Dissertation

submitted

to the

Combined Faculty for the Natural Sciences

and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

submitted by

M.Sc. Alessandro Vianello

born in Dolo (Italy)

Oral examination:

Robust 3D Surface Reconstruction from Light Fields

Advisors: Prof. Dr. Bernd Jähne
Prof. Dr. Karl-Heinz Brenner

Abstract

Light field data captures the intensity, as well as the direction of rays in 3D space, allowing to retrieve not only the 3D geometry information, but also the reflectance properties of the acquired scene. The main focus of this thesis is precise 3D geometry reconstruction from light fields, especially on scenes with specular objects.

A new semi-global approach for 3D reconstruction from *linear light fields* is proposed. This method combines a modified version of the Progressive Probabilistic Hough Transform with local slope estimates to extract orientations, and consequently depth information, in *epipolar plane images* (EPIs). The resulting reconstructions achieve a higher accuracy than local methods, with a more precise localization of object boundaries, as well as preservation of fine details.

In the second part of the thesis the proposed approach is extended to *circular light fields* in order to determine the full 360° view of target objects. Additionally, circular light fields allow retrieving depth even from datasets acquired with telecentric lenses, a task which is not possible using a linearly moving camera. Experimental results on synthetic and real datasets demonstrate the quality and the robustness of the proposed algorithm, which provides precise reconstructions even with highly specular objects.

The quality of the final reconstruction opens up many possible application scenarios, such as precise 3D reconstruction for defect detection in industrial optical inspection, object scanning for heritage preservation, as well as depth segmentation for the movie industry.

Zusammenfassung

Lichtfelddaten bilden sowohl die Intensität als auch die Richtung von Strahlen im dreidimensionalen Raum ab. Daher erlauben sie nicht nur die Rekonstruktion von 3D Geometrieinformationen sonder auch von Reflektanzeigenschaften der aufgenommenen Szene. Der Schwerpunkt dieser Arbeit ist die Gewinnung präziser 3D Geometrie aus Lichtfeldern, besonders bei Szenen mit spiegelnden Objekten.

Einen neuen semiglobalen Ansatz zur 3D Rekonstruktion aus linearen Lichtfeldern wird vorgeschlagen. Diese Methode kombiniert eine modifizierte Variante der Progressive Probabilistic Hough Transform mit lokalen Schätzungen der Orientierung um damit die Tiefe aus Epipolar Bildern zu extrahieren. Die Rekonstruktion erzielt eine höhere Genauigkeit als lokale Methoden, und erhält feine Details.

Im zweiten Teil der Arbeit wird der vorgeschlagene Ansatz auf kreisförmige Lichtfelder ausgedehnt, um die komplette Ansicht des Objektes zu bestimmen. Darüber hinaus ermöglichen zirkuläre Lichtfelder es Tiefendaten auch aus Datensätzen zu gewinnen, die mit einem telezentrischen Objektiv aufgenommen wurden, was mit einer sich linear bewegten Kamera nicht möglich ist. Experimentelle Ergebnisse demonstrieren die Güte und die Robustheit des vorgestellten Algorithmus, welcher selbst bei hochgradig spekularen Objekten präzise Rekonstruktionen liefert.

Die Qualität der endgültigen Rekonstruktion eröffnet viele mögliche Anwendungsszenarien wie zum Beispiel die genaue 3D Rekonstruktion zur Defekterkennung in der industriellen optischen Inspektion, das Einscannen von Objekten zur Bewahrung des kulturellen Erbes, sowie die Tiefenschätzung für die Filmindustrie.

A te che sei il mio grande amore ed il mio amore grande,
a te che hai preso la mia vita e ne hai fatto molto di più,
a te che hai dato senso al tempo senza misurarlo,
a te che sei il mio amore grande ed il mio grande amore.

A te, Milica.

Acknowledgements

First of all I would like to thank my advisor Professor Dr. Bernd Jähne for giving me the possibility of doing a PhD at Heidelberg University, as well as for his great support during these three years. He has been an inspiring and very optimistic mentor, who also helped me to stay always highly motivated. I am also deeply grateful to Dr. Ralf Zink and Robert Bosch GmbH for believing in me and funding this research.

Thank you to all the Bosch colleagues of the CR/APA2 team for being helpful when needed, especially to Dr. Jens Ackermann who advised me in the last part of my PhD studies. It was a pleasure to collaborate with my colleagues at the Heidelberg Collaboratory for Image Processing, an inspiring place to do research. A special thanks to Priv.-Doz. Dr. Christoph Garbe, Dr. Maximilian Diebold, Marcel Gutsche, Hendrik Schilling, Hamza Aziz Ahmad, as well as Giulio Manfredi for the proficient collaboration within his Master's thesis.

I would like to thank my family, Fabio, Antonietta, and Ilaria, for always believing in me, loving me, and supporting me in all the decisions that I made, even the ones that at first glance appeared wrong and too risky.

This PhD was an incredible journey, with many challenges and obstacles. I had the good fortune to have at my side a very special person, who accompanied me on this journey. Thank you Milica for being here all the time and for giving me, with your love and determination, the strength to overcome all the difficulties and reach this important goal. To you Milica, my deepest gratitude.

Contents

Chapter 1

Introduction

1.1 Introduction

The history of photography has deep roots in the antiquity. Leonardo Da Vinci [25] proposed the first description of the *camera obscura* in his writing notebook in 1502. However, the pinhole camera principle was probably already known to the Greek mathematicians Aristotle and Euclid in the 5^{th} and 4^{th} centuries BCE [53]. The first documented and still surviving photo was captured by Nicéphore Niépce in 1826 (or 1827). Since then, photography started to be extensively studied and further developed, leading to many novelties and inventions: the first metal-based commercial photographic process (Louis Daguerre, 1839) which allowed to produce clear pictures within few minutes of exposure time, the paper-based calotype negative and salt print process (Henry Fox Talbot, 1841) which shortened the exposure time to fraction of seconds, the introduction of color photography (Leopold Godowsky Jr. and Leopold Mannes, 1935), and the invention of digital photography (Steven Sasson, 1975). This latest innovation dramatically reduced the delay and the cost of producing a picture, allowing to store, edit, and distribute digital photos by simply using ordinary computers. Later on, thanks the rapid progress in the computer industry, we entered the era of digital image processing, where signal processing theory and algorithms could be used to perform a number of tasks such as pattern recognition, classification, feature extraction, and so on.

Photography revolutionized our world and society, and is now employed in many contexts such as art, journalism, medicine, sport, as well as business and science. However, despite the great number of innovations, photography is still lacking of one fun-

damental component: the depth. When we take a picture, we are simply projecting the 3D world onto the 2D image plane of a camera sensor, losing therefore the third dimension. The first attempts to 3D-photographs are as old as photography itself: two cameras separated by approximately the same distance of the human eyes were used to simulate our vision and acquire stereoscopic images by John Benjamin Dancer in 1856. However, only with digital photography it was possible to develop algorithms able to produce a 2D image showing the distance between the camera and the surface points of the acquired scene. Scientists and researchers proposed algorithms such as stereo and multi-view, which try to find correspondences between scene points projected onto the image planes of two or more cameras placed at different locations. The distance between these correspondences is called *disparity*, and it encodes the depth information. The combination of photography and digital image processing allows to extract a great number of information about the acquired scene, such as 3D geometry, reflectance properties, and incident light. All these properties are enclosed in the flow of light that any object in our world reflects toward all directions in 3D space over time. In 1936 Arun Gershun [38] described this information and defined the function that relates a ray to the radiance it transports as the *light field*. This function was later on named *plenoptic function* by Adelson and Bergen [2]. Unfortunately, Gershun formulated his theory years before the digital computer era, and light field were introduced into computer graphics only in 1996 by both Gortler et al. [41] and Levoy et al. [58] for an image base rendering application. Since then, the scientific community started to apply this concept to many areas of research. Some examples are image based rendering [18, 26, 41, 58], super resolution [13, 40, 95], as well as digital refocusing and all-in-focus imaging [66]. Besides these applications, one the most important tasks where light field can be applied, and the one we are interested in this thesis, is three dimensional reconstruction.

1.2 Motivation

In practical applications, light fields are a dense collection of pinhole views from different locations, therefore they can be considered as multi-view camera systems. However, light fields normally exploit much more views, as well as a more structured sampling of the scene, in order to avoid the problem of determining which parts of one image

correspond to which parts of another image, namely the *correspondence problem*. This is one of the main issues of binocular and multi-view stereo systems, which need to match over large patches to find correspondences within images. In densely sampled light fields this problem is avoided, leading to a more robust and precise correspondence matching. Another major limit of stereo algorithms is their difficulty in finding correspondences in presence of non-Lambertian surfaces. In this case, the intensity of corresponding points changes between the views, based on the *bidirectional reflectance distribution function* (BRDF) of the material. The BRDF describes the distribution of the reflected light on opaque surfaces with respect to the incidence angle and the normal to the surface. Differently from stereo systems, light fields are less influenced by non-Lambertian surfaces, thanks to the use of the redundancy in the acquired data and the smooth angular deviation between two viewpoints. However, highly specular objects are still very challenging even for light field algorithms. One of the first methods to extract depth from densely sampled light fields is the work of Bolles [15], were salient lines were extracted from epipolar-plane images (EPI). Many approaches followed this seminar publication: Criminisi et al. [23] proposed the extraction of EPI-regions by using photoconsistency either in 2D (EPI-strips) or in 3D (EPI-tubes). Wanner [93] used the structure tensor to estimate the local slope of each pixel in the EPI, obtaining a coarse depth map which is then refined by means of a global optimization. Wanner's method was continued by Diebold [30], who introduced a variant of the structure tensor where the inner Gaussian smoothing was replaced by a derivation filter in the x-direction, i.e. the one parallel to the camera motion, in order to be robust against intensity changes along feature paths. Unfortunately, structure tensor methods provide only a local evaluation of EPIs' orientations. This can be a problem especially in noisy datasets, where using all the available information, i.e. the full EPI-line, helps to increase the quality of the final reconstruction. Additionally, global optimization approaches tend to smooth depth discontinuities by averaging between foreground and background disparities, leading to loss of precision at object contours. Štolc et al. [91] proposed to detect EPI-lines by testing a set of slope hypothesis with a block matching approach. Kim et al. [51] compute depth estimates around object boundaries, i.e. in the scene's highly textured areas, by testing all the possible disparity hypotheses and choosing the one that leads to the best color constancy along EPI-lines. All these methods were specifically developed for linear light fields, which are a collection of

images captured along a linear path, and strongly rely on the Lambertian hypothesis. One of the disadvantages of linear light fields is that only one side of the scene can be reconstructed. Therefore, in order to obtain the complete 3D shape, the target object has to be recorded from four different sides, and then the results have to be merged for the final reconstruction. This constrain makes the acquisition procedure long and tedious.

In order to overcome the limitations of linear light fields, some approaches [26, 31] tried to get rid of the linear motion constrain by proposing an unstructured light field approach using a simple hand-held camera. Feldmann et al. [32, 33] used the intensity constancy as a measure to determine valid paths in light field video sequences. In their work, the 3D space is discretized into voxels and then, for each hypothetical 3D point, the algorithm seeks in the image volume if the corresponding path exists. Otherwise, the next voxel is selected until the resulting trajectory fits to the image volume. This method was also adapted to the case of a camera which rotates around the target scene. Crispell et al. [24], and later on Lanman et al. [54], retrieve EPI-trajectories in circular light fields only on depth discontinuities instead of texture edges. They use a contour tracking approach which is not robust to specularities and cannot deal with EPIs having many close trajectories. Similarly to Feldmann et al. [32], Yücer et al. [100] proposed a circular light field setup, where the target object is rotating 360° in front of the camera. They first developed a method to segment objects and compute the visual hull. Then, they extended their approach to estimate the depth also in concave areas, by analyzing the local gradient directions in the image volume [99]. However, also these approaches were developed for Lambertian surfaces, making them unsuitable to reconstruct objects having glossy or specular properties.

In this thesis we address and overcome the described issues by introducing a novel method which is able to produce more accurate and reliable 3D reconstructions out of densely sampled linear light fields. The proposed approach outperforms classical light field algorithms, and is able to deal even with non-Lambertian materials. Additionally, we extend our algorithm to circular light fields, making possible to reconstruct the full 3D shape with just one continuous acquisition. Our method can be applied in fields such as industrial optical inspection, where 3D reconstruction of objects with

non-Lambertian surface properties is often an issue. Besides 3D reconstruction, another application is material classification based on BRDF estimation. In fact, the intensity variation along each trajectory encodes the material properties of the surface. Therefore, the distribution of these intensities can be approximated by mathematical models and associated to a specific BRDF.

1.3 Organization

The thesis is organized in this way. In Chapter 2 we provide an overview of methods for estimating the scene geometry, analysing their advantages and disadvantages. Then, in Chapter 3 we introduce the concept of light field and its mathematical representation. Moreover, practical solution adopted to sample a light field are presented, as well as a detailed explanation on how to retrieve depth information from a light field video sequence. In Chapter 4 we propose a novel algorithm for accurate 3D reconstruction from linear light fields. The method is validated and compared with state-of-the-art light field algorithms on synthetic and real datasets. Then, in Chapter 5 this method is extended to circular light field acquisitions, and the quality of our approach is demonstrated by comparing the 3D reconstructions with state-of-the-art multi-view stereo algorithms. Eventually, in Chapter 6 we conclude the thesis by summarizing the contributions and analysing possible applications and further developments.

Chapter 2

Range Imaging

Range imaging, or depth imaging, is one of the hardest fundamental challenges of computer vision. It consists of a collection of techniques which are used to produce a 2D image showing the distance to surface points on objects in a scene from a known reference point, normally associated with some type of sensor device, the so-called *range camera*. The resulting range image has pixel values which correspond to the distance, e.g. brighter values mean shorter distance, or vice-versa. Depth data plays a crucial role in a number of applications, such as automated driving, human-machine interactions, artificial intelligence, industrial applications, the game and movie industry, as well as common consumer products. Range cameras can be divided in two main categories, *active* and *passive* systems, depending on the need or not of special conditions in terms of scene illumination. In this chapter, these two classes of depth imaging systems will be analysed.

2.1 Active Systems

Active systems use an active light source to illuminate the scene and provide information about the geometry of the visible surfaces. The basic physical principle of these sensors is the same used in electromagnetic and active acoustic sensing: a signal is transmitted to the target scene, reflected, and detected by a sensor. The difference between the sent and received signal encodes the depth of the reflecting surface. In this section, some of the most popular active systems are described.

2.1.1 Laser Range

Laser range cameras acquire the distance information of an object by using a laser beam. Specifically, these devices exploit the *active triangulation* principle by using a laser projector to illuminate a scene with a laser beam and, at the same time, by acquiring the scene with a camera. The laser is mounted on a sweeping device or more commonly in front of a tilting mirror in order to scan the whole scene. The laser beam is reflected from the object and falls on the camera, where is focused on the sensor, usually a *Charge Coupled Device* (CCD) array, through an optical lens. Laser range scanners can provide extremely accurate and dense 3D measurements over a large working volume. However, they can measure a single point at a time, limiting their applications to static scenes only. Moreover, they are quite expensive and difficult to set, especially for large-scale outdoor scenes.

2.1.2 Structured Light

Like laser range systems, structured light cameras estimate the 3D shape of objects using the active triangulation principle. The main difference is that they illuminate the target scene with a known and more complex structure than the simple light beam. A structured light imaging system consists of (at least) one camera and a projection system, which projects a light of a distinct frequency in a particularly structured pattern onto the target surface. This allows to easily distinguish a set of pixels by means of a local coding strategy. An exaustive survey on coded structured light systems is provided by Salvi et al. in [78]. Although a large amount of patterns are available, the most common used structures are composed of one or more parallel laser stripes. The projected light appears distorted from other perspectives than the one of the projector. Therefore, if the camera is not aligned with the projector, the acquired images will have the light pattern distorted according to the distance of the reflecting object surface. The depth of the acquired scene can be computed from the amount of distortion. One of the drawbacks of this method is the high sensitivity to ambient light. To generate patterns with a different light from the ambient one, visible red laser (i.e. $660\,nm$) or infrared (i.e. $760\,nm$) are commonly used [1]. However, structured light still performs better in darkened rooms. Another issue comes from the usage of one camera, which may lead to occlusion problems. As an example, it could happen that the projected

(a) (b) (c) (d)

Figure 2.1: Different types of time-of-flight camera: the PMD CamCube 3.0 [73] with 200 × 200 pixels (a), the MESA Imaging SR4000 [63] with 176 × 144 pixels (b), the Basler ToF Camera [7] having a resolution of 640 × 480 pixels, and the ToF camera from the European Commission funded project ARTTS [5] with 176 × 144 pixel (d).

stripes cannot be seen by the camera due to the shape of the illuminated surface. Eventually, structured light tends perform poorly on highly specular surfaces [64] and in the reconstruction of fine details.

2.1.3 Time-of-Flight

Time-of-flight (ToF) range cameras are relatively new active sensors which allow the acquisition of 3D point clouds at video frame rates. Some recent models of ToF cameras are showed in Figure 2.1. Depth measurements are based on the well-known time-of-flight principle [34]: the time-of-flight τ_d is the time that the light needs to cover the distance d from a light source to an object, and from this object back to the camera. If the light source is assumed to be located near the camera, τ_d can be computed as

$$\tau_d = \frac{2d}{c}, \tag{2.1}$$

where c is the speed of light ($c = 3 \cdot 10^8\, m/s$). According to the camera technology, the ToF method is suitable for ranges starting from some centimeters to several hundreds of meters, with relative accuracies of 0.1%. This means that standard deviations in the millimeter range are realistically achievable at absolute distances of some meters, corresponding to a time-resolution of $6.6\, ps$ [19].

Two types of ToF cameras are available: pulse-based and phase-based, the latter better known as continuous wave [82]. The simplest version are the pulse-based ToF cameras, which directly evaluate τ_d employing discrete pulse of light emitted by a light

source and reflected by the target object. In these devices each pixels has an independent clock, used to measure the time of travelled laser pulse [72]. Pulse-based ToF cameras can be implemented by arrays of Single-Photon Avalanche Diodes (SPADs) [4, 75] or an optical shutter technology [42]. The SPADs high sensitivity enables the sensor to detect low level of reflected light, therefore inexpensive laser sources with milliwatt power can be used for ranges up to several meters [72]. The advantage of using pulsed light is the possibility of transmitting a high amount of energy in a very short time. In this way, the influence of background illumination can be reduced. On the other hand, these systems must be able to produce very short light pulses with fast rise and fall times, which are necessary to assure an accurate detection of the incoming light pulse [89]. Continuous-wave ToF cameras use periodically modulated light sources, and determine the depth by measuring the phase shift between the emitted and the received optical signal. In order to measure this shift, continuous-wave ToF are equipped with special sensors consisting of two quantum wells for every pixel, which store the electrons generated by the incident photons. These electrons are then sorted by an electronic switch, implemented as a variable electrical field, into the one or the other quantum well. The switch is synchronized with the reference modulated signal, thus the number of accumulated electrons in each quantum well corresponds to one sample of the light signal [82]. The special pixels used in continuous-wave ToF, sometimes called *smart pixels*, are larger than the standard ones, yielding to relatively small image resolution (e.g. 640×480 pixels for the Basler ToF Camera [7]).

One of the advantages of these ToF cameras with respect to laser scanners is that they can acquire 3D images without any scanning mechanism and with higher frame rate. However, the depth measured from ToF cameras is affected by many errors, a simplified overview of these problems can be found in [34, 52]. One of the most important is the *photon-shot noise*, which accounts for the statistical Poisson-distributed nature of the arrivals process of photons on the sensor, and affects the measurement precision. Moreover, the estimated depth of a sensor pixel associated to a depth discontinuity area is a combination between the different depth levels of the area, this is the so-called *flying pixels* problem. Eventually, ToF cameras are suffering *multi-path propagation*, which leads to a wrong estimation of the scene's depth [82, 90].

2.2 Passive Systems

In contrast to active systems, passive range imaging systems allow to recover the scene's depth without using any external illumination source, they are easily deployable, and have a lower cost. The drawback is that they require some post-processing steps in order to compute the depth information. The basis of these system is the *passive triangulation* principle: when an object is viewed from two (or more) coplanar viewpoints, the image as observed from one position is shifted laterally when viewed from another other one. The distance between these projections, known as *disparity*, is inversely proportional to the distance between the object's point and the cameras.

2.2.1 Stereo Systems

Stereo vision systems use two standard cameras in order to simulate the human binocular vision and estimate the depth distribution of an acquired scene. An exhaustive overview of stereo systems can be found in [44, 48]. Figure 2.2 shows the imaging of a point in a typical stereo setup. In the figure, two pinhole cameras have parallel optical axes and are laterally shifted by a known distance, the so-called *baseline* b. Moreover, the two cameras have the same *focal length* f. With this configuration, a 3D point $P(X, Y, Z)$ is projected in the two image planes at the locations x_l and x_r. Exploiting similar triangles, the difference between these two projections gives us the disparity

$$d = \Delta x = x_r - x_l = \frac{f\,(X + b)}{Z} - \frac{fX}{Z} = \frac{bf}{Z}. \tag{2.2}$$

From the disparity, the point's depth can be computed by

$$Z = \frac{bf}{d}. \tag{2.3}$$

In Figure 2.2 we assumed that the two projections have the same vertical coordinate in the image planes. However, when two images are acquired with a stereo camera this is not always the case. Figure 2.3 shows the so-called *epipolar geometry* for a two-frame stereo setup. The scene point $P(X, Y, Z)$ is projected into the image plane of each camera in the points $p_l = (x_l, y_l)$ and $p_r = (x_r, y_r)$, respectively. These two points are called *corresponding points* (or conjugate points). The two cameras have coplanar projections centers C_l and C_r, which define the *epipolar plane* π. The intersections between the baseline and the image planes $\mathfrak{I}_l$ and $\mathfrak{I}_r$ define the two *epipoles* e_l and

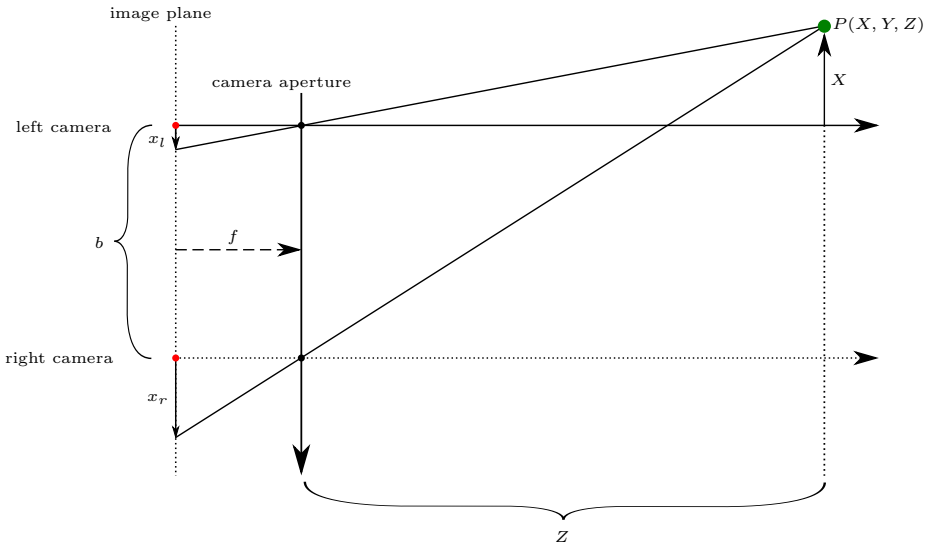


Figure 2.2: Imaging of a point in a typical stereo setup (top view of the XZ plane). The two cameras have the same focal length f, and are displaced horizontally by the baseline b. Both the optical axes are parallel to the Z axis. The 3D world point $P(X, Y, Z)$ is imaged on the position x_l on the left camera, and on the position x_r on the right camera. The difference of these two projections gives the disparity and, consequently, the depth Z of P trough Equations 2.2 and 2.3.

