.2018.8369695. [Online]. Available: https://ieeexplore.ieee.org/document/8369695/.

[73] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Compression of plenoptic point clouds", *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1419–1427, 2019, ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2877486. [Online]. Available: https://ieeexplore.ieee.org/document/8502100/.

[74] T. Ochotta and D. Saupe, "Compression of point-based 3d models by shape-adaptive wavelet coding of multi-height fields", in *Symposium on Point-Based Graphics 2004*, Aire-la-Ville: Eurographics Association, 2004, pp. 103–112, ISBN: 3-905673-09-6. DOI: 10.2312/SPBG/SPBG04/103-112.

[75] R. Schnabel and R. Klein, "Octree-based point-cloud compression", in *Symposium on Point-Based Graphics*, Aire-la-Ville: The Eurographics Association, 2006, pp. 111–121, ISBN: 1811-7813. DOI: 10.2312/SPBG/SPBG06/111-120.

[76] Y. Huang, J. Peng, C.-C. Kuo, and M. Gopi, "A generic scheme for progressive point cloud coding", *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 440–453, 2008, ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.70441. [Online]. Available: http://ieeexplore.ieee.org/document/4378368/.

[77] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams", in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 778–785, ISBN: 978-1-4673-1405-3. DOI: 10.1109/ICRA.2012.6224647. [Online]. Available: http://ieeexplore.ieee.org/document/6224647/.

[78] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, 2017, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2016.2543039. [Online]. Available: http://ieeexplore.ieee.org/document/7434610/.

[79] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2019, ISSN:

2156-3357. DOI: 10.1109/JETCAS.2018.2885981. [Online]. Available: https://ieeexplore.ieee.org/document/8571288/.

[80] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform", in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 2066–2070, ISBN: 978-1-4799-5751-4. DOI: 10.1109/ICIP.2014.7025414. [Online]. Available: http://ieeexplore.ieee.org/document/7025414/.

[81] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based motion estimation and compensation for dynamic 3D point cloud compression", in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 3235–3239, ISBN: 978-1-4799-8339-1. DOI: 10.1109/ICIP.2015.7351401. [Online]. Available: http://ieeexplore.ieee.org/document/7351401/.

[82] ——, "Graph-based compression of dynamic 3D point cloud sequences", *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016, ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2529506. [Online]. Available: http://ieeexplore.ieee.org/document/7405340/.

[83] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a Gaussian process model", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3507–3517, 2017, ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2699922. [Online]. Available: http://ieeexplore.ieee.org/document/7914676/.

[84] ——, "Compression of 3D point clouds using a region-adaptive hierarchical transform", *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016, ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2575005. [Online]. Available: http://ieeexplore.ieee.org/document/7482691/.

[85] X. Sun, H. Ma, Y. Sun, and M. Liu, "A novel point cloud compression algorithm based on clustering", *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2132–2139, 2019, ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2900747. [Online]. Available: https://ieeexplore.ieee.org/document/8648155/.

[86] T. Golla and R. Klein, "Real-time point cloud compression", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 5087–5092, ISBN: 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7354093. [Online]. Available: http://ieeexplore.ieee.org/document/7354093/.

[87] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, "Compressing continuous point cloud data using image compression methods", in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 1712–1719, ISBN: 978-1-5090-1889-5. DOI: 10.1109/ITSC.2016.7795789. [Online]. Available: http://ieeexplore.ieee.org/document/7795789/.

[88] ——, "Continuous point cloud data compression using SLAM based prediction", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 1744–1751, ISBN: 978-1-5090-4804-5. DOI: 10.1109/IVS.2017.7995959. [Online]. Available: http://ieeexplore.ieee.org/document/7995959/.

[89] J.-K. Ahn, K.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Large-scale 3D point cloud compression using adaptive radial distance prediction in hybrid coordinate domains", *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 422–434, 2015, ISSN: 1932-4553. DOI: `10.1109/JSTSP.2014.2370752`. [Online]. Available: `http://ieeexplore.ieee.org/document/6955830/`.