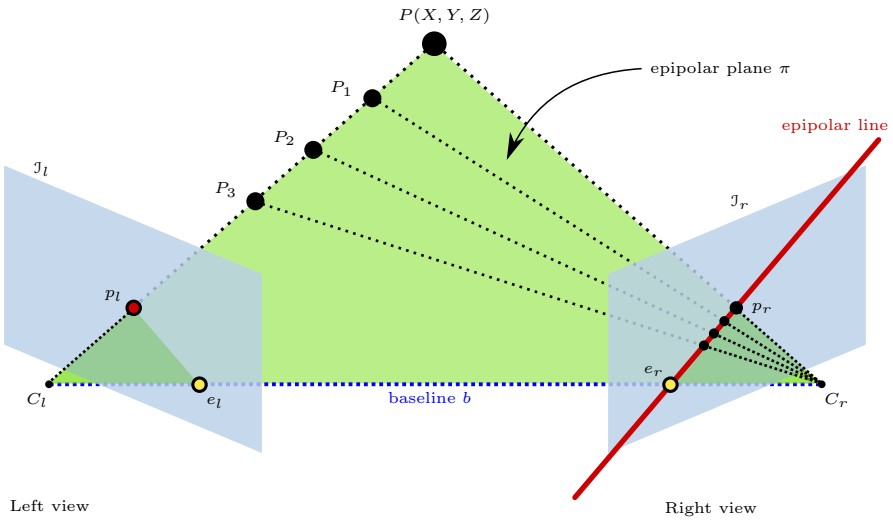


Figure 2.3: Eipolar geometry for a stereo camera setup. Given a left image plane $\mathfrak{I}_l$ with projection center C_l and a right image plane $\mathfrak{I}_r$ with projection center C_r, the line connecting C_l and C_r intersects $\mathfrak{I}_l$ at e_l, and $\mathfrak{I}_r$ at e_r. These two intersection points are called *epipoles*. The green plane containing C_l, C_r and the 3D world point P is called *epipolar plane*. All points lying between P and C_l are projected onto a point p_l on $\mathfrak{I}_l$ and on a line (in red) called *epipolar line* on $\mathfrak{I}_r$, which marks the intersection of the epipolar plane with $\mathfrak{I}_r$. Source: [22].

e_r. All the points on the line between P and C_l are projected in $\mathcal{I}_l$ on p_l. For the right camera, the projections of these points in the image plane $\mathcal{I}_r$ lie on the so-called *epipolar line* (marked in red in the figure), which is defined by the intersection between the epipolar plane π and the image plane $\mathcal{I}_r$. For a given point p_l, the corresponding point on the other view can be found by simply searching along the epipolar line in $\mathcal{I}_r$, reducing the search domain from 2D to 1D, this is also known as the *correspondence problem*. However, in practical applications the image planes of a stereo pair are not coplanar. Therefore, a *rectification* is performed in order to simplify the stereo matching algorithm. Image rectification transforms each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes (usually the horizontal one). The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras in order to generate two coplanar image planes that are also parallel to the baseline. The main challenge of stereo algorithms is to find the conjugate point of every pixel in the so-called reference image ($\mathcal{I}_l$), by searching through the corresponding horizontal line in the so-called target image ($\mathcal{I}_r$). The reliability of point matching is fundamental, because mismatching correspondences lead to wrong depth estimation and gaps in the reconstruction. Stereo algorithms can be subdivided in three main classes: *local*, *global*, and *semi-global*. This subdivision can be stripped-down as a trade-off between robustness and low computational complexity.

2.2.1.1 Local Methods

With local methods the disparity of every point in the reference image is computed by exploiting the local similarity of the intensity values, within a finite window, in the correspondent row of the target image. One of the most used algorithms is the so-called *fixed window* stereo algorithm. With this technique, for each pixel on the reference image its conjugate is searched using a window centered around the pixel. This window is compared with other windows of the same size centered around each possible candidate in the target image, as shown in Figure 2.4. The computed disparity is the shift associated at the maximum similarity between the values of each pair of windows. In order to evaluate this similarity a cost function is normally used, the most commons are *sum of squared differences* (SAD) [27, 67, 101], *normalized cross correlation* [11], and the *Census transform* [55]. The main limitation of these algorithms lies in selecting an appropriate window size, which should be large enough to include sufficient intensity

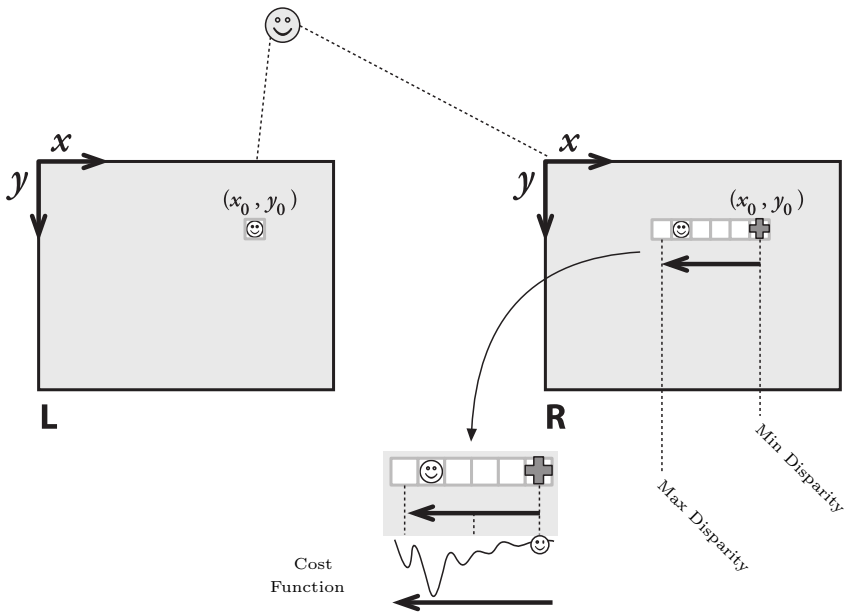


Figure 2.4: Fixed window stereo algorithm: the feature from the left image (location (x_0, y_0)) is used to find the corresponding feature on the right image by shifting a window over a disparity range defined by the minimum and maximum disparity. The conjugate feature is determined by using a cost function which computes the correlation (or the difference) between the two windows. In this case, the shift corresponding to the minimum cost defines the disparity of the feature. Source: [104].

variation and small enough to avoid including scene's depth discontinuity and having projective distortions effects.

2.2.1.2 Global Methods

In contrast to local approaches, global methods compute the whole disparity image D at once by imposing smoothness constraints on the scene's depth in the form of regularized energy functions. In general they formulate the problem as a global energy cost minimization

$$\arg\min_{D} \left(E_{data}\left(I_1, I_2, D\right) + E_{smooth}\left(D\right) \right). \tag{2.4}$$

The term $E_{data}\left(I_1, I_2, D\right)$, called *data term*, measures the similarity between the two images I_1 and I_2 under the condition of a certain disparity D. The term $E_{smooth}\left(D\right)$, called *smoothness term*, is used to regularize the disparity estimation. Generally it is assumed that the disparity map has smooth regions separated by sharp transition along object boarders. An example of regularization is the combined adaptive second order total variation proposed by Lenzen et al. [56, 57] and Papafitsoros [70]. Equation 2.4 leads to an optimization problem which typically has a greater computational than local methods. To solve this problem, many different methods have been developed, such as representing the disparity map as a Markov random field [59] which can be solved by means of belief propagation [86] or using graph cut optimization [16, 17]. Global methods are more robust than the local ones in textureless areas. On the other hand, they are computationally expensive, and the final reconstruction is highly sensitive to the smoothness assumptions, which determine the quality of depth discontinuities at object contours [87].

2.2.1.3 Semi-Global Methods

Semi-global stereo algorithms use a global disparity model like the global methods, but they impose constraints only on a portion of the image, in order to reduce the computational cost. An example of these methods is the *semi-global matching* from Hirschmüller [45], which is based on the idea of pixel-wise matching by using mutual information as matching cost. This algorithm computes many 1D energy functions along different paths (usually 8 or 16), then the functions are minimized and finally their costs are summed up. For each point, the chosen disparity corresponds to the

minimum aggregated cost. As compared to local and global methods, this algorithm is very fast and works well even with textureless regions.

2.2.2 Multi-View Stereo Systems

The two views case discussed so far can be extended to setups where the target scene is acquired by more than two different known camera viewpoints, the so-called multi-view stereo systems, widely described in [84] by Seitz et al.. The acquisition can be done by simultaneously recording a dynamic scene with a camera array, or sequentially capturing a static scene with a moving camera. Multi-view stereo techniques exploit stereo correspondence to recover the full scene geometry, normally as a point cloud, instead of a single depth map of the reference view. In the last two decades, many multi-view stereo algorithms were proposed. These algorithms consider a large number of views, from tens [36, 49, 83, 105] to hundreds or several thousands of images [35, 39, 85]. Among them, Goesele et al. [39] propose an approach which heavily relies on an adaptive view selection to produce high quality reconstructions. Furukawa and Ponce [36] present a method that clusters the images into several sets that can be processed in parallel by generating and propagating a semi-dense set of rectangular patches covering the surfaces visible in the images. In Chapter 4 a new algorithm for 3D reconstruction from multiple views will be introduced, and the methods from Goesele et al. and Furukawa and Ponce will be used for the comparison.

Chapter 3

Light Fields

In this chapter the background and the mathematical definition of light fields will be presented. Moreover, it will be shown how light fields can be acquired and used in the context of image processing, and specifically the relation between the unknown scene depth and the information contained in a light field video sequence. Eventually, the structure tensor approach for EPI-lines local slope estimation will be described.

3.1 The Plenoptic Function

The world is made of three-dimensional objects which are filling the space around them with a dense array of light rays having different intensities. These light rays are travelling independently through the space, and contain all the information that characterizes the scene. A light field can be described by the *plenoptic function*, a seven-dimensional function, introduced by Abelson and Bergen [2], which allows the reconstruction of every possible view, at every moment, from every position, at every wavelength, within the bounds of the space-time wavelength region under consideration. Its mathematical definition is

$$P(\Theta, \phi, \lambda, t, X, Y, Z), \tag{3.1}$$

where (X, Y, Z) indicate the position in space of an object point which is hit by some light, (Θ, ϕ) describe the direction of one of the point's reflected rays, λ is the wavelength, and t denotes the time. In practice, it is possible to use a camera to sample the plenoptic function: a black and white image of a scene is nothing but the accumulation

of the light seen from a single viewpoint, at a single time, averaged over the wavelengths of the visible spectrum. In fact, when taking an image, one simply records the intensity distribution $P(\Theta, \phi)$ within the pencil of light rays passing through the lens. In the following, we will use the concept of plenoptic function to derive a mathematical description of a light field.

3.2 The Lumigraph Parametrization

In practical applications, it is not realistic to sample the plenoptic function defined in Equation 3.1, due to his seven dimensions and the measurement difficulties. However, it is possible to simplify it in order to reduce the number of dimensions. The first dimension is the time t, which can be eliminated by considering static scenes. Additionally, the radiance is usually sampled at three wavelength bands (i.e. red, green, and blue), so the wavelength λ can be ignored as well. After these simplifications, the plenoptic function becomes

$$P(\Theta, \phi, X, Y, Z). \tag{3.2}$$

This is a five dimensional function which depends only on the position and the direction of the light rays, but unfortunately is still not suited for computer graphics. As an example, placing a sensor to measure the radiance in concave parts of the scene will probably block the natural illumination. For these reasons, other assumptions have to be made. In their 1991 paper, Gortler et al. [41] considered the light rays leaving the convex hull of a bounded object, which are propagating in the free space. These light rays can be represented with a two plane system: one plane Π of coordinates (s, t) encoding the camera position in the world coordinate system, and a second plane Ω of coordinates (x, y) encoding the image plane coordinates. This new representation, better known as *Lumigraph* or *4D light field*, is defined as

$$L : \Pi \times \Omega \to \mathbb{R}, \quad (s, t, x, y) \mapsto L(s, t, x, y). \tag{3.3}$$

A graphical representation of the two-plane parametrization of a light ray is showed in Figure 3.1. A 4D light field can be considered as a collection of pinhole views, all located on a common image plane and with parallel viewing direction.

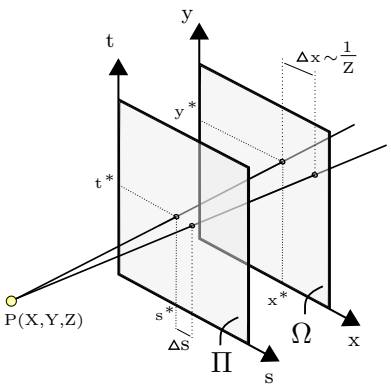


Figure 3.1: With the two-plane parametrization, a light ray can be specified by its two points of intersection with two parallel planes. The plane Π encodes the spatial information and contains the centers of projection of the cameras used to acquire the light field. The plane Ω encodes the location in the image plane. Let us consider two cameras, with the coordinates t and y fixed, i.e. $t = t^*$ and $y = y^*$, and the centers of projection horizontally displaced by Δs (the baseline). The projections of a point $P(X, Y, Z)$ on Ω through (s^*, t^*) and $(s^* + \Delta s, t^*)$ are spaced by Δx. Δx defines the disparity and is inversely proportional to the depth of P. This example corresponds to the classical two-frame stereo setup discussed in Section 2.2.1 Source: [92].

3.3 Epipolar Plane Images

One of the most important applications of light field imaging is extracting the three dimensional geometry of objects and scenes. In practice, the Lumigraph has to be sampled by acquiring a set of images from different viewpoints. One way to do this is separating the 4D light field into horizontal and vertical 3D light fields. Let us consider Figure 3.1, and assume that a vertical world coordinate t^* is fixed on the plane Π. A set of pinhole cameras, all with their center of projections lying on Π, are placed at the height t^*, horizontally spaced by a fixed baseline Δs. When the images from all these cameras are stacked one on top of each other, creating a 3D *image volume* (s, x, y), a 2D slice through this volume at a fixed image plane coordinate y^* equals to the Lumigraph subspace (s, x). This subspace is known as *epipolar-plane image* (EPI), and was firstly defined by Bolles et al. [15]. An example of how an EPI can be extracted from an image volume is presented in Figure 3.2, for the specific case of horizontal camera motion. It can be seen that the flow of the pixels motion lies along straight lines in the EPI-space. The slope of the lines corresponds to the depth of the 3D points on the object tracing out the lines. As for the two-frame stereo Equation 2.3, the relation between the depth

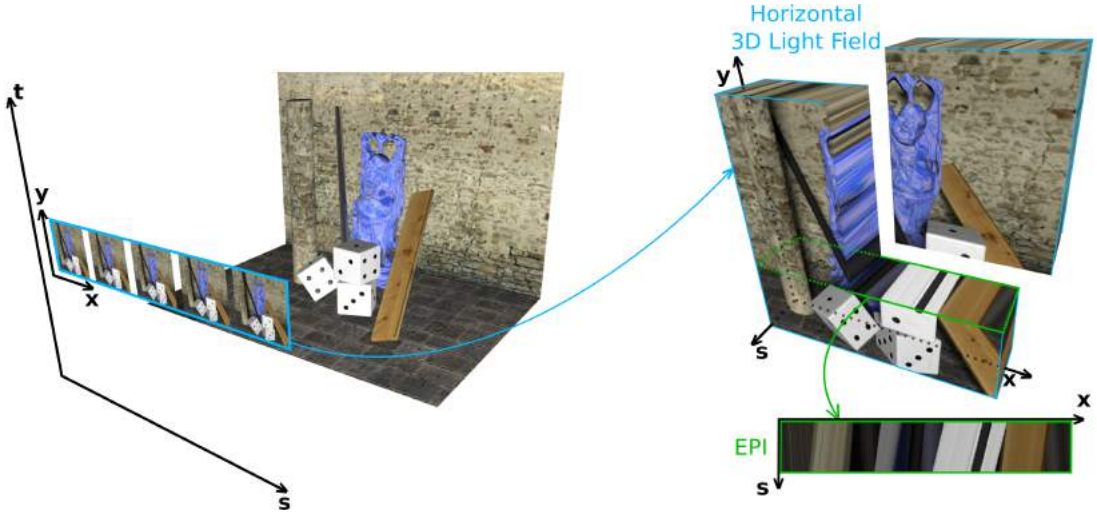


Figure 3.2: Representation of a scene acquired with a horizontal 3D light field. Successive images are stacked into an image volume, and an epipolar-plane image (EPI) can be obtained by horizontally slicing this volume. The same considerations can be applied for the vertical 3D light field case. Source: [28].

of a 3D point and the slope of its corresponding EPI-line is

$$Z = \frac{\Delta s f}{\Delta x}, \tag{3.4}$$

where Δs is the baseline in meters between two neighboring cameras, Δx is the disparity, and the focal f is the distance between the two planes Π and Ω.

In a sx-plane, the disparity of an EPI-line is proportional to the slope of the line itself. Specifically, the disparity corresponds to the shift of line points between two neighboring views. Figure 3.3 shows a simplified representation of an EPI-line in the sx-plane. If we define the orientation θ as the angle between the EPI-line and the s-axis, then the relation between orientation and disparity d is

$$d = \frac{\Delta s}{\Delta x} = \frac{1}{\Delta x} = \tan\theta, \tag{3.5}$$

where in this case $\Delta s = 1\,px$ is simply the distance in the EPI space of two neighbouring views, and Δx is still the disparity. All the considerations of this section can be applied to vertical 3D light fields setups, where a set of cameras are vertically displaced at a fixed coordinate s^*, by simply substituting t with s and y with x.

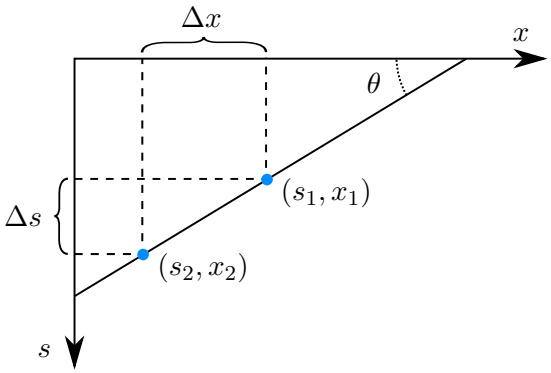


Figure 3.3: Schematic representation of an EPI-line in the sx-plane: if two adjacent views s_1 and s_2 are considered, the slope of the EPI-line is $\Delta s/\Delta x$ and its orientation equals to $\theta = \arctan(\Delta s/\Delta x)$.

3.4 Light Field Acquisition

A 4D light field can acquired in many ways. Two of the most common are plenoptic cameras and camera arrays. Additionally, a light field can be synthetically generated with computer graphics software. In this section, this acquisition methods will be briefly described.

3.4.1 Plenoptic Cameras

A plenoptic camera is a special type of camera which allows to capture a 4D light field by means of a matrix of micro-lenses, which measures the directional distribution of the light. This type of device, also known as light field camera, was firstly described by Lippmann [60] in 1908, who called the idea "integral photography". However, a plenoptic camera as we know it today is mainly based on the works of Adelson et al. [3] and Ng et al. [66]. A micro-lenses array is placed at the focal point of the main lens, between the lens and the image sensor (see Figure 3.4). In this way it is possible to acquire the angular information of the scene, due to the fact that the light is split by these micro-lenses. Each micro-lens projects a small sharp image of the the main lens aperture from different viewing angles, and covers as many pixels underneath it as possible, without overlapping with other sub-images. The main drawbacks of these sensors are the small baselines and the lack of resolution in favor of angular information. Two common light field cameras available on the market are the Raytrix and the Lytro

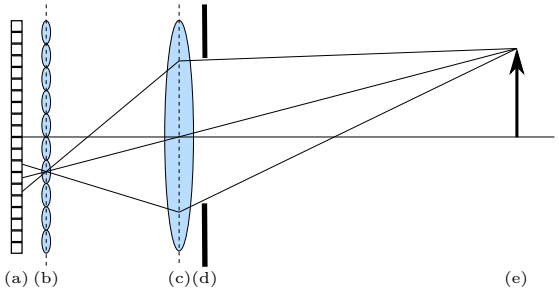


Figure 3.4: Representation of a plenoptic camera: camera sensor (a), micro-lens matrix (b), main lens (c), camera aperture (d), and the acquired object (e).

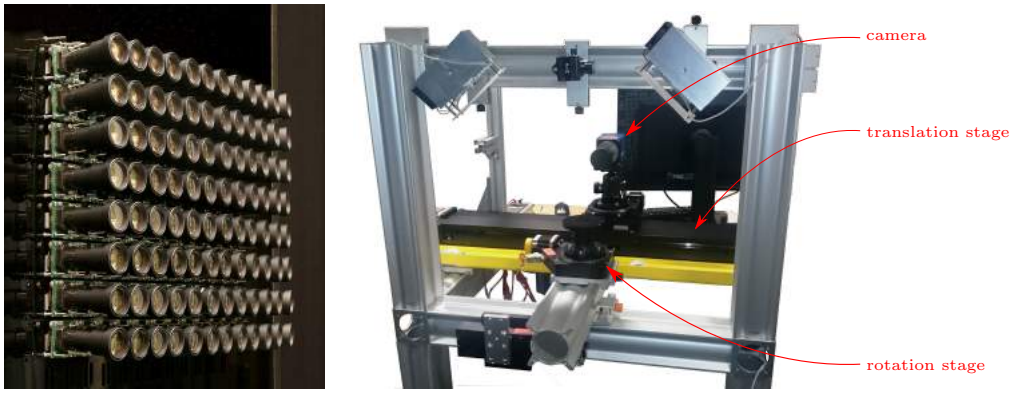
(a) (b)

Figure 3.5: Two commercial plenoptic cameras: the Raytrix [74] (a), and the Lytro [61] camera (b).

cameras, showed in Figure 3.5.

3.4.2 Camera Arrays and Gantries

Looking back at Figure 3.1, the sampling of the Π-plane can be achieved by acquiring a set of images at different positions. The easiest way to do this is by means of an array of standard cameras, or a camera gantry. Compared to a plenoptic camera, this setups have the advantage that the acquired images have a much higher resolution, and any commercially available camera can be used. Wilburn et al. [97] proposed a camera array, showed in Figure 3.6 (a), which captures dynamic scenes from different viewpoints. One of the disadvantages of such configurations is that they are difficult and expensive to build. Moreover, a highly precise calibration is fundamental in order to correct alignment errors, and project all the images onto a reference plane [88]. All



(a) (b)

Figure 3.6: Two systems for acquiring a light field: the multi-camera array from Wilburn et al. [97] (a), and the camera mounted on a translation stage built in our laboratory (b).

these issues can be avoided by using a single camera mounted on a translation stage (or vice versa a fixed camera recording an object mounted on a stage). An example of this acquisition system is showed in Figure 3.6 (b). With the translation stage the placement problem of the camera arrays is solved thanks to the precision of the stage itself, which in some cases can provide baselines down to $0.1\,\mu m$. Moreover, only one camera is needed, making the setup cheaper and eliminating issues related to the small variations between the array's sensors. The obvious drawback of a moving camera is that only static scenes can be acquired.

3.4.3 Synthetic Light Fields

In order to test and evaluate the quality of our algorithms, we will also use synthetic light fields. These are light field sequences generated with Blender [14], an open source 3D computer graphics software. Blender allows to acquire synthetic scenes with unlimited, perfectly aligned cameras, which can be placed anywhere in the 3D world space. Each generated image is completely undistorted, and comes with the corresponding ground truth depth, a very important information for the evaluation of the 3D reconstructions. Blender also allows to simulate material properties of the objects, such as Lambertian and specular surfaces. All these characteristics make Blender the perfect tool for fast and inexpensive dataset generation, as well as rapid testing of new

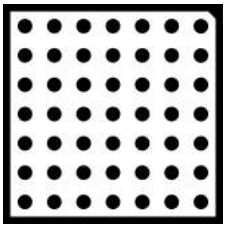


Figure 3.7: The Halcon calibration plate composed of 49 black dots on a white background used for the internal camera calibration.

algorithms.

3.5 Capture and Calibration

To capture our light fields we chose the approach of a single camera mounted on a translation stage. Figure 3.6 (b) shows the acquisition setup, composed of a Owis Limes 170 [68] high-precision linear stage, and a pco.edge 5.5 [71] USB 3.0 camera with a resolution of $2560 \times 2160\,px$ and pixel pitch $6.5\,\mu m/px$. Additionally, a high precision rotation stage (Owis DTM 130N [69]) is used to rotate the target objects 360° in front of the camera. With this setup we will acquire and analyze light fields generated from two different types of motion: linear and circular. Linear light fields are captured by linearly moving the camera in front of a fixed object, acquiring one image for each position of the translation stage. Circular light fields are acquired by keeping the camera fixed and using the rotation stage to rotate the object in front of it.

After the acquisition, each image has to be processed in order to remove the distortion introduced by the camera lens. To this end, we acquired images of the calibration plate showed in Figure 3.7, and then used the software Halcon [43] to estimate the intrinsic camera parameters and correct the images.

3.6 Local Orientation Estimation

A common way to estimate orientations in images is by means of the *structure tensor* (ST), which was firstly used for this purpose in the work of Bigun and Granlund [10]. This section follows the chapter "Directions in 2D" from the book of Bigun [9].

Let $f(\mathbf{r})$, with $\mathbf{r} = (x, y)^\top$, represent an image with some oriented textures, while the unit vector $\mathbf{n}$ represents the direction of these textures in the image (see Figure 3.8 (a)). The function f is called a *linearly symmetric image* if its isocurves have a common direction, i.e. there exists a scalar function of one variable g such that

$$f(x, y) = g\left(\mathbf{n}^\top \mathbf{r}\right) = g\left(n_x x + n_y y\right). \tag{3.6}$$

In this case, isocurves are parallel lines of constant intensity. In an EPI, isocurves correspond to lines, whose orientation is related to the disparity through Equation 3.5. In general, an EPI is not linearly symmetric, since it does not have constant disparity. However, an approximation to a linearly symmetric image can be obtained by considering a sufficiently small local neighbourhood.

From the previous definition, it derives that a linearly symmetric image can be generated solely from a 1D function $g(t)$ and a direction $\mathbf{n}$. The magnitude of the Fourier transform $|F(\boldsymbol{\omega})|$ of such an image is confined to a line through the origin having direction $\mathbf{n}$. Along this line, $|F(\boldsymbol{\omega})|$ is proportional to the 1D Fourier transform of the signal $g(t)$. Figure 3.8 shows a synthetic EPI of constant disparity and the magnitude of its Fourier transform. If the magnitude is zero, than the complex values be zero as well. The same can be said for the power spectrum $|F(\boldsymbol{\omega})|^2$, which will be used instead of the magnitude. The estimation of the direction of $\mathbf{n}$ is performed by fitting an axis to the power spectrum of the image f in the total least squares sense (see Figure 3.9). This is equivalent to finding an axis of direction $\mathbf{n}$ that minimizes the error function

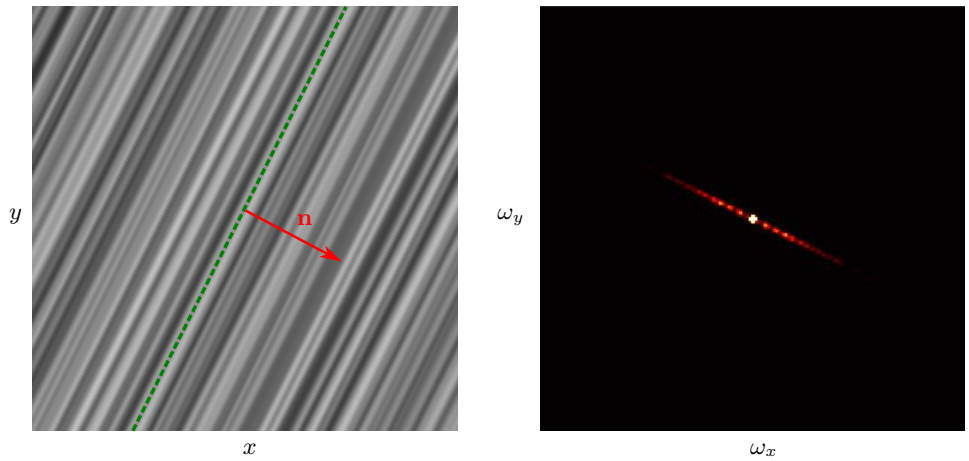
$$e(\mathbf{n}) = \int d^2(\boldsymbol{\omega}, \mathbf{n}) \, |F(\boldsymbol{\omega})|^2 \, d\boldsymbol{\omega}, \tag{3.7}$$

where $d(\boldsymbol{\omega}, \mathbf{n})$ is the shortest distance of the point $\boldsymbol{\omega}$ to the axis $\mathbf{n}$, i.e. the norm of $\mathbf{d}$, as shown in Figure 3.9. By rewriting the distance of points from the axis in quadratic form it derives

$$\begin{aligned} d^2(\boldsymbol{\omega}, \mathbf{n}) &= \mathbf{n}^\top \left(\mathbf{I}\boldsymbol{\omega}^\top \boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\omega}^\top\right) \mathbf{n} = \\ &= \mathbf{n}^\top \left[\begin{pmatrix} \omega_x^2 + \omega_y^2 & 0 \\ 0 & \omega_x^2 + \omega_y^2 \end{pmatrix} - \begin{pmatrix} \omega_x^2 & \omega_x \omega_y \\ \omega_x \omega_y & \omega_y^2 \end{pmatrix} \right] \mathbf{n}. \end{aligned} \tag{3.8}$$

For notational convenience, Equation 3.7 can be rewritten as

$$e(\mathbf{n}) = \mathbf{n}^\top \mathbf{J} \mathbf{n} = \mathbf{n}^\top \left(\mathbf{I} \cdot \mathrm{Trace}(\mathbf{S}) - \mathbf{S}\right) \mathbf{n}, \tag{3.9}$$



(a) (b)

Figure 3.8: The 2D image $f(x, y) = g(\mathbf{k}^\top \mathbf{r})$, which represents an EPI of constant disparity (a). The orientation vector $\mathbf{n}$ was chosen in order to set the disparity of the lines to $0.5\,px$. The subfigure (b) shows the magnitude $|F(\boldsymbol{\omega})|$ of the Fourier transform of $f(x, y)$, where $\boldsymbol{\omega} = (\omega_x, \omega_y)$. Here the line along which $|F(\boldsymbol{\omega})|$ is concentrated has the direction of $\mathbf{n}$.

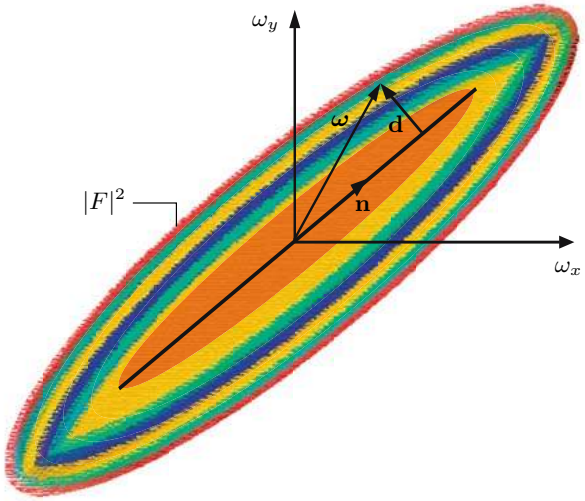


Figure 3.9: Visual representation of the power spectrum $|F(\boldsymbol{\omega})|^2$ of an image. Finding the direction of linear symmetry, i.e. the vector $\mathbf{n}$, equals to fitting an axis to $|F(\boldsymbol{\omega})|^2$ in the total least squares sense. The vector $\mathbf{d}$ is the distance vector of a point $\boldsymbol{\omega}$ from this axis and is orthogonal to $\mathbf{n}$. Source: [9].

where $\mathbf{I}$ is the *identity matrix*, $\mathbf{J}$ is the *inertia tensor*, and $\mathbf{S}$ is the structure tensor of the image f, defined as

$$\mathbf{S} = \int \boldsymbol{\omega}\boldsymbol{\omega}^\top \left|F\left(\boldsymbol{\omega}\right)\right|^2 \, d\boldsymbol{\omega} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix}, \tag{3.10}$$

with

$$\mathbf{S}_{ij} = \int \omega_i \omega_j |F\left(\boldsymbol{\omega}\right)|^2 \, d\boldsymbol{\omega}. \tag{3.11}$$

The matrix $\mathbf{S}$ is defined in the frequency domain. Therefore, when computing the direction in all the local patches of an image, one would have to perform the Fourier transform multiple times. However, by means of the Parseval's theorem and the differentiation property of the Fourier transform, Equation 3.10 can be reformulated in the spatial domain as

$$\mathbf{S}_{ij} = \frac{1}{4\pi^2} \int \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \, d\mathbf{x}, \quad \text{with} \quad i, j : 1, 2 \tag{3.12}$$

where $x_1 = x$, $x_2 = y$, $d\mathbf{x} = dxdy$, and the integral is a double integral over the 2D plane. The correspondent matrix form is

$$\mathbf{S} = \frac{1}{4\pi^2} \int \nabla f \; \nabla^\top f \, d\mathbf{x}. \tag{3.13}$$

Aside from the $1/4\pi^2$ factor, the introduction of a window function $w(\mathbf{r})$ to restrict the computation to a local neighbourhood leads to the same definition given by Jähne [48]:

$$\mathbf{S} = \int w\left(\mathbf{r} - \mathbf{r}'\right) \left[\nabla f\left(\mathbf{r}'\right) \; \nabla^\top f\left(\mathbf{r}'\right)\right] \, d\mathbf{r}'. \tag{3.14}$$

This equation is derived by maximizing the squared scalar product between the direction vector $\mathbf{n}$ and the gradient vector ∇f:

$$(\nabla^\top f \cdot \mathbf{n})^2 = |\nabla f|^2 \cos^2(\sphericalangle(\nabla f, \mathbf{n})), \tag{3.15}$$

which reaches the maximum when the two vectors are parallel or antiparallel.

The structure tensor associates a 2×2 symmetric matrix to every point in an image, so that each component S_{ij} is of the same size of the image. From now the notation $\mathbf{S}$ will be used to indicate the matrix

$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}. \tag{3.16}$$

In order to find the optimal orientation, the quadratic form $\mathbf{n}^\top \mathbf{S}\mathbf{n}$ has to be maximized. This can be done by finding the eigenvector of $\mathbf{S}$ corresponding to the highest eigenvalue. As shown by Jähne [48], if we call λ_1 and λ_2 the eigenvalues of $\mathbf{S}$ and assume, without loss of generality, that $\lambda_1 > \lambda_2$, three cases arise:

- when $\lambda_1 = 0$, $\lambda_2 = 0$, the neighborhood is constant;
- when $\lambda_1 > 0$, $\lambda_2 > 0$, the neighborhood changes in all directions;
- when $\lambda_1 > 0$, $\lambda_2 = 0$, the neighborhood is linearly symmetric.

The eigenvalues give a measure of the scatter of the spectral energy of f and correspond to the inertia about the axes defined by the respective eigenvectors. In the last case (i.e. $\lambda_1 > 0$, $\lambda_2 = 0$), we have an ideal local orientation. In fact, the power spectrum of f is concentrated to a central line, and the error $e(\mathbf{n}_1)$ equals to 0, where $\mathbf{n}_1$ is the eigenvector of λ_1.

In general, it is useful to define a measure to quantify how well an image patch approximates linear symmetry. Jähne [48] defines the *coherence* as

$$c = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{\sqrt{(S_{11} - S_{22})^2 + 4S_{12}^2}}{S_{11} + S_{22}}, \tag{3.17}$$

which varies between 0 for isotropic structures and 1 in presence of ideal orientations.

The disparity can be computed by determining the local orientation angle θ. Thus we identify the real orthogonal matrix that diagonalizes $\mathbf{S}$ by solving

$$\mathbf{R}^{-1}\mathbf{S}\mathbf{R} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \quad \text{with} \quad \mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3.18}$$

By setting the off-diagonal elements of the left side to zero we end up with the formula for the angle

$$\theta = \frac{1}{2}\arctan\left(\frac{2S_{12}}{S_{11} - S_{22}}\right), \tag{3.19}$$

where $0 < \theta < \pi$.

In an EPI, the angle θ is formed by $\mathbf{n}_1$ with the x-axis and it is equivalent to the orientation of the corresponding EPI-line (see Figure 3.3). Hence, by substituting Equation 3.19 in Equation 3.5, we obtain the disparity

$$d_{y^*,t^*} = \tan\theta = \tan\left(\frac{1}{2}\arctan\left(\frac{2S_{12}}{S_{11} - S_{22}}\right)\right). \tag{3.20}$$

3.6.1 Classic Structure Tensor

The implementation of the structure tensor for discrete images generally consists of four steps: an initial smoothing to reduce noise, the gradients computation, the structure tensor components computation, and the smoothing of these components.

In the first step, a symmetric Gaussian filter $\mathcal{G}_\rho$ is used, which is obtained by discretizing the function

$$G_\rho(x, y) = \frac{1}{2\pi\rho^2} \mathrm{e}^{-\frac{x^2+y^2}{2\rho^2}}. \tag{3.21}$$

A Gaussian has infinite support, but since about 99.7% of the area under a 1D Gaussian is contained in a region $[\mu - 3\rho, \mu + 3\rho]$ (where μ is the mean which we set to 0), we can neglect the coefficients outside this region. The kernel radius of the filter is then $r = \lceil 3\rho \rceil$, yielding a filter kernel of a size $[\delta \times \delta]$, with $\delta = 2r + 1 = 2\lceil 3\rho \rceil + 1$.

The gradients can be computed with standard derivative filters, applied horizontally and vertically. Two of the most common are the 3×3 Sobel and Scharr [80] filters, which compute the derivative in one direction and apply smoothing in the perpendicular one. The Scharr operator is better suited to estimate the orientation of lines, since it optimizes the rotational symmetry. The filter masks for the horizontal derivative are

$$\text{Sobel:} \quad \mathcal{H}_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad \text{Scharr:} \quad \mathcal{H}_x = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}, \tag{3.22}$$

and in the vertical direction $\mathcal{H}_y = \mathcal{H}_x^T$. The derivative of image I is thus defined as

$$D_{x_i, \rho} = I * \mathcal{G}_\rho * \mathcal{H}_{x_i}, \quad \text{with} \quad i = 1, 2 \tag{3.23}$$

where $x_1 = x$, $x_2 = y$, and $*$ denotes the convolution. In accordance with Wanner and Goldlücke [94], we call ρ the *inner scale* of the structure tensor.

Alternatively, the smoothing and differentiation steps can be combined with the derivative of Gaussian (or Gaussian gradient) filter $\partial\mathcal{G}_{x_i, \rho}$ which is obtained by sampling the function

$$\frac{\partial G_\rho(x_1, x_2)}{\partial x_i} = -\frac{x_i}{2\pi\rho^4} \mathrm{e}^{-\frac{x_1^2+x_2^2}{2\rho^2}}. \tag{3.24}$$

In this case we will have

$$D_{x_i, \rho} = I * \partial\mathcal{G}_{x_i, \rho}. \tag{3.25}$$

The window function w of Equation 3.14 is implemented by convolving the three structure tensor components with a Gaussian filter $\mathcal{G}_\sigma$ obtained from Equation 3.21. Wanner and Goldlücke [94] define σ the *outer scale*. The structure tensor is then defined as

$$S_{\rho,\sigma}(I) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} = \begin{pmatrix} \mathcal{G}_\sigma * (D_{x,\rho} \cdot D_{x,\rho}) & \mathcal{G}_\sigma * (D_{x,\rho} \cdot D_{y,\rho}) \\ \mathcal{G}_\sigma * (D_{x,\rho} \cdot D_{y,\rho}) & \mathcal{G}_\sigma * (D_{y,\rho} \cdot D_{y,\rho}) \end{pmatrix}, \tag{3.26}$$

where $\cdot$ identifies the pointwise multiplication. Eventually, the disparity can be obtained through Equation 3.20.

3.6.2 Modified Structure Tensor

Another local orientation estimator is the so called *modified structure tensor* proposed by Diebold [30]. Even though this method was originally developed to process heterogeneous light fields (e.g. image sequences with changing properties between the captured frames), it is also claimed to yield better results for standard light fields. Differently from the classic tensor, the modified one replaces the inner Gaussian smoothing by differentiating the EPI in the x-direction. To this aim, Diebold used a 1D derivative filter with kernel

$$\mathcal{D}_x = \frac{1}{2}[1 \quad 0 \quad -1]. \tag{3.27}$$

The gradient components are then computed as

$$D_{x_i} = I * \mathcal{D}_x * \mathcal{H}_{x_i}. \tag{3.28}$$

The horizontal differentiation allows to be more robust to intensity changes along feature paths, which can result, for example, from changes in illumination across views.

3.6.3 2.5D Structure Tensor

In contrast to the classic and the modified approach, which are applied to a single 2D EPI, a 2.5D variant of the structure tensor, which considers also the vertical direction of the image volume, was proposed by Diebold [28]. This method has an additional 1D smoothing along the y-coordinate, i.e. in the direction perpendicular to the EPIs. This is implemented by convolving, along the y-direction, the structure tensor components $\mathbf{S}_{ij}$ for each view with a 1D Gaussian kernel of standard deviation σ (outer scale). The 2.5D structure tensor extends the smoothing range from the 2D EPI to the 3D light field

volume, including also the local image information of neighbouring EPIs. This leads to more support for the local orientation computation. Therefore, smaller kernels can be used to achieve results with similar precision to those of the 2D structure tensor. On the other hand, increasing the support reduces the precision at depth discontinuities.

3.6.4 Refocusing

In general, an EPI-line has to be continuous in order to be correctly estimated by the local structure tensor. If the disparity between neighbouring views is too large, a line degenerates into a sequence of disconnected segments. To avoid this, a *refocusing* step is needed to ensure that EPI-lines are in the disparity range $\pm 1\,px$. As described by Diebold and Goldlücke [29] and Wanner [92], EPI-rows are shifted in the opposite direction with respect to the center view (see Figure 3.10). In case of datasets with a total disparity range larger than 2 pixels, the refocus step has to be repeated iteratively. Eventually, the results of different refocus levels are merged by choosing, for each pixel, the disparity with the highest coherence.

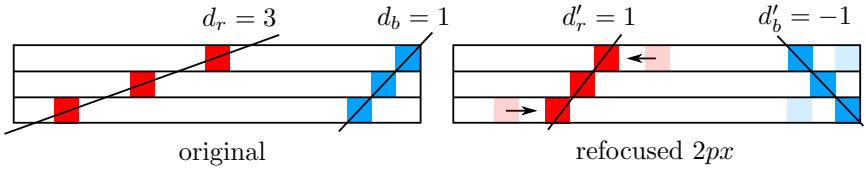


Figure 3.10: The original EPI has disparities in the range $[1\,px, 3\,px]$. It can be refocused with a shift of $2\,px$ in order to make disparities lie in the range $[-1\,px, 1\,px]$. In the refocused EPI, the red line with an original disparity $d_r = 3\,px$ will have a disparity $d'_r = 1\,px$, and the blue line will go from $d_b = 1\,px$ to $d'_b = -1\,px$. Source: [92].

Chapter 4

Depth Reconstruction from Linear Light Fields

This chapter presents a new method for estimating high quality depth maps from linear light fields. The approach exploits a coarse slope map generated by computing the local structure tensor of the EPI, together with a binary edge map of the EPI, and extracts feature paths, also called trajectories, by using a modified version of the Hough transform. The result is a set of highly accurate depth maps of the target scene from all the viewpoints. Since the Hough transform uses binarized EPI images to retrieve trajectories, it is possible to get rid of the Lambertian hypothesis and process even datasets with intensity changes along the EPI-lines. In fact, if a feature path is only partially visible, or its intensity saturates because of a specular reflection, the Hough transform can still recover the full line. Differently from the structure tensor, which provides local orientation estimation by only using a portion of the line, with the Hough transform all the points lying on the line contribute to the detection of the line itself. Therefore, the proposed approach can be considered a *semi-global* method. In the following, Section 4.1 explains the Hough transform, focusing on the specific case of line detection. The application of this method to linear EPIs, and the proposed algorithm, are presented in Section 4.2. Eventually, both local structure tensor and the Hough transform based methods are evaluated for synthetic and real datasets in Sections 4.3 and 4.4, respectively. Part of the work presented in this chapter has already been published by G. Manfredi in his Master of Science thesis [62].

4.1 Hough Transform for Line Detection

The Hough transform [46] is an elegant method for estimating parametrized line segments in an image. This approach can locate regular curves such as straight lines, circles, parabolas, ellipses, etc. in an image. In practice, Hough transform is applied on a binary image, which we obtain with a Canny edge detector [20]. The simplest case is straight line detection: let (x_i, y_i) be an image point, all the lines passing through this point must satisfy the equation

$$y_i = mx_i + c, \tag{4.1}$$

where m and c are the slope and the offset of the line, respectively. The main idea behind the Hough transform is to consider a straight line not as a collection of image points (x, y), but as parameters (m, c). These define a *parameter space* of coordinates (m, c) where an image point maps to a line. All the points lying on a straight line in the image space will be mapped onto lines in the parameter space. All these lines intersect in a point which uniquely defines the parameters of the line in the image. Therefore, image lines can be detected by simply considering points in the mc-plane where enough lines intersect. The drawback of this parametrization is that the slope m of a line approaches infinite for vertical lines. Therefore, in practical applications a line is expressed in polar coordinates as

$$\rho = x\cos\theta + y\sin\theta, \tag{4.2}$$

where ρ is the perpendicular distance from the origin to the closest point on the line (e.g. the vector $\mathbf{d}$ in Figure 4.1), and θ is the angle between the x-axis and the $\mathbf{d}$ vector. In general this angle is comprised in a range $|\theta_i| \leq \theta_{\max}$. With this parametrization, each edge point in the image maps to a sinusoidal curve in the new (ρ, θ) parameter space, the so called *Hough space*.