[90] V. Morell, S. Orts, M. Cazorla, and J. Garcia-Rodriguez, "Geometric 3D point cloud compression", *Pattern Recognition Letters*, vol. 50, pp. 55–62, 2014, ISSN: 0167-8655. DOI: `10.1016/j.patrec.2014.05.016`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S016786551400172X`.

[91] X. Zhang, W. Wan, and X. An, "Clustering and DCT based color point cloud compression", *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 41–49, 2017, ISSN: 1939-8018. DOI: `10.1007/s11265-015-1095-0`. [Online]. Available: `http://link.springer.com/10.1007/s11265-015-1095-0`.

[92] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data", *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013, ISSN: 1057-7149. DOI: `10.1109/TIP.2013.2264820`. [Online]. Available: `http://ieeexplore.ieee.org/document/6519266/`.

[93] P. Schelkens, T. Ebrahimi, A. Gilles, P. Gioia, K.-J. Oh, F. Pereira, C. Perra, and A. M. G. Pinheiro, "JPEG Pleno, Providing representation interoperability for holographic applications and devices", *ETRI Journal*, vol. 41, no. 1, pp. 93–108, 2019, ISSN: 1225-6463. DOI: `10.4218/etrij.2018-0509`. [Online]. Available: `http://doi.wiley.com/10.4218/etrij.2018-0509`.

[94] F. Dufaux, Y. Xing, B. Pesquet-Popescu, and P. Schelkens, "Compression of digital holographic data, an overview", in *Proceedings of SPIE*, vol. 9599, 2015, pp. 95990I–1–95990I–11. DOI: `10.1117/12.2190997`. [Online]. Available: `http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2190997`.

[95] D. Blinder, A. Ahar, A. Symeonidou, Y. Xing, T. Bruylants, C. Schreites, B. Pesquet-Popescu, F. Dufaux, A. Munteanu, and P. Schelkens, "Open access database for experimental validations of holographic compression engines", in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, IEEE, 2015, pp. 1–6, ISBN: 978-1-4799-8958-4. DOI: `10.1109/QoMEX.2015.7148145`. [Online]. Available: `http://ieeexplore.ieee.org/document/7148145/`.

[96] T. J. Naughton, Y. Frauel, B. Javidi, and E. Tajahuerce, "Compression of digital holograms for three-dimensional object reconstruction and recognition", *Applied Optics*, vol. 41, no. 20, pp. 4124–4132, 2002, ISSN: 0003-6935. DOI: `10.1364/AO.41.004124`. [Online]. Available: `https://www.osapublishing.org/abstract.cfm?URI=ao-41-20-4124`.

[97] Y. Xing, B. Pesquet-Popescu, and F. Dufaux, "Comparative study of scalar and vector quantization on different phase-shifting digital holographic data representations", in *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, IEEE, 2014, pp. 1–4, ISBN: 978-1-4799-4758-4. DOI: `10.1109/3DTV.2014.6874769`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6874769`.

[98] P. A. Cheremkhin and E. A. Kurbatova, “Numerical comparison of scalar and vector methods of digital hologram compression”, in *Proceedings of SPIE*, vol. 10022, 2016, pp. 1 002 227–1–1002227–10. DOI: 10.1117/12.2246411. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2246411.

[99] D. Blinder, T. Bruylants, E. Stijns, H. Ottevaere, and P. Schelkens, “Wavelet coding of off-axis holographic images”, in *Proceedings of SPIE*, vol. 8856, 2013, pp. 88561L–1–88561L–12. DOI: 10.1117/12.2027114. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2027114.

[100] N. Chamakhi, I. Bouzidi, A. O. Zaid, and F. Dufaux, “JPEG based compression of digital holograms”, in *2018 7th European Workshop on Visual Information Processing (EUVIP)*, IEEE, 2018, pp. 1–6, ISBN: 978-1-5386-6897-9. DOI: 10.1109/EUVIP.2018.8611713. [Online]. Available: https://ieeexplore.ieee.org/document/8611713/.

[101] L. T. Bang, Z. Ali, P. D. Quang, J.-H. Park, and N. Kim, “Compression of digital hologram for three-dimensional object using wavelet-bandelets transform”, *Optics Express*, vol. 19, no. 9, pp. 8019–8031, 2011, ISSN: 1094-4087. DOI: 10.1364/OE.19.008019. [Online]. Available: https://www.osapublishing.org/oe/abstract.cfm?uri=oe-19-9-8019.