In order to identify lines, the Hough space is discretized in so called *cells* and initially populated with zeros. Then each edge point *votes*, which means it increments by 1 the cell having coordinates (ρ_i, θ_i) with $|\theta_i| \leq \theta_{\max}$, where ρ_i is the discrete ρ coordinate whose value is closest to the one computed with Equation 4.2. Once voting is complete, cells whose values are local maxima or peaks define the parameters for the lines in the image. Eventually, the disparity of an EPI is computed through Equation 3.20 by using the orientation θ of a line instead of the structure tensor orientation. Line detection in

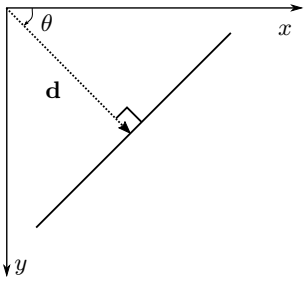


Figure 4.1: Parametrization of a line in polar coordinates: if $\mathbf{d}$ is the shortest vector connecting the origin to the line (to which it is perpendicular), then θ is the angle it makes with the x-axis and $\rho = \|\mathbf{d}\|$ is its norm. The line can then be expressed through Equation 4.2.

EPIs by means of the Hough transform has the advantage of being robust against noise and unaffected by occlusions of the feature paths. Moreover, the Hough transform is tolerant to gaps in the edges, because even a partial line can be reconstructed if it has enough support.

4.2 Application to Linear Light Fields

Given its characteristics, a Hough transform based approach seems ideal to treat linear light fields. To do this, the first step consists in defining the parameter space (ρ_i, θ_i). For the same reasons explained in Section 3.6.4, only continuous EPI-lines, i.e. within the ± 1 disparity range, can be correctly estimated. Thus, the refocusing procedure of Section 3.6.4 has to be applied also in this case. That said, it will be set $\theta_{\max} = \arctan(1) = 45°$, so that $-\theta_{\max} \leq \theta \leq \theta_{\max}$. The sampling of θ is performed by taking the inverse of the tangent of linearly spaced disparity values. The *disparity resolution* can be computed from the height of the EPI, i.e. by the number of views N, and is defined as

$$\Delta d = \frac{1}{N-1}. \tag{4.3}$$

The discretization of ρ depends on the chosen sensor. Specifically, a sensor with resolution $N_x \times N_y$ pixels yields to $\rho \in [0, 1, ..., N_x]$ pixels.

The preliminary step to identify lines using the Hough transform is to generate a binary edge map of the EPI. Similarly to Criminisi et al. [23] a Canny edge detector is

applied. The gradient used by the detector is computed with a derivative of Gaussian filter $\partial \mathcal{G}_{x_i,\sigma}$ as defined by Equation 3.24, where σ is the *edge scale*.

The problem of determining the extension of a line, i.e. where it is actually visible, can be solved in various ways. A possible approach would be to find the intersection of lines using their equations. This would yield a high number of intersection points, defining as many line segments. Then either all the line segments, or all the intersection points, would need to be classified in order to determine which segments are visible or which points are actual intersections, respectively. In fact, a feature path is theoretically visible in the whole EPI (i.e. in all views), unless it is occluded at some point by another line with higher disparity, which corresponds to a closer feature.

In this work, we choose a different approach, which adapts a particular implementation of the Hough transform, the *Progressive Probabilistic Hough Transform* (PPHT) [37], to the characteristics of an EPI. The choice of the PPHT was mainly guided by speed considerations. In fact, for the standard Hough transform, Equation 4.2 has to be solved for every value of θ for every edge point. Therefore, the voting process can be a very costly operation, particularly considering that it has to be repeated for every EPI, i.e. as many times as the rows in a single view (vertical camera resolution), and even more if several refocusing steps are required. The advantage of the PPHT is that voting is restricted to a subset of all edge points: to detect a line only as many points have to vote as are required to bring the value of the corresponding accumulator cell above a threshold *thr*.

In the following sections, the general algorithm is described. A list of the parameters used can be found in Appendix A.1.

4.2.1 Outline

During the processing of an EPI, edge points vote in an order defined by a random probability distribution. Once a line has been detected, it is processed in two steps. In the *first step* the end points are determined based on the edge map: the longest segment is found which is either continuous or has gaps of a given maximum length. In the *second step*, all line points remove their votes (if any) from the accumulator and are deleted from the edge map, so that they will not vote again.

Although applying the original algorithm is possible, we decided to leverage the local orientation provided from the structure tensor. Therefore, we have three inputs: the edge map from the Canny filter, the local disparity, and the coherence map. The structure tensor disparity is used to:

(a) reduce the voting range of edge points

(b) control the deletion of points from the edge map in the second step

(c) determine the end points of lines in the disparity map

4.2.2 Voting Range Reduction

To speed up computation and remove noise in the Hough space, it is desirable to restrict the angle range, over which an edge point p_i votes, to a region around the structure tensor orientation θ_i'. In order to determine how large this region should be, it is possible to use the coherence (defined in Equation 3.17) which characterizes the quality of the structure tensor estimate. If the coherence c_i at that point is low, i.e. below a threshold c_{th}, p_i will vote over the whole angle range. On the other hand, as the coherence grows, the range can be decreased. In this way, for large coherences ($c_i \geq c_{th}$) the size of the search range is determined by a linear function of the coherence. More formally, a point p_i votes over $|\theta - \theta_i'| \leq \Delta\theta(c_i)$, with

$$\Delta\theta(c_i) = \Delta\theta_{\min} + (\theta_{\max} - \Delta\theta_{\min})\frac{c_i - 1}{c_{th} - 1}, \tag{4.4}$$

where $\Delta\theta_{\min}$ defines the minimum size of the angle range and $\theta_{\max} = 45°$. The comparison between the standard Hough space and the one with the structure tensor initialization is shown in Figure 4.2

4.2.3 Controlled Edge Points Deletion

A common situation is that background lines (lower disparity), which are occluded at some point, are detected before the foreground line marking the occlusion boundary (higher disparity). Since in the second step of the algorithm each detected line is deleted from the edge map, these background lines will also delete points from the boundary line, making its detection more difficult. In fact, if the accumulator threshold *thr* is not low enough, it might be the case that this line, which is of high importance for

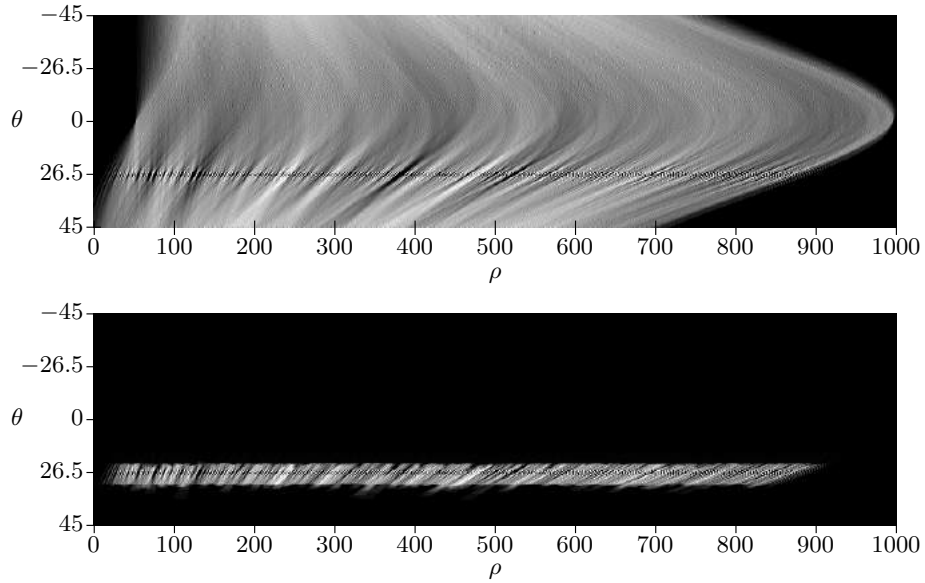
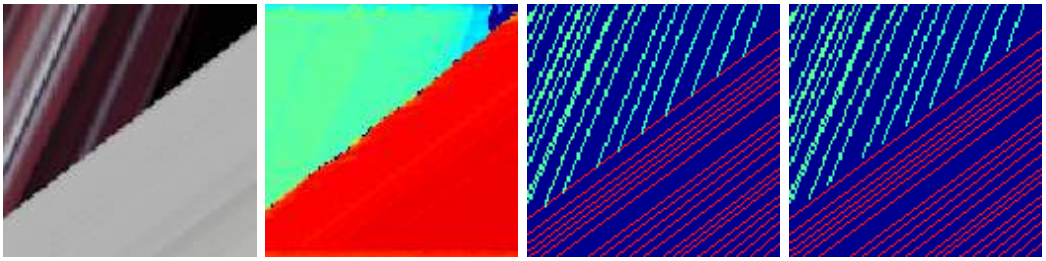


Figure 4.2: On the top, the accumulator representing the Hough space for a synthetic EPI of constant disparity $0.5\,px$. The domain of θ is $[-45°, 45°]$. As can be seen, image points map to sinusoidal curves in the Hough space, and these intersect for θ being equal to the orientation of lines in the image. As expected, maxima can be observed at $\theta \approx 26.57° \approx \arctan(0.5)$. The bottom figure shows the accumulator resulting by restricting the voting range to a region around the structure tensor orientation θ'_i. The size of the range depends on the coherence c_i of a point, which leads to the different length of the curves in the accumulator.



(a) (b) (c) (d)

Figure 4.3: Controlled edge point deletion: the EPI portion (a) shows a depth discontinuity. With the standard PPHT the line marking the occlusion boundary is overwritten by background lines (c). The local orientation estimation from the structure tensor (b) can be used to guide the deletion of the edge points and preserve the occlusion border (d). Note: the binary edge maps (c-d) are color coded to differentiate background from foreground lines.

subsequent processing of the disparity map, will not be detected. The optimal solution would be to first detect foreground lines. This could be implemented by using the local orientation estimates to define a custom probability distribution, where a point has a probability of being selected for the voting proportional to its disparity. The task of redefining a new custom probability distribution for every point has however proven to be prohibitively slow[1]. Our solution consists in deleting a point in the edge map only if the structure tensor's disparity is smaller than the detected line's disparity by a margin.

As an example, let us suppose that a low disparity line is found during voting. Let us further suppose that this line is occluded at some point and that, in the first step (end points detection), the detected end point of the line lies on the higher disparity line marking the occlusion boundary (if the edge map is good this should always be the case). In the second step (removal of the detected line from the edge map), we do not want to delete this point (as happens in Figure 4.3 (c)), since it in fact belongs to the boundary line, and not to the detected one. Therefore we check the disparity returned by the structure tensor at that point and establish that it is higher than the disparity of the detected line. In this case we do not delete the point from the edge map, so that

[1]We tested two implementations of the `discrete_distribution` class, as supplied by the C++ Standard Template Library (STL) [47] and by Boost [79].

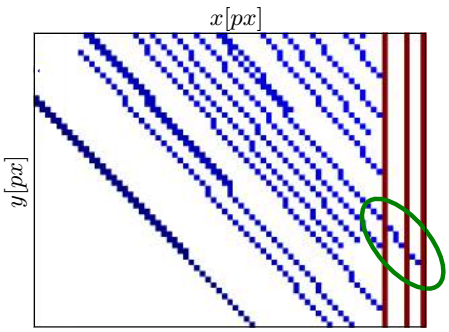


Figure 4.4: This figure shows what can happen if the standard PPHT is applied to an EPI. The leftmost red line (higher disparity) marks an occlusion boundary, and the region to its right belongs to a foreground object. This means that the blue lines (lower disparity) should stop at the boundary. However, if no constraint is enforced it can occur that some of these lines propagate too far. In the region inside the green circle, it is shown what happens if the maximum gap is set to 4 pixels, which is less than the distance between the parallel red lines (foreground): the foreground lines "support" the propagation of the blue line (background).

it will still cast its votes and/or be a "supporting point" of the foreground line marking the occlusion boundary, as shown in Figure 4.3 (d).

4.2.4 Controlled Line Propagation

The last feature is needed in cases where lower disparity lines are erroneously propagated over an occlusion boundary (i.e. in a foreground region) due to the sparsity of the EPI edge map. In fact, if the maximum line gap is larger than the distance of two neighbouring foreground lines, it can happen that these lines end up "supporting" the background line. In order to detect this scenario, line points for which the disparity of the line lies below the structure tensor's one by a given margin are counted, and if their number exceeds a threshold the line is marked as "suspect". This means that a line should be suspect if it penetrates a region of higher disparity, as shown in Figure 4.4, a situation that cannot happen in a real EPI. If this is the case, the line is saved for further processing (see Section 4.2.5), otherwise it is drawn directly in the disparity map during the second step.

4.2.5 Handling of suspect lines

The algorithm terminates when the edge map is empty, either because all points have voted or because they have been removed, as belonging to a detected line. At this stage, the disparity map already contains all the non-suspect lines, while suspect lines are saved in a list. This list is sorted by decreasing disparity, i.e. from foreground to background lines, and lines are drawn in the disparity map from a given point until a line with a higher disparity, which determines the occlusion point, is met. The main problem is finding the point from which a line has to be propagated. To this end, the difference between line's and structure tensor's disparities is checked at the two end points of a line, to determine if the end points yield the correct disparity. Thus, three cases arise:

1. Both points are correct: the line is propagated from both points.

2. Only one point is correct: the line is propagated from this point.

3. None of the points is correct: the algorithm walks the line until a point is met where local orientation and line orientation are similar. The line is then propagated from this point in both directions.

4.2.6 Line score

Similarly to the local structure tensor, also the proposed Hough transform approach needs a quality measure to characterize the reliability of the detected lines. We therefore define the *line score* as

$$\text{line score} = \frac{1}{2}\left(\frac{\text{supported points}}{\text{line length}} + \frac{\text{line length}}{N}\right), \tag{4.5}$$

where *supported points* is the number of line points in the edge map, and N is the number of views, i.e. the EPI height. The line score can be used to merge disparity maps belonging to different refocusing steps, and to filter out lines with low scores. As for the coherence, its value lies in the range of $[0, 1]$. However, the score above which lines can be considered correct is generally lower, and will be further analyzed in Section 4.3.

Figure 4.5: Synthetic EPIs for disparity values of -1, -0.5, 0, 0.5, 1 pixel (from left to right).

4.3 Synthetic EPIs Evaluation

In this section, an extensive evaluation of the proposed algorithm is presented. Synthetic EPIs of constant disparity are used to compare our method with local structure tensor approaches.

4.3.1 Synthetic EPIs Generation

To assess the performance of the algorithms, we generate synthetic EPIs of constant disparity in the range $\pm 1\,px$, linearly spaced by $0.01\,px$. Each EPI has a height of 101 pixels, representing 101 views. A synthetic EPI of disparity $d = 0$ is obtained by creating an image where each column has a random constant intensity value between 0 and 1. The image is then convolved with a one-dimensional Gaussian filter to mimic the appearance of a real EPI, where lines have different widths and transitions are smooth. This image is used to generate EPIs of arbitrary disparity by simply shifting each row with sub-pixel accuracy (using linear interpolation), where the shift is determined by the row index. This is the same procedure applied for the refocusing described in Section 3.6.4. It is important to notice that the slope of a line is determined by a shift of the pixels between the views, i.e. by the disparity. This means that the orientation can be detected reliably only for lines having disparities in the range $[-1\,px, +1\,px]$. Figure 4.5 shows examples of EPIs for five different disparity values. The synthetic EPIs are processed with a given algorithm, and the disparity values of the center row are compared with the ground true value to determine the estimation error. We note that in our implementation disparity values outside the range $[-1\,px,\, 1\,px]$ are rejected. For each disparity value, 50 EPIs were created to make the results statistically reliable, giving a total of $N = 201 \times 50$ EPIs. Additionally, to better evaluate the robustness of

the algorithms, zero-mean Gaussian white noise of variance σ_n^2 was added to the EPIs (the noise variance refers to intensities in the range $[0, 1]$).

4.3.2 Bias, Precision and Accuracy

The performance of a disparity estimator can be quantified through three measures: bias, precision and accuracy [6]. The bias corresponds to the systematic error of an estimator and is equal to the mean error. On the other hand, the precision represents the statistical (random) error and can be measured by the variance (or the standard deviation) of the estimates, meaning that it does not depend on the true value. The accuracy of an estimator measures both its bias and its precision, as well as the closeness of the estimates to the true value. A common way to quantify accuracy is through the *mean square error* (MSE), as it can be computed as the sum of the squared bias and the variance. More formally, the MSE for a disparity estimator d with respect to the ground true disparity gt is defined as

$$MSE(d) = (Bias(d, gt))^2 + Var(d) = (\mathbb{E}(d - gt))^2 + \mathbb{E}[(d - \mathbb{E}(d))^2], \tag{4.6}$$

where $\mathbb{E}$ denotes the expected value. In practice, for a disparity map, the MSE is computed as

$$MSE = \mathbb{E}[(d - gt)^2] = \frac{1}{P} \sum_p (d_p - gt_p)^2, \tag{4.7}$$

where d_p is the estimated disparity at pixel p, gt_p is the ground truth disparity, and P is the total number of pixels for which both a disparity estimate and a ground truth value[1] exist.

4.3.3 Results

In order to provide a graphical representation of the achievable accuracy, we first show the *box-and-whisker* diagrams of the errors. An example of this type of diagram is shown in Figure 4.6. The central red line of the box, which corresponds to the second quartile q_2, represents the median value of the errors, giving a rough idea of the bias (as opposed to the mean error, it is less affected by outliers). The extension of the blue rectangles corresponds to the *interquartile range* (IQR), i.e. the difference between the

[1]For synthetic EPIs a ground truth value always exists. This specification is needed later on in Section 4.4 when discussing actual light field datasets (synthetic and real).

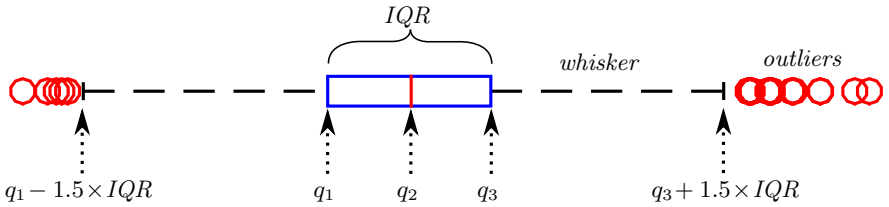
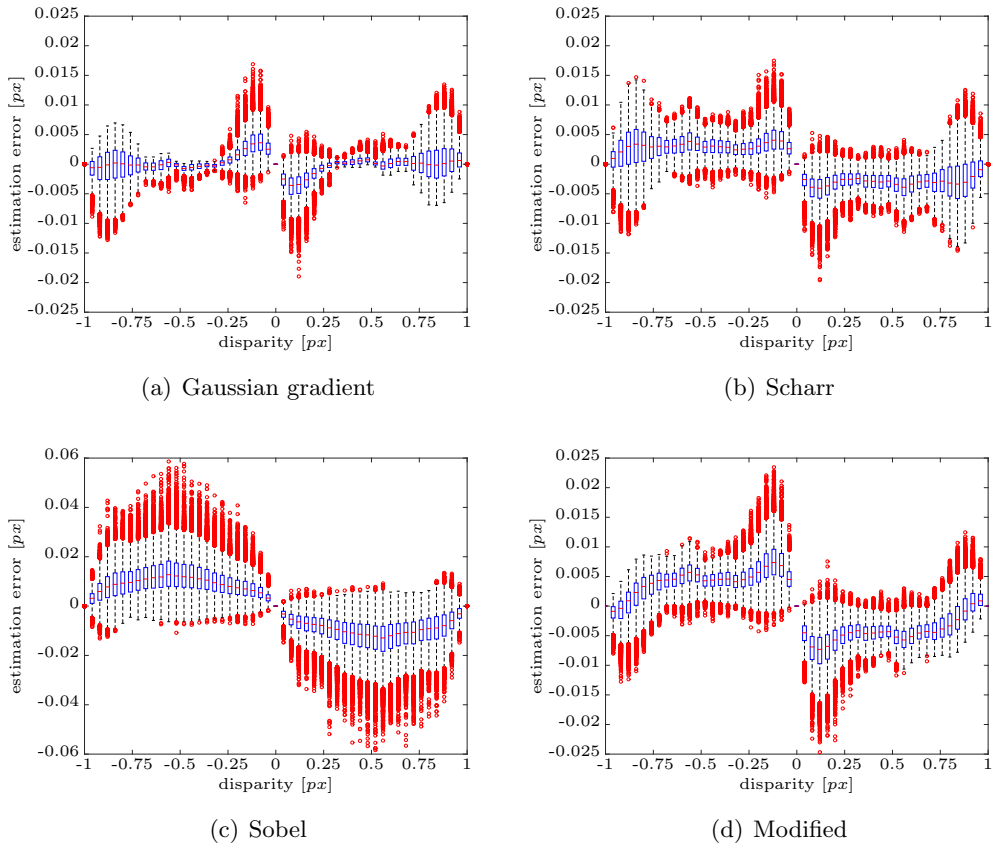


Figure 4.6: Sample box from a box-and-whisker diagram rotated clockwise by 90°. q_1 is the first quartile (25th percentile), q_2 the second (median, 50th percentile), and q_3 the third (75th percentile). The region between q_1 and q_3 is the interquartile range (IQR). The total extension of the whiskers includes values in the range $[q_1 - 1.5 \times IQR,\ q_3 + 1.5 \times IQR]$. Red circles represent outliers.

first quartile q_1 (lower edge of the rectangle) and the third quartile q_3 (upper edge). The quartiles q_1, q_2, and q_3 mark the 25^{th}, 50^{th}, and 75^{th} percentile, respectively. This means that 50% of the errors reside within the IQR. The IQR also gives a rough idea of the precision of the evaluated method. The black dotted lines extending outside this area are called *whiskers* and contain error data in the range $[q_1 - 1.5 \times IQR,\ q_3 + 1.5 \times IQR]$. If the data were normally distributed, the whiskers would correspond to approximately $\pm 2.7\sigma$ and cover about 99.3% of the data. Values outside this range are considered as outliers and marked with red circles.

In this section, the proposed Hough transform approach is compared with the structure tensor, which is implemented using three different derivative filters: Gaussian gradient ($[5 \times 5]$), Scharr ($[3 \times 3]$), and Sobel ($[5 \times 5]$). The Gaussian gradient filter is also used in the Hough transform approach to compute the structure tensor. Moreover, we include in the evaluation also the modified structure tensor method from Diebold [30]. Since we are dealing with EPIs of constant disparity, increasing the outer scale σ, and therefore the window over which the structure tensor is computed, would give increasingly better results, as long as border artifacts are neglected. However, using a too large window is unrealistic, since in practice an EPI contains different orientations, and averaging over these would yield to wrong results. In general, the outer scale has to be chosen based on the size of the features we want to detect. Based on the experiments of Wanner [92], an optimal inner scale for the gradients ρ of 0.75 is used, whereas the outer scale σ is set to 1.5. The resulting box-and-whisker diagrams for the structure tensor estimations are shown in Figure 4.7. From these plots it can be already seen



(a) Gaussian gradient (b) Scharr

(c) Sobel (d) Modified

Figure 4.7: Box plots of disparity error for classic structure tensor with 3 derivative filters: (a) Gaussian gradient [5×5], (b) Scharr [3×3], and (c) Sobel [5×5], for $\rho = 0.75$ and $\sigma = 1.5$. Box plot of disparity error for modified structure tensor with Scharr derivative filter and $\sigma = 1.5$ (d). To simplify visualization, boxes are shown each 4 disparity values.

that the Sobel filter is the least accurate estimator, whereas the remaining three have similar performances. The errors produced using the Hough transform approach are shown in Figure 4.8. Here it can be observed the effect of the angle quantization in the Hough space. In fact, a disparity value is either estimated exactly (see the median values centered at zero), or with an error a multiple of the quantization step of θ (i.e. the disparity resolution). In this specific case, the EPIs have a height of 101 pixels, which leads to a disparity resolution of $0.01\,px$.

Figure 4.9 (a) shows the mean errors for each disparity value of the noiseless EPIs. This plot gives a measure of the estimation's biases for all the different methods. It can

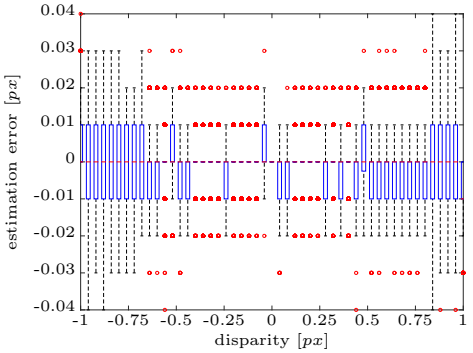


Figure 4.8: Box plot of disparity error for Hough transform. Edge scale is 1.5. The local orientation is computed via classic structure tensor with Gaussian gradient ($\rho = 0.75$, $\sigma = 1.5$). Other parameters: accumulator threshold 40, minimum line length 20, maximum gap 3, $c_{th} = 0.9$ (see Appendix A.1 for a description of the parameters).

be observed that all the curves have a peculiar behavior around the disparities $0\,px$, $\pm 0.5\,px$ and $\pm 1\,px$. This is due to the fact that the EPI-lines are not continuous but quantized on a discrete pixel grid. Therefore, lines with a disparity of $0\,px$ and $\pm 1\,px$ can be exactly represented on the grid, i.e. the quantization error is zero, and lines at $\pm 0.5\,px$ can be approximated relatively well. As a consequence, all the methods can better detect these orientations. On the other hand, the quantization error is maximized around the disparities $0\,px$, and $\pm 1\,px$, leading to larger errors. Aside from this phenomenon, the results of the classic structure tensor with Gaussian gradient and of the Hough transform exhibit the smallest biases. Regarding the other methods, we can observe how the Sobel filter gives the worst results, whereas the Scharr operator, which as previously mentioned optimizes rotational symmetry, performs clearly better. In addition, Figure 4.9 (b) shows the resulting standard deviations of the estimates, which give a measure of each method's precision. We can observe that the Hough transform disparity estimates are roughly as unbiased as the classic structure tensor, but less precise. This is due to the fact that, differently from the structure tensor, which produces continuous orientation estimates, the Hough transform discretizes the disparity space with a certain resolution. The variance of the estimates directly depends on the disparity resolution defined by Equation 4.3. For 101 pixels EPIs this resolution is $0.01\,px$. Figures 4.9 (c) (d) show the mean error and standard deviation for the noisy EPIs case. Here, a zero-mean Gaussian white noise of variance $\sigma_n^2 = 0.01$ was added to

each EPI (an example of noised EPI is shown in Figure 4.10). The results show that the proposed Hough transform method dramatically outperforms the local structure tensor approaches.

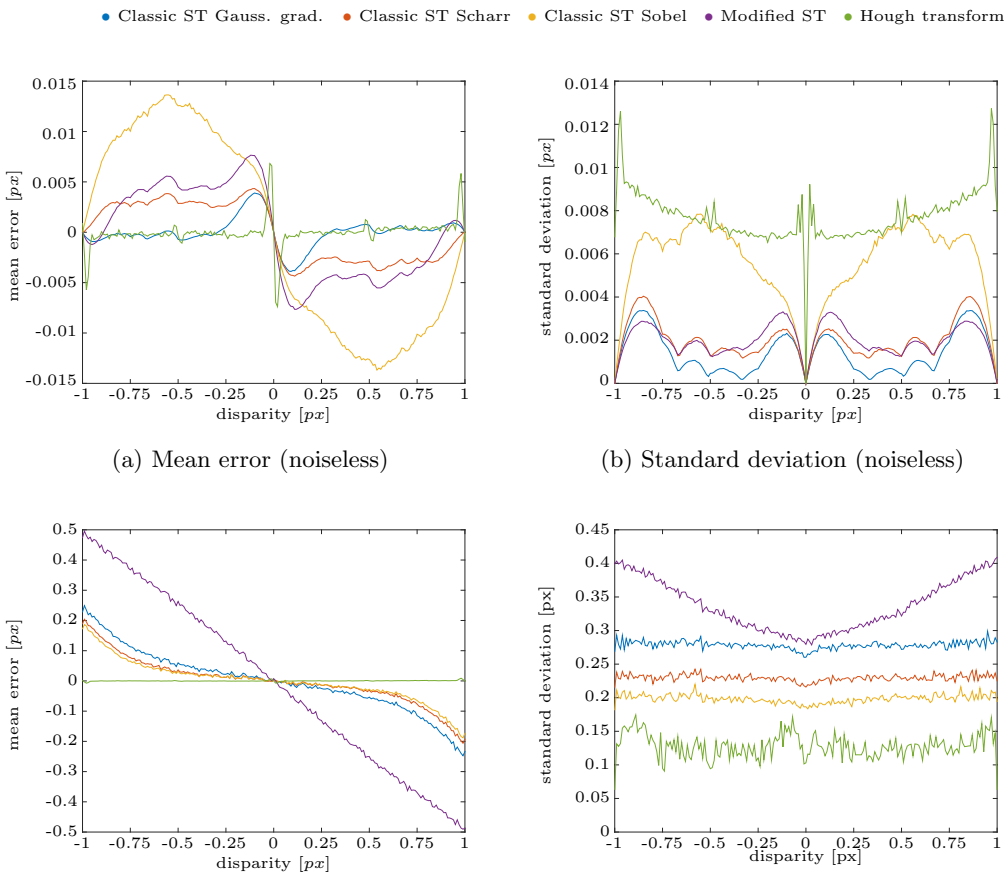
In order to better understand the relation between disparity resolution and quality of the Hough transform, we applied the proposed method on EPIs with 201 pixels height. In this way it is possible to discretize the disparity space with a resolution of $0.005\,px$. The resulting mean error and standard deviation are shown in Figure 4.11, where they are compared with the previous results on 101 pixels EPIs. The effect of the new disparity resolution can be noticed especially in the standard deviation plot, where the 201 pixels EPIs show a more precise estimation (about half of the previous standard deviation).

A complete overview of the noise's effect on the disparity estimation is shown in Figure 4.12. Here the mean disparity error of the classic structure tensor with Gaussian gradient and the Hough transform are plotted for different noise levels. These figures demonstrate the noise robustness of the Hough transform approach.

The overall accuracy of the algorithms is evaluated by means of the *root-mean-square error* (RMSE), which is computed over all N EPIs as

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{NP}\sum_{i}\sum_{p}(d_{i,p} - gt_{i,p})^2}, \quad \text{with} \quad i = 1, \ldots, N; \tag{4.8}$$

the sum is taken over all pixels (excluding the ones within a margin from the border) of the center rows of all EPIs for all the 201 disparity values. Differently from the MSE, the RMSE has the same units as the quantity being estimated, which in this case is pixel. The results are reported in Table 4.1, for both noiseless and noisy EPIs. In the noiseless case all the local tensor based methods, besides the one using the Sobel operator, have a higher accuracy than the Hough transform. As previously explained, this is due to the discretization that the Hough transform applies to the disparity space. In these experiments the disparity resolution is $0.01\,px$. Therefore, the resulting RMSE of $0.0079\,px$ agrees with this value. On the other hand, for the noisy EPIs (which better represents real EPIs) this discretization effect is negligible, and the Hough transform approach gives the best result. It could be argued that the structure tensor estimates are worse but also dense, and that, if we restrict the evaluation to the areas where the



(a) Mean error (noiseless)

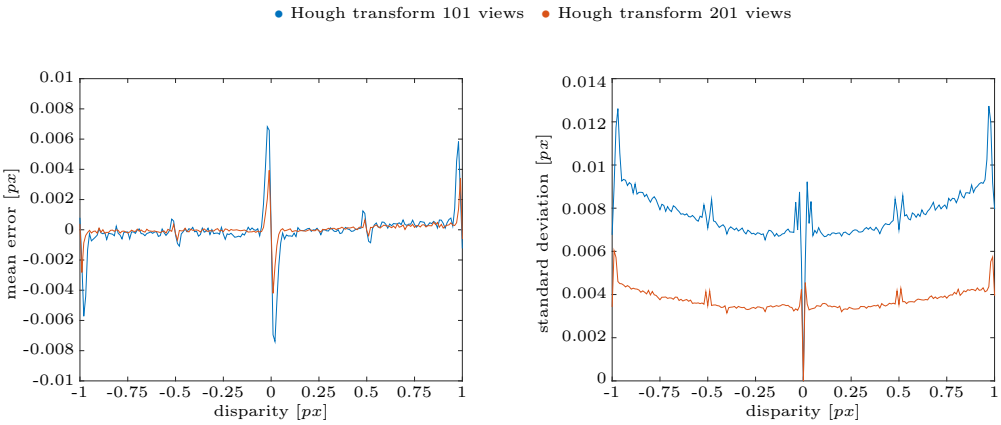
(b) Standard deviation (noiseless)

(c) Mean error (noisy)

(d) Standard deviation (noisy)

Figure 4.9: Mean disparity estimation error and standard deviation of the estimates for noiseless (a-b) and noisy EPIs (c-d): classic structure tensor with Gaussian gradient, Scharr and Sobel filters, modified structure tensor with Scharr filter, and Hough transform. The mean error represents the bias, whereas the standard deviation can be used to measure the precision. In the noiseless case the classic structure tensor with Gaussian gradient and the Hough transform have the lowest mean disparity errors. Moreover, the tensor methods have generally a lower standard deviation. In the noisy dataset the Hough transform performs better than all the tensor methods, both in error and standard deviation.

Figure 4.10: EPI of disparity 0.5 with added zero-mean Gaussian white noise of variance $\sigma_n^2 = 0.01\,px^2$.



(a) Mean error

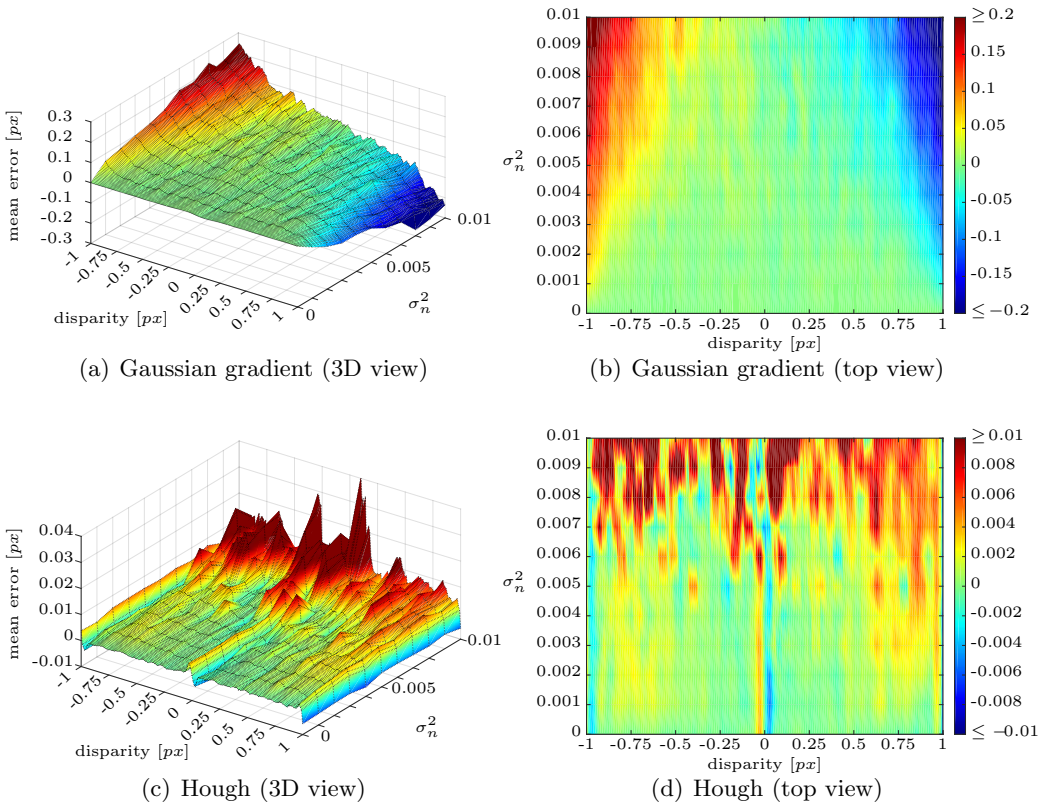
(b) Standard deviation

Figure 4.11: Mean disparity estimation error (a) and standard deviation (b) of the estimates computed with the proposed Hough transform approach from 101 pixels height (blue curves) and 201 pixels height EPIs (red curves). A higher number of pixels in the vertical axis allows a more precise angle space sampling, leading to a higher disparity resolution. For this reason the estimations of the 201 pixels EPIs' case are more precise and show a smaller mean error and standard deviation.

Hough transform finds lines, the average error would be lower. Thus, in Table 4.1 also this case is reported in columns marked with an asterisk (*). In this case it can be seen that in general, restricting the analysis to "high-coherence" regions reduces the RMSE of the tensor methods. However, the Hough approach is still the best performing algorithm in the noisy case.

In Figure 4.13 (a) we analyze what happens to the RMSE of the Gaussian gradient structure tensor when points having a coherence lower than a threshold are neglected, and what percentage of points lies above the threshold. We do the same in Figure 4.13 (b) for the Hough transform, where the line score substitutes the coherence. While for the structure tensor there is no clear value to use as threshold, for the Hough transform it can be set to 0.45, thereby retaining about 91% of points and reducing the RMSE from $0.10\,px$ to $0.027\,px$. This indicates that there is a small number of outliers which negatively influences the result, but also that the line score is a good criterion to filter them out while keeping the majority of points.

Eventually, we study the sparsity of the disparity map returned by the Hough transform. We investigate the number of lines returned by the algorithm, by counting



(a) Gaussian gradient (3D view)

(b) Gaussian gradient (top view)

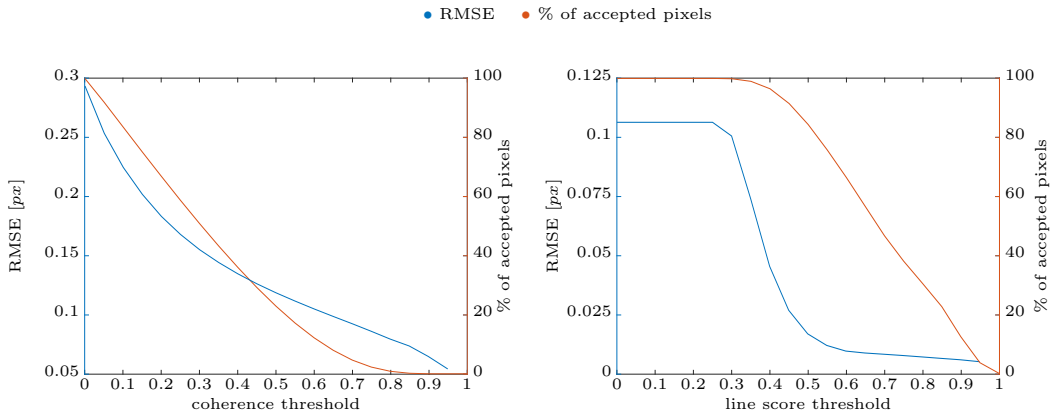
(c) Hough (3D view)

(d) Hough (top view)

Figure 4.12: Mean disparity estimation errors for classic structure tensor (top) and Hough transform (bottom) with increasing noise variance (10 levels of noise from $\sigma_n^2 = 0.001\,px^2$ to $\sigma_n^2 = 0.01\,px^2$). The Hough transform shows a much higher robustness to noise.

Method	RMSE $[px]$			
	noiseless	noiseless*	noisy	noisy*
Classic ST Gauss. Grad.	**0.0022**	0.0022	0.2926	0.2533
Classic ST Scharr	0.0037	0.0045	0.2391	0.2148
Classic ST Sobel	0.0114	0.0135	0.2068	0.1955
Modified ST	0.005	0.0047	0.4429	0.4167
Ours	0.00795	-	**0.1042**	-

Table 4.1: RMSE of the different methods, for noiseless and noised EPIs ($\sigma_n^2 = 0.01\,px^2$). The asterisk (*) indicates that the method is evaluated only in the EPI regions where the Hough transform gives a result, i.e. a line is found.



(a) Structure tensor (b) Hough transform

Figure 4.13: The plots show how the RMSE can be reduced by filtering out points having a low certainty measure, for EPIs with additive Gaussian noise of variance $\sigma^2 = 0.01\,px^2$. This is done by thresholding points based on either their coherence for the structure tensor (a), or their line score for the Hough transform (b). The percentage of accepted pixels for different threshold values is also shown.

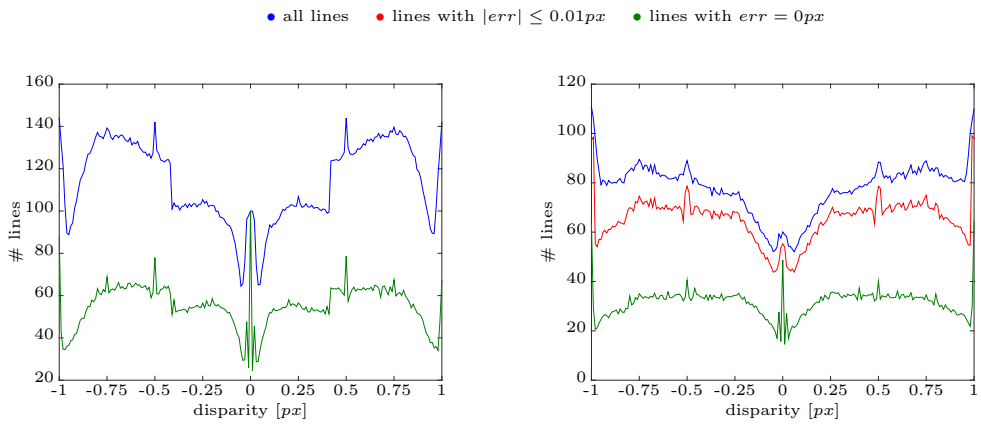
the number of pixels in the center row for which a disparity estimate exists. This is shown in Figure 4.14 for a noiseless and a noisy EPI. It can be observed that the noise does not have a critical impact on the number of lines. A fact that emerges from these plots is that a higher number of correct lines is found at disparities $0\,px$, $\pm 0.5\,px$ and $\pm 1\,px$, which is in accordance with what we observed before about the quantization of lines on the pixel grid and with the lower estimation errors at these disparities.

4.4 Light Field Datasets Evaluation

When dealing with real light field datasets, the output of a reconstruction algorithm is a disparity map. To evaluate this disparity, we use the *peak signal-to-noise ratio* (PSNR):

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE}, \tag{4.9}$$

where MAX is set to the disparity range of the ground truth and MSE is the mean square error as defined by Equation 4.7. This error measure was chosen because of its scale independence, which allows comparing results obtained with different disparity ranges. Additional error measures, such as RMSE and the percentage of bad matching



(a) Noisless EPIs

(b) Noisy EPIs

Figure 4.14: Number of lines passing through the center row returned by the Hough transform. For noiseless EPIs (a) the blue curve shows the total number of lines, whereas the green one shows the number of lines whose error $err = 0\,px$. For noisy EPIs ($\sigma_n^2 = 0.01\,px^2$) (b), the blue curve represents all lines, the red one those for which $|err| \leq 0.01\,px$ and the green one those for which $err = 0\,px$. As previously noted, disparities and thus errors are quantized, hence there are no errors in the open range $[0, 0.01]\,px$. For the noiseless EPIs the curve with $|err| \leq 0.01\,px$ is not shown since it is very similar to the one representing all lines. Peaks can be observed at disparities $0\,px$, $\pm 0.5\,px$ and $\pm 1\,px$.

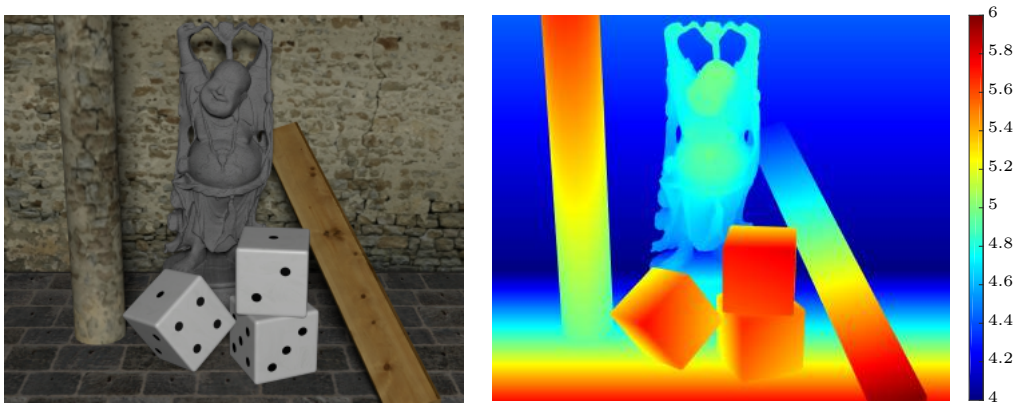
pixels (later defined in Equation 5.11) are reported in Appendix A.2. In the following, the algorithms are evaluated on synthetic datasets generated with Blender, as well as on real light fields. The proposed Hough transform algorithm is compared with the already discussed structure tensor methods (Gaussian gradient, Sobel, Scharr, and modified) and with the 2.5D structure tensor [28] from Diebold. In order to give a qualitative idea of the results, only the disparity maps and point clouds most relevant to the discussion are presented. Further disparity maps and point clouds can be found in Appendix A.3 and A.4.

4.4.1 Synthetic Datasets

In this section the results from synthetic light field datasets generated with Blender are presented. For the structure tensor some images are reported, which show the difference between the estimated disparity map and the ground truth, highlighting regions where the error is higher. For the Hough transform these images are not shown, as they are relatively hard to visualize due to the sparse representation.

4.4.1.1 Synthetic Buddha

The first synthetic scene is the Synthetic Buddha dataset. For this light field the used camera has a resolution of $2560 \times 2160 \ px$ and a pixel pitch of $6.5\,\mu m/px$. The resulting sensor size of $16.64\,mm$ has been used also for the other synthetic datasets presented in the following. A lens with a focal length of $28\,mm$ was used. The camera position and baseline were chosen to fit a given disparity range. All the results refer to the center view, shown in Figure 4.15. The classic structure tensor with Gaussian gradient was applied with an inner scale $\rho = 0.75$ and an outer scale $\sigma = 1.5$. The resulting disparity and coherence maps for a disparity range of $2\,px$ are shown in Figure 4.16, along with the estimation errors. It can be observed that the structure tensor estimation is worse at the occlusion boundaries (e.g. the right side of wooden plank) and in regions with few or texture (some parts of the dice), including dark areas (shadows). While for texture-less areas there is simply no feature path in the EPI, occlusions represent a problem for the structure tensor, which averages between the foreground and background disparities. On the contrary, the Hough transform, with its better edge localization properties, gives sharp depth discontinuities and less noise in the texture-less areas. The resulting disparity map of the Hough method is shown in Figure 4.16 (c).