[102] D. Blinder, T. Bruylants, H. Ottevaere, A. Munteanu, and P. Schelkens, “JPEG 2000-based compression of fringe patterns for digital holographic microscopy”, *Proceedings of SPIE*, vol. 53, no. 12, pp. 123 102–1–123102–13, 2014, ISSN: 0091-3286. DOI: 10.1117/1.OE.53.12.123102. [Online]. Available: http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/1.OE.53.12.123102.

[103] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux, “Adaptive nonseparable vector lifting scheme for digital holographic data compression”, *Applied Optics*, vol. 54, no. 1, A98–A109, 2015, ISSN: 1559-128X. DOI: 10.1364/AO.54.000A98. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=ao-54-1-A98.

[104] L. Onural, “Diffraction from a wavelet point of view”, *Optics Letters*, vol. 18, no. 11, pp. 846–848, 1993, ISSN: 0146-9592. DOI: 10.1364/OL.18.000846. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=ol-18-11-846.

[105] M. Liebling, T. Blu, and M. Unser, “Fresnelets, new multiresolution wavelet bases for digital holography”, *IEEE Transactions on Image Processing*, vol. 12, no. 1, pp. 29–43, 2003, ISSN: 1057-7149. DOI: 10.1109/TIP.2002.806243. [Online]. Available: http://ieeexplore.ieee.org/document/1187353/.

[106] E. Darakis and J. J. Soraghan, “Use of fresnelets for phase-shifting digital hologram compression”, *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3804–3811, 2006, ISSN: 1057-7149. DOI: 10.1109/TIP.2006.884918. [Online]. Available: http://ieeexplore.ieee.org/document/4011953/.

[107] J. P. Peixeiro, C. Brites, J. Ascenso, and F. Pereira, "Holographic data coding, benchmarking and extending HEVC with adapted transforms", *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 282–297, 2018, ISSN: 1520-9210. DOI: 10.1109/TMM.2017.2742701. [Online]. Available: http://ieeexplore.ieee.org/document/8013856/.

[108] D.-H. Lee, J.-Y. Sim, C.-S. Kim, and S.-U. Lee, "Viewing angle dependent coding of digital holograms", in *2011 19th European Signal Processing Conference*, 2011, pp. 1367–1371. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7074151.

[109] K. Viswanathan, P. Gioia, and L. Morin, "Wavelet compression of digital holograms, Towards a view-dependent framework", in *Proceedings of SPIE*, vol. 8856, 2013, 88561N–1–88561N–10. DOI: 10.1117/12.2027199. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2027199.

[110] A. E. Rhammad, P. Gioia, A. Gilles, M. Cagnazzo, and B. Pesquet-Popescu, "Color digital hologram compression based on matching pursuit", *Applied Optics*, vol. 57, no. 17, pp. 4930–4942, 2018, ISSN: 1559-128X. DOI: 10.1364/AO.57.004930. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=ao-57-17-4930.

[111] ——, "View-dependent compression of digital hologram based on matching pursuit", in *Proceedings of SPIE*, vol. 10679, SPIE, 2018, pp. 106790L–1–106790L–14, ISBN: 978-1-5106-1884-8. DOI: 10.1117/12.2315233. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10679/2315233/View-dependent-compression-of-digital-hologram-based-on-matching-pursuit/10.1117/12.2315233.full.

[112] F. Bellard, *Bpg image format*, 2019. [Online]. Available: https://bellard.org/bpg/.

[113] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon, "Accurate depth map estimation from a lenslet light field camera", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 1547–1555, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298762. [Online]. Available: http://ieeexplore.ieee.org/document/7298762/.

[114] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras", in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2015, pp. 3487–3495, ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.398. [Online]. Available: http://ieeexplore.ieee.org/document/7410755/.

[115] W. Williem and I. K. Park, "Robust light field depth estimation for noisy scene with occlusion", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 4396–4404, ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.476. [Online]. Available: http://ieeexplore.ieee.org/document/7780845/.