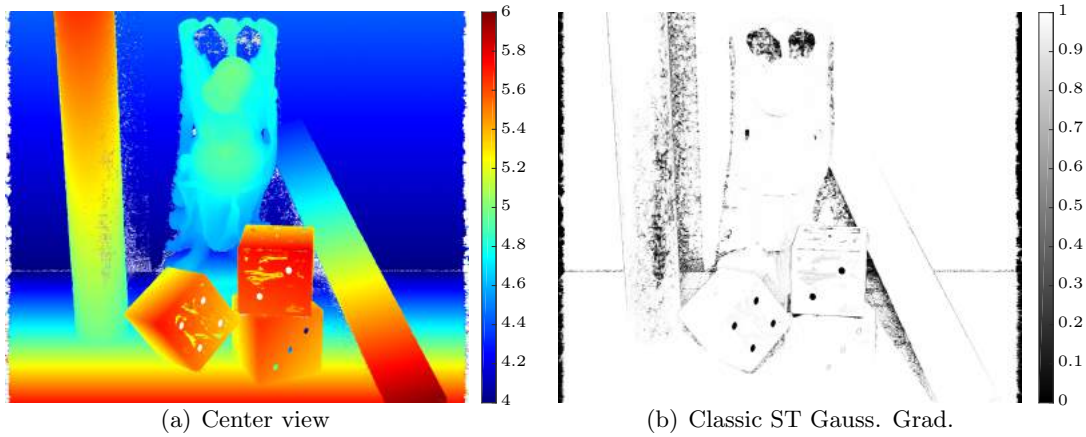
(a) Center view (b) Ground truth

Figure 4.15: Center view for the SYNTHETIC BUDDHA dataset (a) and corresponding ground truth for $\Delta\, d \;=\; 2\,px$ (b).

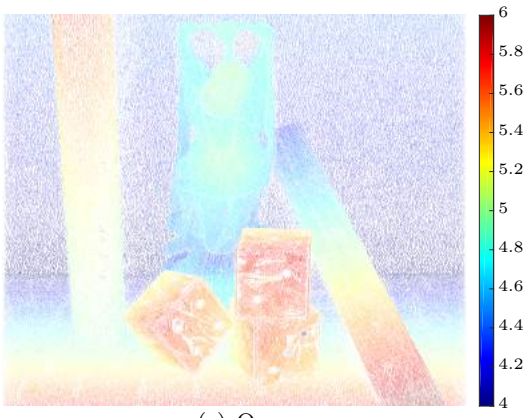
This result is visually complicated to evaluate, especially because of the sparsity of the disparity image. Therefore, to better visualize the quality of the proposed method, the point clouds generated from the center view disparities are presented in Figure 4.17. For the Hough transform the following parameters were used: edge scale 1, accumulator threshold 17, minimum line length 8, maximum gap 3, $c_{th} = 0.9$, minimum line score 0.7 (see Appendix A.1 for a description of the parameters).

For a $2\,px$ disparity range, we have $d_- = 4\,px$ and $d_+ = 6\,px$, requiring a single refocusing step at disparity of $5\,px$. Since after refocusing this disparity becomes zero, we expect to have smaller errors around it, according to what we observed in Section 4.3. This is indeed the case, as indicated in Figure 4.16 (e) by the white horizontal stripe on the wooden plank (highlighted by the green circle) and on the floor. After the refocusing, these areas have a disparity of $0\,px$, where the structure tensor error is minimized. In the $4\,px$ range case, we have $d_- = 8\,px$ and $d_+ = 12\,px$, leading to three refocusing steps at 9, 10 and 11 pixels. As in the previous case, in Figure 4.18 three low error regions (highlighted by the green circles) can be observed around the refocused disparities.

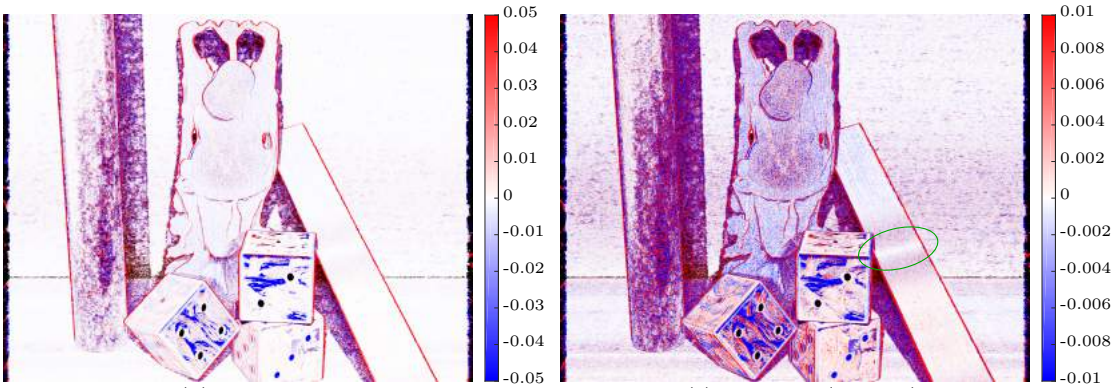
The PSNR values are shown in Table 4.2 for 41 views. We would expect the results to improve as the disparity range grows (i.e. the achievable resolution increases). However, other effects have to be considered as the range increases. For the structure



(a) Center view (b) Classic ST Gauss. Grad.



(c) Ours



(d) ST error (e) ST error (clipped)

Figure 4.16: Synthetic Buddha dataset: classic structure tensor disparity map (a), coherence map (b), and our Hough transform disparity map (c) for a disparity range of 2 px and 41 views. In the disparity map white represents an invalid estimate, i.e. outside the range $\pm 1\,px$ with respect to the disparity used for refocusing. Disparity errors for structure tensor (d-e). Red represents a positive error, blue a negative one, white zero error, and black no value. In order to highlight errors at different scales, in (d) the range is clipped to $\pm 0.05\,px$ and in (e) to $\pm 0.01\,px$.

(a) Classic ST Gauss. Grad. (b) Ours

Figure 4.17: Point clouds of the SYNTHETIC BUDDHA dataset with $\Delta\, d = 2\,px$: classic structure tensor with Gaussian gradient and coherence threshold 0.9 (a), and Hough transform computed with 101 views and line score threshold 0.7 (b). The Hough transform approach gives better results, especially at depth discontinuities.

Figure 4.18: Disparity errors of the structure tensor for a disparity range of $4\,px$ and three refocusing steps. To compute the difference, both estimated disparity and ground truth have been scaled to fit a $2\,px$ disparity range, and thus the error also refers to this range; this allows direct comparison with Figure 4.16 (e). To highlight the behavior of the estimates around the refocused disparities (9, 10 and 11) we clipped the errors to $\pm 0.01\,px$ and added three green circles.

tensor, the area around occlusion boundaries where artifacts occur becomes larger. On the other hand, for the Hough transform the poorer performance is due to the high errors that occur around disparities $0\,px$ and $\pm 1\,px$: three refocusing steps mean that each of these regions occurs three times.

Method	Disparity Range $[px]$		
	1.2	2	4
Classic ST Gauss. Grad.	25.66	27.27	27.82
Classic ST Gauss. Grad.*	29.39	30.21	31.35
Classic ST Scharr*	29.67	30.48	31.43
Classic ST 2.5D*	29.69	30.55	**31.89**
Modified ST*	27.52	28.72	29.52
Ours (41 views)	27.31	28.98	28.86
Ours (101 views)	**30.52**	**30.94**	29.56

Table 4.2: PSNR for the SYNTHETIC BUDDHA scene with different baselines (and therefore disparity ranges). The asterisk (*) marks that a coherence threshold of 0.9 has been applied. For the Hough transform a line score threshold of 0.7 was used.

For the local tensor methods it is sufficient to have enough views so that the filter kernels are entirely contained in one EPI. Increasing the number of views does not improve the results. This is not the case for the Hough transform, for which the disparity resolution, i.e. the resolution of the Hough space, increases with the number of views. Thus, the same experiment was repeated with 101 views, obtaining, as expected, a higher PSNR. A different number of views also implies adapting the parameters: the accumulator threshold was set to 40 and the minimum line length to 20.

In general we can observe how the Hough transform gives the highest PSNR for disparities requiring a single refocusing step if enough views are used. Among tensor based methods, the 2.5D variant of the classic structure tensor gives the highest value; this is expected as the additional smoothing of the tensor components removes noise, although it does not mean that the disparity map is more accurate.

In order to analyze the noise robustness, Gaussian noise of variance $\sigma_n^2 = 0.01\,px^2$ (referred to RGB intensities scaled to the $[0, 1]$ range) was added to each view in the same way as in Section 4.3.1. Disparity maps where computed with the classic structure

tensor and with the Hough transform. In the latter, the used edge scale was set to 4, in order to get a cleaner edge map. The structure tensor gives a PSNR of 8.71 while for the Hough transform the PSNR is 16.81. These results confirm that, as observed in Section 4.3.3, the Hough transform is more robust to noise than the structure tensor.

4.4.1.2 Bronze Man

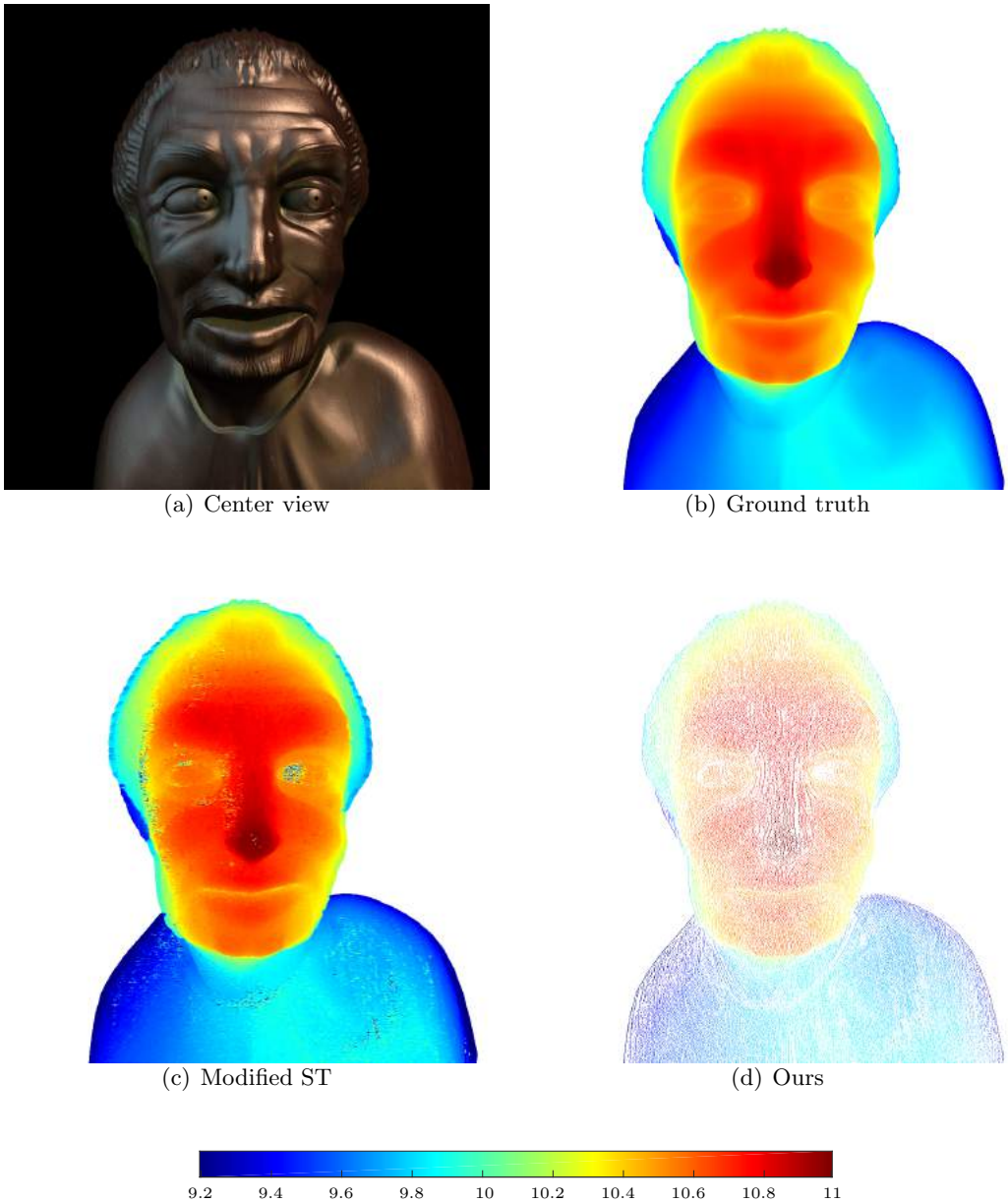
The synthetic BRONZE MAN model (Figure 4.19) was rendered with a focal length of $50\,mm$ and with a baseline of $21.2\,mm$. The 3D model is located between $Z_+ = 5.8\,m$ and $Z_- = 6.9\,m$, giving a disparity range of $\Delta\ d\ =\ 1.8\,px$. The acquired object has the peculiarity of presenting non-Lambertian surfaces, which cause the intensity of a pixel to change across views. Although for this dataset this effect is not so dramatic, it still negatively affects the estimates of the classic structure tensor. On the other hand, this dataset does not have large depth discontinuities, a characteristic that favors local tensor methods. From Table 4.3 it can be seen that the modified structure tensor gives the best result. The Hough transform approach gives a lower PSNR in comparison to the modified structure tensor, but nonetheless it performs better than all the classic ones.

Method	Disparity Range $[1.8\,px]$
Classic ST Gauss. Grad.	28.69
Classic ST Gauss. Grad.*	32.01
Classic ST Scharr*	30.02
Classic ST 2.5D*	33.22
Modified ST*	**36.01**
Ours (41 views)	34.9

Table 4.3: PSNR for the BRONZE MAN dataset with a baseline $b\ =\ 21.2\,mm$ ($\Delta\ d\ =\ 1.8\,px$). The asterisk (*) marks that a coherence threshold of 0.9 has been applied. For the Hough transform a line score threshold of 0.7 was used.

4.4.1.3 Clutter

The CLUTTER dataset, whose center view is shown in Figure 4.20 (a), has a depth range between $Z_+ = 24\,m$ and $Z_- = 40\,m$. It was captured with a $50\,mm$ lens and



(a) Center view (b) Ground truth

(c) Modified ST (d) Ours

Figure 4.19: BRONZE MAN dataset: center view (a), ground truth disparity (b), modified structure tensor disparity (c), and Hough transform disparity (d). Resolution: $1001 \times 1001\,px$.

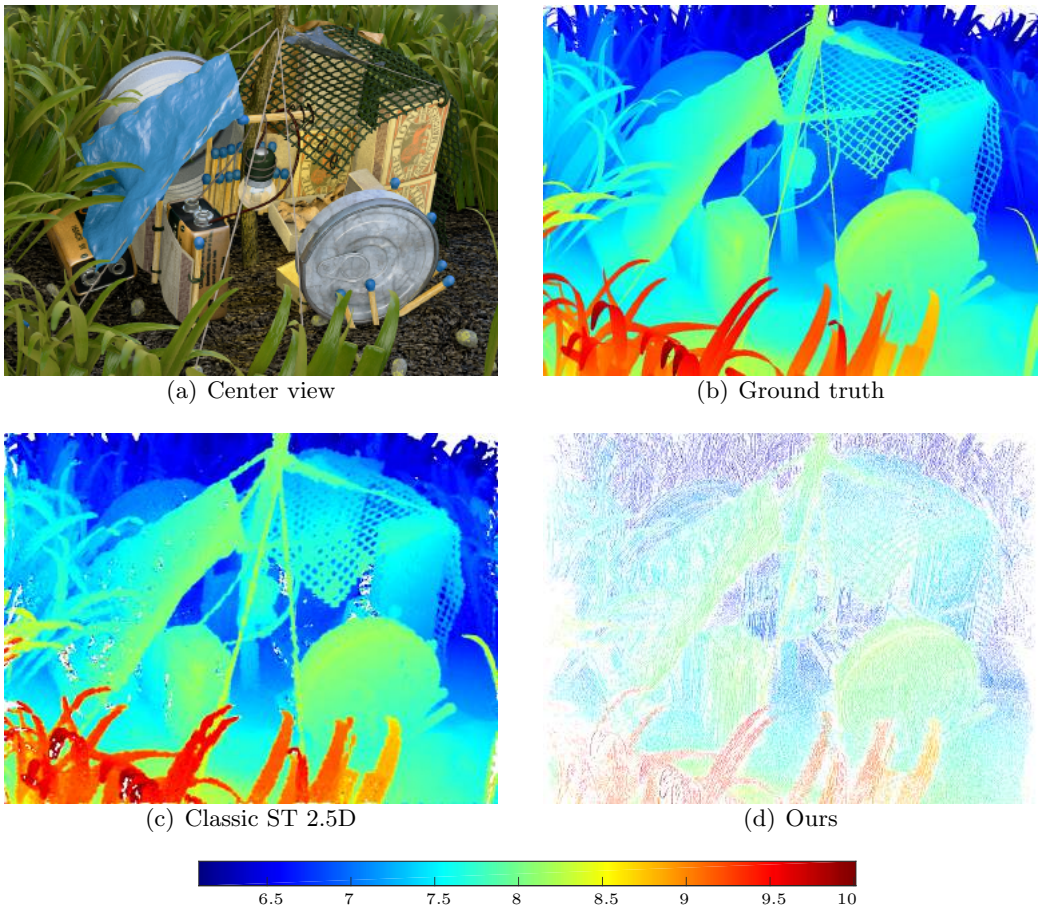
with two baselines, $39\,mm$ and $78\,mm$, yielding to disparity ranges of $\Delta\ d\ =\ 2\,px$ and $\Delta\ d\ =\ 4\,px$, respectively. Although it does not present many reflecting surfaces, this dataset is very challenging, since it has a lot of fine structures and occlusions. The disparity maps estimated with the 2.5D structure tensor and the Hough transform are shown in Figures 4.20 (c) and (d). Table 4.4 reports the PSNR values. Like in the SYNTHETIC BUDDHA dataset, which also has many occlusions, the best scoring method for $\Delta\ d\ =\ 2\,px$ is the Hough transform, if supplied with enough views (101 in this case). For $\Delta\ d\ =\ 4\,px$ so many views could not be acquired, since the frustum would have been too shallow in relation to the scene. Therefore, for this setup the best results are obtained by the 2.5D version of the classic structure tensor. As opposed to the BRONZE MAN dataset, the modified structure tensor gives the lowest score. This leads to the conclusion that, although the modified structure tensor can handle specular reflections better than the other methods, this method has difficulties with complicated structures in the scene.

Method	Disparity Range $[px]$	
	2	4
Classic ST Gauss. Grad.	21.66	22.82
Classic ST Gauss. Grad.*	25	26.6
Classic ST Scharr*	24.65	26.08
Classic ST 2.5D*	26.29	**27.94**
Modified ST*	24.58	26.08
Ours (41 views)	24.46	24.62
Ours (101 views)	**27.46**	-

Table 4.4: PSNR for the CLUTTER dataset with a baseline of $39\,mm$ ($\Delta\ d\ =\ 2\,px$) and $78\,mm$ ($\Delta\ d\ =\ 4\,px$). 101 views could not be acquired with a $78\,mm$ baseline because the frustum would have been too shallow with respect to the object.

4.4.2 Real Datasets

In this section results from three real datasets are presented. One of these is a Buddha head statue, for which "ground truth" data from a structured-light scanner was available. Additionally, two outdoor scenes are analysed.



(a) Center view (b) Ground truth

(c) Classic ST 2.5D (d) Ours

Figure 4.20: CLUTTER dataset: center view (a), ground truth disparity ($b = 78\,mm$, $\Delta\, d = 4\,px$) (b), classic structure tensor 2.5D disparity (c), and Hough transform disparity for 41 views (d). Resolution: $1024 \times 768\,px$.

4.4.2.1 Buddha Head

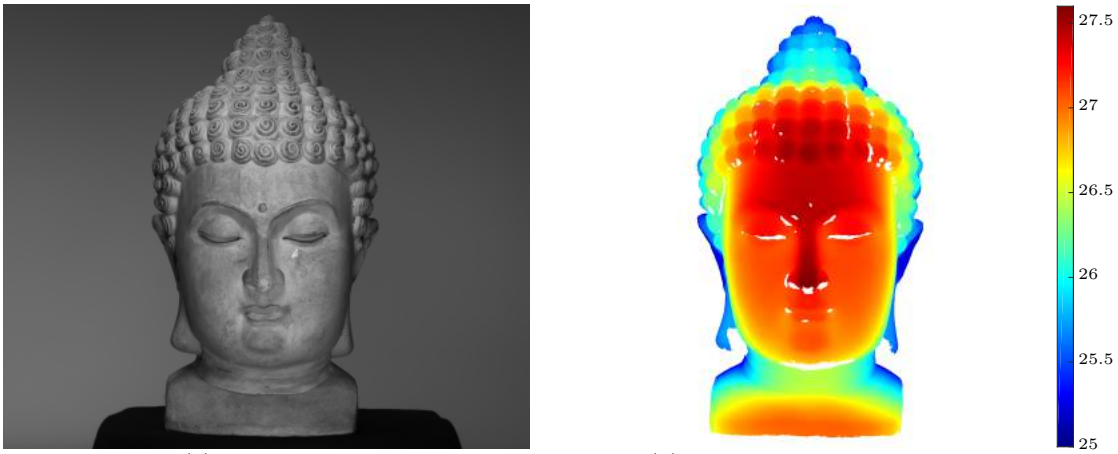
The BUDDHA HEAD sculpture was captured with a $50\,mm$ lens. The visible side of the head has a depth range $\Delta Z = 18.1\,cm$, with $Z_+ = 1.773\,m$ and $Z_- = 1.954\,m$. The light field was captured with a single camera mounted on a translation stage, as described in Section 3.4.2. Baselines of $b = 4\,mm$ ($d_- = 16.7\,px$, $d_+ = 18.4\,px$, $\Delta\, d = 1.7\,px$) and $6\,mm$ ($d_- = 25.0\,px$, $d_+ = 27.6\,px$, $\Delta\, d = 2.6\,px$) were used. For both cases two refocusing steps were executed during the disparity computation. As reference, the Buddha's head was measured with a structured-light scanner. The center view and the structured-light disparity are shown in Figure 4.21. Using this measure as ground truth implies that the reliability of the evaluation is affected by the correctness of the alignment of the estimated and reference point clouds, and by the presence of holes in the reference mesh. Nonetheless, the ranking of the algorithms based on the PSNR, shown in Table 4.5, agrees with the one shown in Table 4.3 for the Bronze Man dataset, even though the statue apparently has Lambertian surfaces. In particular, the modified structure tensor has the best score, while the Hough transform scores better than the classic structure tensor, but worse than its 2.5D variant. Unfortunately, as previously stated, a better PSNR does not necessarily mean that the disparity map is better, since it cannot measure aspects like the quality of depth discontinuities and the noise in texture-less areas. Figure 4.22 shows the center view disparities of the two reconstructions. Moreover, the point clouds are presented in Figure 4.23. Here it is possible to visualize the improvements of the Hough transform reconstruction, which again is more precise and less noisy than the other methods.

4.4.2.2 Backyard

The BACKYARD is a rather challenging dataset, as it presents underexposed noisy areas and fine detail in the form of the corrugated iron walls of the building (Figure 4.24 (a)). For the acquisition a $28\,mm$ lens was used. 43 views were captured with a baseline of $14\,mm$, yielding a total displacement of $588\,mm$. Approximately, the depth limits of the scene are $Z_+ = 17.4\,m$ and $Z_- = 28.5\,m$, which give $d_- = 2.2\,px$ and $d_+ = 3.6\,px$ ($\Delta\, d = 1.4\,px$). Figures 4.24 (b) and (c) show the results for the classic structure tensor ($\rho = 0.75$, $\sigma = 1.5$) and the Hough transform (same parameter used for

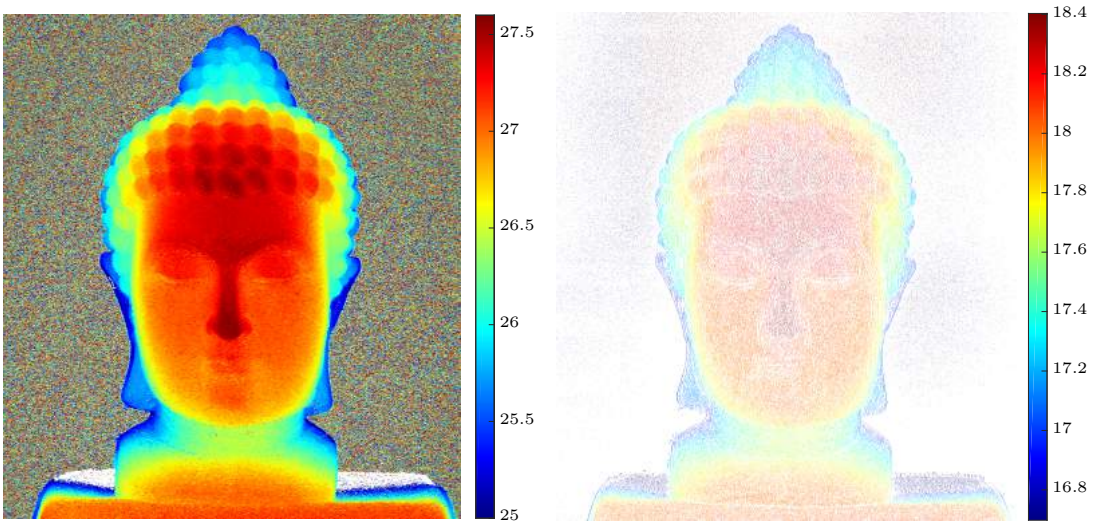
Method	Disparity Range [px]	
	1.7	2.6
Classic ST Gauss. Grad.	24.4	25.4
Classic ST Gauss. Grad.*	27.69	28.51
Classic ST Scharr*	26.65	27.76
Classic ST 2.5D*	29.79	30.55
Modified ST*	**30.81**	**31.7**
Ours (41 views)	28.24	29.27
Ours (71 views)	29.16	-

Table 4.5: PSNR for the BUDDHA HEAD dataset with a baseline of $4\,mm$ ($\Delta\ d\ =\ 1.7\,px$) and $6\,mm$ ($\Delta\ d\ =\ 2.6\,px$). 71 views could not be acquired with a $6\,mm$ baseline because the frustum would have been too shallow with respect to the object.



(a) Center view (b) Structured light disparity

Figure 4.21: BUDDHA HEAD dataset: center view (a) and structured light ground truth disparity ($b = 6\,mm$, $\Delta\ d\ =\ 2.6\,px$) (b). The holes in the ground truth are due to holes in the reference mesh.



(a) Modified ST (b) Ours

Figure 4.22: BUDDHA HEAD dataset: modified structure tensor disparity ($b = 6\,mm$, $\Delta\ d\ =\ 2.6\,px$) (a), and Hough transform disparity map for 71 views ($b = 4\,mm$, $\Delta\ d\ =\ 1.7\,px$) (b).

(a) Classic ST (b) Classic ST 2.5D (c) Modified ST (d) Ours

Figure 4.23: Point clouds for the BUDDHA HEAD dataset with $b = 6\,mm$ ($\Delta\ d\ =\ 2.6\,px$): classic structure tensor (a), classic structure tensor 2.5D (b), and modified structure tensor (c). Coherence threshold 0.7 for all the tensor methods. Hough transform computed with 71 views and line score threshold 0.6 (d).

the synthetic Buddha dataset, except for an edge scale of 1.5). The close-ups in Figures 4.24 (d) and (e) indicate a better performance of the Hough transform, particularly in the difficult underexposed areas.

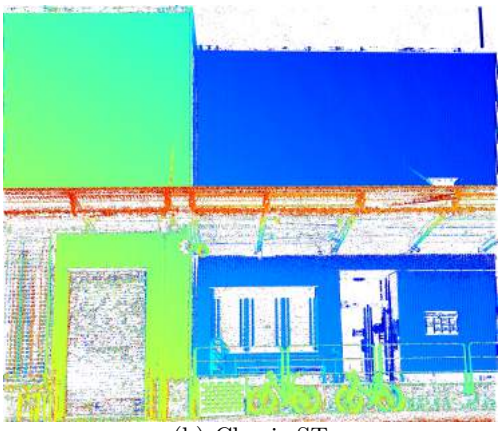
4.4.2.3 Mathematikon

The MATHEMATIKON scene, for which the center view is shown in Figure 4.25 (a), has a wide depth range, going from $Z_+ = 5.3\,m$ to $Z_- = 90\,m$. A $28\,mm$ lens and baseline of $8\,mm$ were used, yielding $d_- = 0.4\,px$, $d_+ = 6.8\,px$, and $\Delta\, d = 6.4\,px$. To process this light field, six refocusing steps were necessary. The resulting disparity maps are shown in Figures 4.25 (b) and (c) for the classic structure tensor and for the Hough transform (for 41 views). In the processing we used the same parameters as for the backyard dataset. The results highlight two situations in which both methods struggle at estimating disparities, namely texture-less areas like the vases, which are essentially solid gray, and horizontal structures, like the upper sections of the steel bike racks, which are almost parallel to the motion of the camera and do not exhibit any parallax. The fine structures of the plants can be recovered quite well by both methods. In the structure tensor disparity map, the white regions above the plants are due to the fact that the finer vegetation was moved by the wind, causing random patterns in the EPIs; this phenomenon affects the Hough transform estimates to a lesser extent. Looking at the close-up of the bike racks (Figures 4.25 (d) and (e)) we can observe how the fine vertical structures are preserved by the Hough transform, while the structure tensor blurs them out because of the averaging window.

4.5 Conclusion

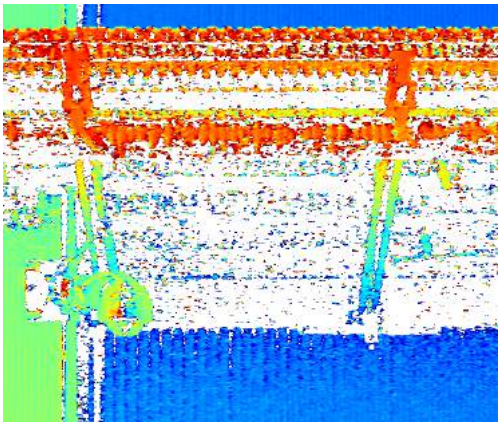
We have presented a new semi-global approach for 3D reconstruction from linear light fields. This method retrieves reliable EPI-lines by exploiting both the benefits of local and global slope estimation. The global aspect comes from the Hough transform, in which all the line's points contribute to increase the value of the accumulation matrix location corresponding to the real line. The local information from the structure tensor is used to speed-up the process and guide the disparity estimation near occlusion boundaries. The proposed approach was compared with five variants of the structure tensor: Gaussian gradient, Scharr, Sobel, modified and 2.5D. Experiments on synthetic

(a) Center view

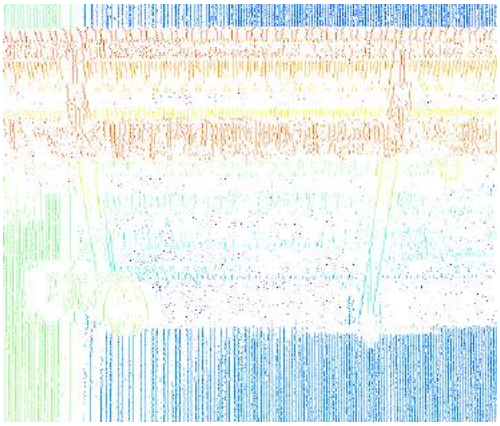


(b) Classic ST

(c) Ours



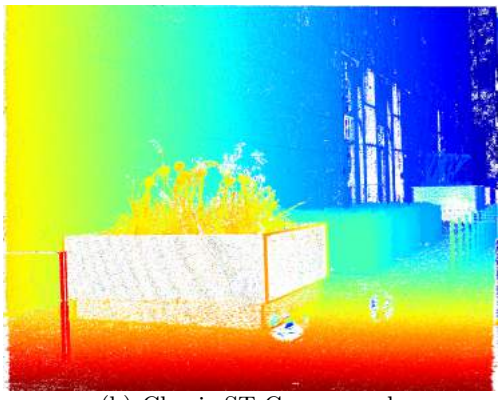
(d) Classic ST (close-up)



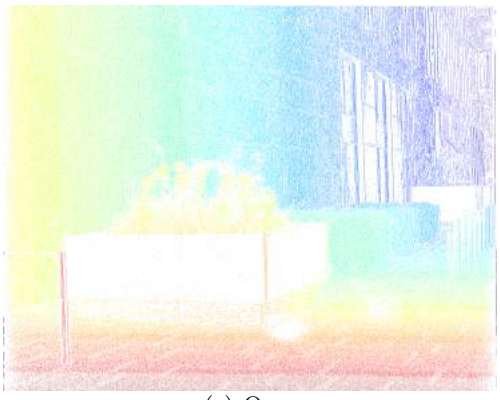
(e) Ours (close-up)

Figure 4.24: BACKYARD dataset: center view (a), disparity maps obtained with classic structure tensor (b), and with the Hough transform (c). The close-ups (d-e) highlight how structures in noisy areas are better recovered by the Hough transform.

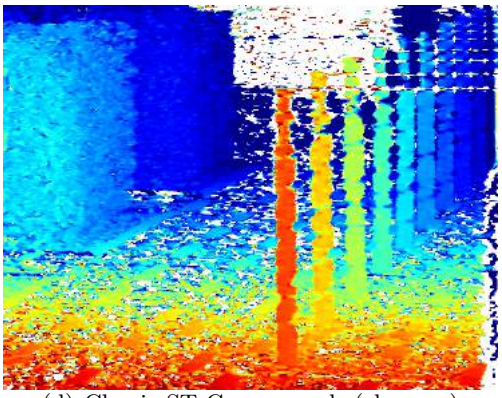
(a) Center view



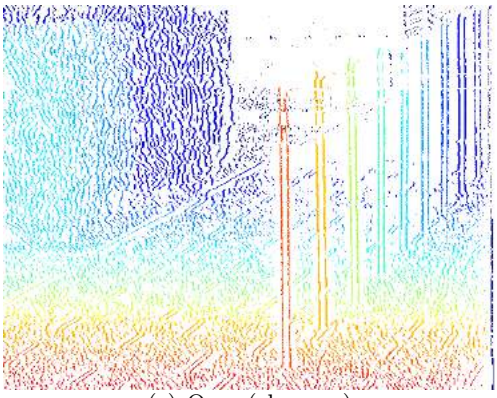
(b) Classic ST Gauss. grad.



(c) Ours



(d) Classic ST Gauss. grad. (close-up)



(e) Ours (close-up)

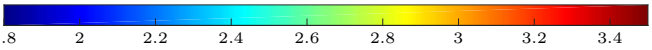


Figure 4.25: MATHEMATIKON dataset: center view (a), disparity maps obtained with classic structure tensor (b), and the Hough transform (c). The two white spots at the bottom right are caused by two dust particles on the sensor. The close-ups (d-e) of the bike racks (with scaled colors) show how vertical edges are better recovered by the Hough transform.

EPIs demonstrated that our method has a much higher robustness to noise, outperforming the local tensor approaches. Moreover, it was shown that the accuracy of the Hough transform method depends on the number of views (i.e. the EPI's height), and it can be improved by simply adding more images to the dataset. Eventually, all the methods were evaluated on real and synthetic scenes. Our approach showed better reconstruction than the local methods, especially on complicate datasets with many occlusions. This is due to the better edge localization properties of the Hough transform, which lead to sharp depth discontinuity edges and preserve fine details.

In the future, the algorithm could be further improved with respect to occlusion handling, by computing the intersection points directly from the lines' equations. Besides precise 3D reconstruction, another application of the presented method could be material classification based on BRDF estimation. In fact, the intensity variations along the trajectories could be used to associate the material properties to specific BRDFs.

Chapter 5

Depth Reconstruction from Circular Light Fields

Three dimensional reconstruction with linear light fields is based on the fact that scene points trace straight lines on the EPIs, whose slopes are inversely proportional to the distance of the point. Unfortunately, the disadvantage of this acquisition setup is that only one side of the scene can be reconstructed. To have the complete 3D shape, the target object has to be recorded from four different sides, and then the results have to be merged for the final reconstruction. This constrain makes the acquisition procedure long and tedious. Another limitation is that linear light field algorithms are generally developed for data acquired with standard *perspective lenses*. However, for certain specific applications, e.g. precise measurement tasks in optical inspection, *telecentric lenses* are better suited. This particular type of lens allows to obtain an *orthographic projection*, where two identical objects look the same even if one is closer to the camera than the other. Thus, a linear light field acquired with a telecentric lens would lead to EPIs where all the lines have the same slope, making it impossible retrieving any depth information.

To overcome all the described issues, we extend the semi-global light field approach presented in Chapter 4 to dataset acquired with a circular motion, termed circular light fields. A circular light field captures the scene either by rotating the object in front of the camera, or the camera around the object. In this way it is possible to reconstruct the full 360° shape with just one continuous acquisition. With this particular setup, every captured scene point corresponds to a curved trajectory in the EPI. Variations

(a) (b) (c) (d)

Figure 5.1: The proposed algorithm processes data generated from a circular camera motion (a), retrieving the trajectories of 3D points in the EPIs (b). The resulting depth maps can be used to generate a point cloud (c) and a mesh (d) of the target scene.

of the depth lead to sine shaped curves with different amplitudes and phase offsets, as showed in Figure 5.1. It will be shown that circular light fields can be used to retrieve depth information even from datasets acquired with a telecentric lens.

The proposed algorithm uses a coarse EPI-slope map, generated with the local structure tensor, together with a binary edge map of the EPI, to extract trajectories by using an adapted version of the Hough transform. The result, is a set of highly accurate depth maps of the target scene from all sides. Since the Hough transform uses binarized EPIs to retrieve trajectories, it is possible to get rid of the Lambertian hypothesis and process datasets with strong intensity changes along the EPI-curves. In fact, even if a trajectory is only partially visible or its intensity saturates because of a specular reflection, the Hough transform can still recover the full curve. In order to apply circular light fields to both perspective and telecentric lenses, two slightly different versions of the algorithm are proposed.

5.1 Circular Light Fields

EPI analysis was extended to the case of circular camera movements by Feldmann et al. [32]. The acquisition setup is composed of a fixed camera and an object rotating around a point M aligned with the camera's optical center C, as shown in Figure 5.2 (a).

In this section, the image formation of circular light fields is explained, for both orthographic and perspective camera projection models.

5.1.1 Orthographic Camera

The simplest camera projection is the orthographic projection, which can be obtained through a telecentric lens. Let $\mathbf{P} = [X, Y, Z]^\top$ be an arbitrary 3D point, assuming a sensor with square pixels, its projection into image coordinates (x, y) is expressed as

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & x_c \\ 0 & m & 0 & y_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{5.1}$$

where m is the telecentric lens *magnification* divided by the pixel pitch σ, and (x_c, y_c) denotes the sensor's principal point.

Figure 5.2 (a) shows the xz image plane of an orthographic camera and two points P_1 and P_2 rotating with two different radii R_{P_1} and R_{P_2} around the rotation center M with a phase θ. The points have, respectively, a phase offset ϕ_{P_1} and ϕ_{P_2}. From Figure 5.2 it is possible to define the X, Z components of the generic point $\mathbf{P}$ in polar coordinates as

$$X = R \cdot \sin(\theta + \phi) \tag{5.2a}$$

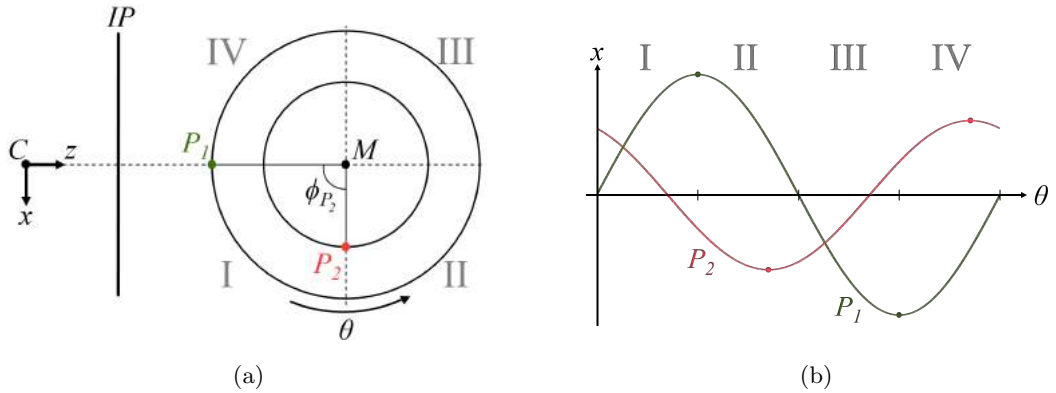
$$Z = R_M - R\cos(\theta + \phi), \tag{5.2b}$$

where R is the point's radius, $\theta \in [0, 2\pi]$ is the rotation's phase, and R_M is the distance between the center of rotation M and the camera optical center C. On Figure 5.2 (b), the corresponding trajectories of the points P_1 and P_2 projected onto the image plane are shown. In general, the trajectory of a point $\mathbf{P}$ can be derived from Equations 5.1 and 5.2 as

$$x = A \cdot \sin(\theta + \phi) + x_c \tag{5.3a}$$

$$y = H + y_c, \tag{5.3b}$$

where $A = m \cdot R$ is the trajectory's amplitude in pixel, and $H = m \cdot Y$ is the point's height in pixel.



(a) (b)

Figure 5.2: Orthographic camera: xz-plane showing the projection in the image plane IP of the points P_1 and P_2 rotating with a phase θ around M (a); trajectories of the two points in the EPI $x\theta$-plane, the dots indicate the points of maximum amplitude (b).

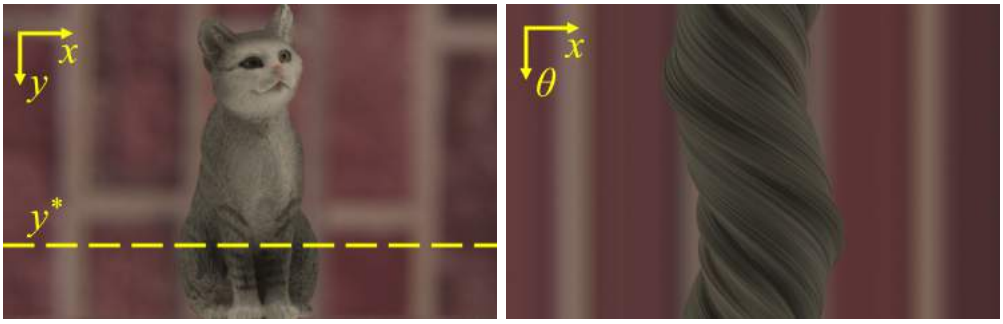
It is important to note that y only depends on the height Y of the 3D point (due to the depth independence of the orthographic projection). Consequently, in the orthographic case, the full trajectory of a rotating 3D point is imaged in one EPI. An example of such a circular light field in shown in Figure 5.3.

From Equations 5.3 it can be seen that any scene point is simply defined by its radius R and its phase offset ϕ. With this parametrization, Feldmann et al. [32] defined two occlusion rules:

1. All the points in the quadrants I and IV will occlude those in the quadrants II and III if their projection rays are equal;

2. In the quadrants I and IV all the points with a larger radius will occlude those with a smaller one if their projection rays are equal. Vice versa, for the quadrants II and III, points with a smaller radius will occlude those with a larger one.

Points moving in the quadrants I and IV correspond to curves with positive slope ($\frac{\delta x}{\delta \theta} > 0$), whereas points moving in the quadrants II and III lead to curves with negative slope ($\frac{\delta x}{\delta \theta} < 0$).



(a) (b)

Figure 5.3: Example of circular light field acquired with a telecentric lens: first image (a); EPI corresponding to the coordinate y^* highlighted by the dashed line (b).

5.1.2 Perspective Camera

With a standard lens, a *perspective projection* is obtained. The pinhole camera model defines the projection of the 3D point $\mathbf{P} = [X, Y, Z]^\top$ into image coordinates (x, y) as

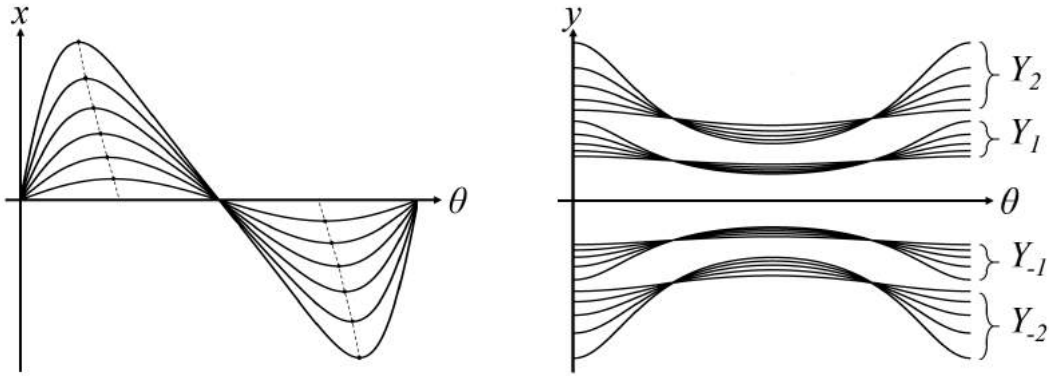
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & x_c & 0 \\ 0 & f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{5.4}$$

In this case, due to the projection's depth dependency, the sinusoidal trajectories of Section 5.1.1 are slightly distorted. If the generic 3D point $\mathbf{P}$ is again considered, its trajectory can be derived from Equations 5.2 and 5.4 as

$$x = f \cdot \frac{R \sin(\theta + \phi)}{R_M - R \cos(\theta + \phi)} + x_c \tag{5.5a}$$

$$y = f \cdot \frac{Y}{R_M - R \cos(\theta + \phi)} + y_c, \tag{5.5b}$$

where f is the focal length. Figure 5.4 shows the trajectories' x components for different rotation radii, and the y-components for different rotation radii and Y-coordinates, as a function of the rotation phase θ. For the perspective case, the trajectory of a point does not completely lie in the xz-plane, as it was in the orthographic projection, but is also moving in the y-direction during the rotation.



(a) (b)

Figure 5.4: Perspective camera: trajectories with increasing radius in the $x\theta$-plane (a), the dots indicate the points of maximum amplitude; trajectories with increasing radius and different height Y in the $y\theta$-plane (b).

5.2 Hough Transform for Orthographic Camera

As already explained in Section 4.1, the Hough transform can locate regular curves such as straight lines, circles, parabolas, and ellipses in an image. It has the advantage of being robust against noise and unaffected by occlusions or gaps in the trajectory. Also for circular light fields, the Hough transform works with binary images computed with a Canny edge detector [20]. When looking at Equations 5.3 and 5.5 it is clear that a Hough transform based approach could be used to retrieve such parametrized curves from video sequences acquired with a circular motion. Specifically, the orthographic projection case of Equations 5.3 can be correctly solved by analyzing a single EPI. On the contrary, the perspective projection case of Equations 5.5 can have only an approximated solution by analyzing EPI like slices, due to the fact that a point changes its y-coordinate during the rotation. In the following, we describe the general Hough transform algorithm for orthographic circular light fields. The perspective circular approximation will be explained in Section 5.3.

5.2.1 Hough Space Generation

With the parametrization of Equation 5.3a each EPI-trajectory t (i.e. a 3D point) can be associated with a pair (A_t, ϕ_t). The (A, ϕ) plane is termed Hough space H. The

occlusion ordering rule 1 imposes that two Hough spaces have to be computed in order to identify the trajectories: a Hough space H_1 for trajectories in the quadrants (I, IV), and a Hough space H_2 for the ones in quadrants (II, III). The two Hough spaces are discretized into cells and initially populated with zeros. This discretization depends on the chosen sensor and the acquired images. Specifically, a sensor with resolution $N_x \times N_y$ pixel yields to amplitudes $A \in [0, 1, ..., N_x/2]$. On the other hand, the phase offsets are determined by the number of images N, and are $\phi \in [0, 2\pi/N, ..., 2\pi]$. In fact, each image corresponds to a rotation angle of $2\pi/N$, which defines the phase resolution. Now that the Hough spaces are defined, each non-zero point i of the EPI binary image has to vote by incrementing of 1 the cell having coordinates (A_i, ϕ_i) in the correct Hough space. In order to determine if an edge point is related to H_1 or H_2, i.e. the trajectory point is in the (I, IV) or (II, III) quadrants, the local slope of the EPI-image is computed with the structure tensor: points with positive slope belong to the quadrants (I, IV), whereas points with negative slope belong to the quadrants (II, III).

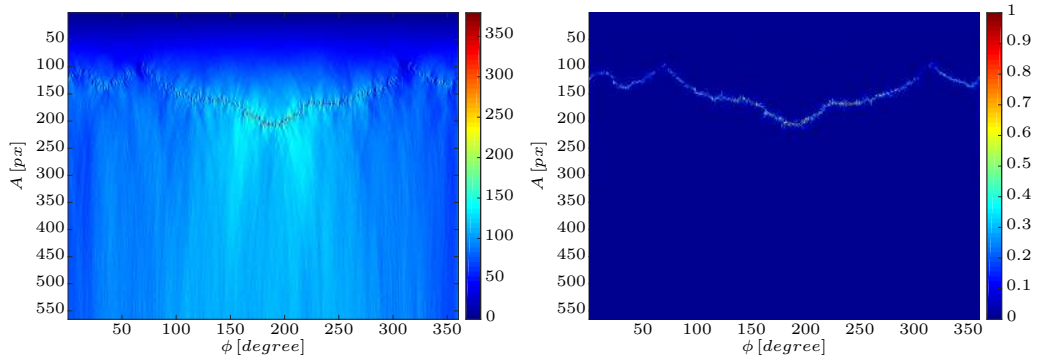
In the voting procedure, for each edge point in the EPI binary image, its coordinates (x_i, θ_i) identify a rotation phase θ_i and a point coordinate x_i, i.e. the amplitude of the possible trajectory at θ_i. From these two values it is possible to invert Equation 5.3a and derive the trajectory's phase offset ϕ with

$$\phi = \begin{cases} \arcsin\left(\dfrac{x_i - x_c}{A_i}\right) - \theta_i & \text{if } \dfrac{\delta x}{\delta \theta} > 0 \\ \arccos\left(\dfrac{x_i - x_c}{A_i}\right) - \theta_i + \dfrac{\pi}{2} & \text{otherwise.} \end{cases} \tag{5.6}$$

This equation has to be solved for all the possible trajectory's amplitudes $A_i \in [1, ..., x_i]$ (with $x_i \leq N_x/2$), and each resulting pair (A_i, ϕ_i) determines the cell in the Hough space which has to be incremented. Once all the edge points have been processed, cells whose values are local maxima or peaks define the parameters for the trajectories in the EPI.

5.2.2 Trajectories Determination

Since the 3D points can have different radii, the corresponding EPI-trajectories will also have different amplitudes. This leads, in the EPI binary image, to a set of sinusoidal



(a) (b)

Figure 5.5: Orthographic Hough transform: the Hough space H_1 for the EPI of Figure 5.3 (a); the corresponding post-processed space (rescaled between 0 and 1) for the local maxima detection (b).

curves with a different number of points: less points for small amplitude curves, more points for large amplitude ones. Consequently, in the Hough spaces, the local maxima corresponding to larger amplitudes A will have a higher cell value than local maxima corresponding to smaller amplitudes. Moreover, EPI-points with large amplitude can be fitted to more curves than points with small amplitude. Therefore, the noise in the Hough space increases with large A, as can be seen in Figure 5.5 (a). In order to correctly detect all these local maxima, each one of the two Hough spaces has to be post-processed. The first step consists of removing the low frequencies by subtracting from H its low-pass filtered version $H_{LP} = \mathcal{G}_\rho * H$, where $\mathcal{G}_\rho$ is the Gaussian filter defined in Equation 3.21. Then, the result is rescaled by multiplying it with a *weighting matrix* W which gives more weight to small amplitudes and reduces high amplitudes. All the columns of W have the same weighting vector: an exponential function $e^{-0.001} * [1, 2, ..., A_{max}]^\top$ was chosen. Eventually, thanks to the post-processing, it is possible to apply a global threshold (Otsu's method) to identify all the local maxima. The Hough space H_1 for the EPI of Figure 5.3 and the corresponding post-processed space are shown in Figure 5.5.

5.2.3 Trajectories Propagation

Each one of the identified local maxima is a pair (A, ϕ) which defines a trajectory that will be propagated in the EPI. For this task, the occlusion ordering rules of Section 5.1.1

are fundamental. To ensure the visibility of the foreground points, all the pairs (A, ϕ) from H_1 are sorted in descending order of amplitude A, whereas the pairs from H_2 are sorted in ascending order of amplitude (see rule 2). For each trajectory, defined by a pair (A_i, ϕ_i), its x-coordinates are computed with Equation 5.3a. Then, an *occlusion visibility range*, based on the rules defined in Section 5.1.1, is used to determine the phase locations θ, i.e. the EPI vertical coordinates, where the trajectory is visible. This range differs from H_1 to H_2, and is defined as

$$\begin{aligned} &\frac{\pi}{2} < \theta + \phi_i < \frac{3}{2}\pi && \text{for } H_1 \\ &0 < \theta + \phi_i < \frac{\pi}{2} \wedge \frac{3}{2}\pi < \theta + \phi_i < 2\pi && \text{for } H_2. \end{aligned} \tag{5.7}$$

For example, the point P_1 in Figure 5.2 has $\phi_{P_1} = 0$ and a visibility range equal to $\left[\frac{\pi}{2}, \frac{3}{2}\pi\right]$, i.e. the quadrants II and III.

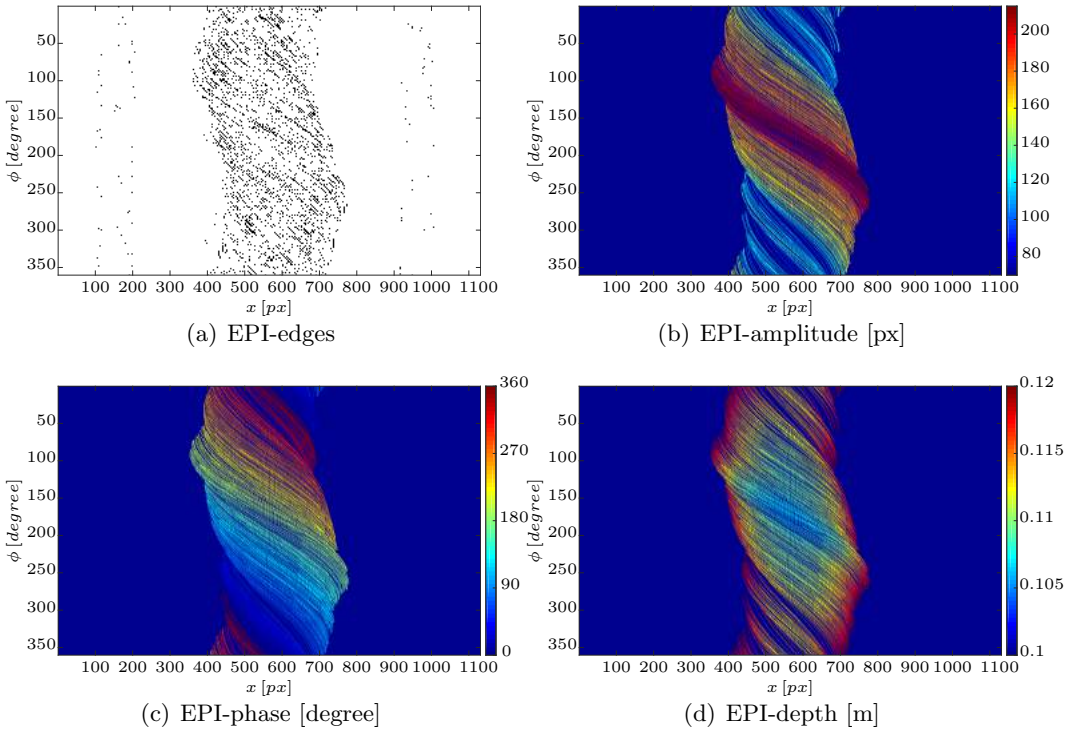
In order to take into account already propagated trajectories, and avoid new ones to overwrite them, an *EPI binary mask* is introduced. Moreover, to prevent propagation in wrong areas, trajectories from H_1 (H_2) are only propagated where the EPI-slopes are positive (negative). Eventually, the remaining portion of the trajectory can be written in the EPI, and in parallel into the EPI-mask. These steps are repeated for all the pairs (A, ϕ). The first propagated trajectories are the ones related to points belonging to the quadrants (I, IV), i.e. H_1. Then, also the trajectories from H_2 are propagated (see rule 1). The propagation procedure is summarized in Algorithm 1.

foreach $(A_i, \phi_i) \in H$ **do**
    compute the trajectory coordinates (x_i, θ_i);
    remove occluded coordinates;
    remove masked coordinates;
    remove wrong slope coordinates;
    propagate the remaining trajectory portion;
end

Algorithm 1: Trajectory propagation.

5.2.4 EPI-Depth Generation

Once all the trajectories have been propagated, it is straightforward to compute the depth map. In fact, any 3D point $\mathbf{P}$, which corresponds to a trajectory with parameters



(a) EPI-edges (b) EPI-amplitude [px]

(c) EPI-phase [degree] (d) EPI-depth [m]

Figure 5.6: The input EPI-edge image (Canny) and the results of the orthographic Hough transform: EPI-amplitude, EPI-phase, and EPI-depth image. Note that the depth of a trajectory changes along the trajectory itself, since the 3D point is moving in space, whereas amplitude and phase are constant for each trajectory.

(A, ϕ), has a depth Z, with respect to the origin C, defined by Equation 5.2b, with $R = m \cdot A$. The results of the orthographic Hough transform for the EPI of Figure 5.3 (b) are shown in Figure 5.6.

By applying the described steps to all the EPIs of the image volume (i.e. for each y-coordinate), the whole scene can be reconstructed. The final output is a set of sparse depth images, one for each rotation angle θ.

5.3 Hough Transform for Perspective Camera

When a standard perspective camera is considered, Equations 5.5 show that a trajectory is not confined in a single EPI, but also moves in the $y\theta$-plane, leading to a curve through the whole 3D image volume. The y-shift increases with the distance between a 3D point

and the horizontal plane through the camera's optical center, where there is no shift. Therefore, in order to find a trajectory, a 3D search in the full image volume should be performed, as described by Feldmann et al. [32]. In this section, the Hough transform approach is adapted to the perspective projection case. In order to continue using EPI like slices, we propose an approximate solution which neglects the y-shift. Even though the full trajectory is not available, a portion of it is always visible in the EPI. One of the advantages of the Hough transform is that even a portion of a curve can be retrieved if it has enough support. Therefore, it is possible to reconstruct the EPI-trajectories and achieve good results even with this approximation. The algorithm for the perspective case is similar to the orthographic one, with a few differences described in the following.

5.3.1 Hough Space Generation

As in the orthographic case, the discretization of the Hough spaces depends on the chosen sensor and the acquired images. In this case, the relation between amplitude A in pixel and radius R in meters is defined in [33] as

$$R = 2 \cdot \frac{R_M \tan\left(1/\left(2FOV\right)\right) \cdot A}{\sqrt{4\left(\tan\left(1/\left(2FOV\right)\right)\right)^2 \cdot A^2 + N_x^2}}, \tag{5.8}$$

where $FOV = 2\arctan\left(\left(\sigma * N_x\right)/2f\right)$ is the field of view. With this formula it is possible to associate the correct radius value to each trajectory's amplitude $A \in [0, 1, ..., N_x/2]$.

In the perspective case, the behavior of a trajectory in the xz image plane is determined by Equation 5.5a. Therefore, in the voting procedure this equation has to be solved in order to find the trajectory's phase offset ϕ from the EPI binary image point (x_i, θ_i). We chose to invert the equation by means of a look-up table. The *trajectories determination* via local maxima detection follows the same procedure used for the orthographic case in Section 5.2.2.

5.3.2 Trajectories Propagation

Once a trajectory is determined, its amplitude A (radius R) and phase offset ϕ are used to compute its coordinates through the EPI. The propagation procedure is similar to the one described in Section 5.2.3, with two differences:

1. The x-coordinates are computed with Equation 5.5a;

2. The trajectories are no longer perfect sines. As can be seen in Figure 5.4(a), three-dimensional points with larger radius R deviate more from the ideal sinusoidal curve. The phase of the maximum amplitude, which determines the *occlusion visibility range* is

$$\phi_{max} = \arccos\left(R/R_M\right). \tag{5.9}$$

From this peak it is possible to determine the segments where the trajectory is visible and can be propagated.

The remaining steps are the same as in Algorithm 1. Eventually, the depth maps are computed by projecting every 3D point into each camera's image plane, taking into account the perspective projection.

5.4 Experiments and Results

To evaluate the quality of the reconstruction, tests with both synthetic and real datasets were performed. The result of the proposed algorithm are a set of depth images, one for each rotation angle θ, which can be converted into a point cloud. Afterwards, it is possible to generate a mesh from the point cloud. To this end, we used the *Poisson Surface Reconstruction* of [50]. The same meshing procedure was employed to generate meshes from the point cloud obtained with two recent publicly available multi-view algorithms. The first is the patch based method *Clustering Views for Multi-view Stereo* (CMVS) from Furukawa and Ponce [36], which has an optimized view selection that discards some images due to the small baseline. The second is the *Multi-View Environment* (MVE) from Goesele et al. [39], which computes per-image depth maps, later merged in 3D space. For the evaluation, the obtained meshes are aligned with the ground truth by means of the *iterative closest point* (ICP) algorithm [8]. In the comparison, is important to note that the two multi-view methods are designed to process data acquired with perspective lenses. For all the datasets we report a visual comparison of the resulting meshes. Additionally, for the synthetic dataset we provide also quantitative results, since the ground truth is available.

5.4.1 Synthetic Datasets

Similarly to what was presented in Section 4.4.1, we used Blender to generate synthetic circular light fields of a test scene. The accuracy of the final reconstructions is evaluated

by means of the *one-sided Hausdorff distance* [76]. This measure gives the geometric difference between the reconstructed mesh and the ground truth, and is defined as

$$\sup_{x \in X} \inf_{y \in Y} d\left(x, y\right), \tag{5.10}$$

where X is the reference mesh (the reconstructed one), Y is the target mesh (the ground truth), and $d(x, y)$ is the distance between the 3D points x and y. This operation is performed with Meshlab [21], which searches, for each point x of the reconstructed mesh X, the closest point y on the ground truth mesh Y. We will report the RMSE in percentage normalized with respect to the diagonal of the bounding box of the mesh, a measure that can be always understood without knowing anything about the mesh units. Additionally, we compute the *percentage of bad matching pixels* (BadPix) [81] of the reconstructions. The general definition of the BadPix measure for a disparity map d and a ground truth disparity gt is

$$\text{BadPix} = \frac{1}{P} \sum_{p} \left(|d_p - g_p| > \delta\right), \tag{5.11}$$

where d_p is the estimated disparity at pixel p, gt_p is the correspondent ground truth disparity, P is the total number of pixels, and δ is the error tolerance. This measure can be directly applied to the algorithms' depth maps, as well as the 3D meshes, by means of the Hausdorff distance.

Note that for computing both RMSE and BadPix, we discarded points with a distance larger than 5% of the diagonal of the bounding box. In this way outliers are not considered.

5.4.1.1 Synthetic Buddha Head

In order to evaluate the robustness with respect to specular surfaces, we decided to use the mesh obtained with the structured light scan of the BUDDHA HEAD sculpture (see Section 4.4.2.1). Specifically, we used Blender to generate synthetic datasets by setting the surface properties to Lambertian and specular. For each type of surface, two datasets were created by setting the virtual cameras to telecentric and perspective, for a total of four datasets. All the datasets are composed of 720 images (i.e. one image each 0.5°) with a resolution of 1001×1001 pixel, and pixel pitch $\sigma = 6\,\mu$m. The focal length for the perspective camera is $f = 18$ [mm], whereas the telecentric camera has a

(a) (b) (c)

Figure 5.7: Synthetic circular light field BUDDHA HEAD generated with a perspective lens: one frame for the specular dataset (a). EPIs corresponding to the y-coordinate highlighted by the yellow dashed line: specular dataset (b) and Lambertian dataset (c). Note the strong intensity variations in the EPI-trajectories for the specular case.

magnification of 0.1. A frame of the perspective light field, as well as two EPIs showing the specular and the Lambertian case are showed in Figure 5.7. In addition to the following evaluation, supplementary figures showing the quality of the reconstructions are presented in Appendix B.2.

Figure 5.8 shows a visual comparison of the obtained meshes for the telecentric lens and Lambertian surface. The results are visually quite similar, however our method seems less noisy than MVE and with more details than CMVS. These impressions will be later confirmed in the quantitative evaluation.

In the telecentric specular dataset, showed in Figure 5.9, CMVS and MVE have much more problems due to the non-Lambertian effects on the surface. Especially CMVS, which has clear errors in the reconstruction. On the contrary, our Hough transform based approach still gives a very good reconstruction, with much less noise, especially in the Buddha's face. This is due to the way the voting procedure of the Hough transform uses all the available images to find the EPI-trajectories, which can be retrieved even if they are only partially visible.

The resulting meshes for the datasets acquired with the perspective lens (for both Lambertian and specular surfaces) are showed in Figures 5.10 and 5.11. Now our method is using an approximated model (since we are ignoring the y-shift), whereas MVE and CMVS assume the right pinhole model. Nevertheless, the results are quite

Dataset	PSNR [dB]		
	Ours	CMVS	MVE
Telecentric Lambertian	**36.24**	33.45	33.49
Telecentric specular	**35.96**	28.08	28.48
Perspective Lambertian	**33.62**	32.72	32.82
Perspective specular	**33.96**	27.56	27.55

Table 5.1: BUDDHA HEAD synthetic datasets: PSNR of the meshes. Our method outperforms the others and is not affected by specular surfaces.

similar to the telecentric case. Our method produces superior meshes, more precise and less noisy, especially for the dataset with specular surface.

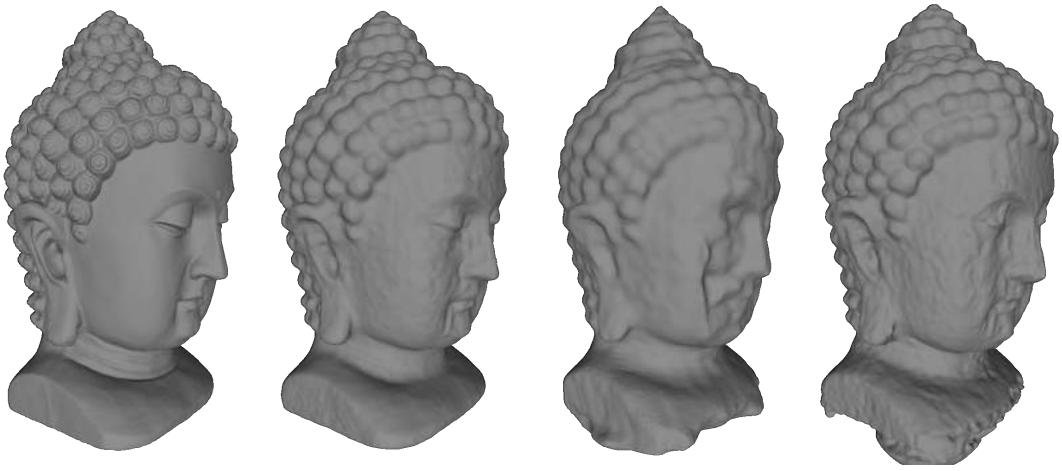
The visual results gave us an idea of the algorithms' quality. However, a quantitative evaluation is important to confirm the impressions of the visual comparison. Table 5.1 shows the PSNR computed through the Hausdorff distance of the meshes in percentage. Our Hough transform approach is always better than the multi-view methods. Moreover, our method is robust to specular surfaces. In fact, the PSNR is almost the same for both Lambertian and specular datasets. Additional error measures, such as the BadPix and the RMSE of the reconstructed meshes are reported in Appendix B.2.

Additionally, we evaluate the quality of the reconstruction for all the 360° views. To this end, we used Blender to generate circular light fields containing the depth maps of the reconstructed meshes (i.e. with our Hough transform approach, CMVS, and MVE), and then we compute the BadPix with respect to the ground truth depth for each view. For this measure, we chose an error tolerance $\delta = 0.05$ [m]. Figure 5.12 shows the plots for all the four cases; the starting view is the front Buddha face, then the object rotates clockwise for 360 °.

The telecentric camera and Lambertian surface case is showed in Figure 5.12 (a), where it can be seen that our approach gives the best result for all the views. MVE and CMVS have very similar performances, and have more bad matching pixels than our method. It is important to remember that, in this case, both the multi-view algorithms are using an approximated model, since they were designed for perspective cameras. Figure 5.12 (c) shows the same plot for the perspective camera and Lambertian surface

(a) Ground Truth (b) Ours (c) CMVS (d) MVE

Figure 5.8: BUDDHA HEAD Lambertian dataset acquired with a telecentric camera, mesh comparison: ground truth (a), ours (b), CMVS (c), and MVE (d). The results are quite similar, however our Hough transform approach looks smoother than MVE and slightly more detailed than CMVS.

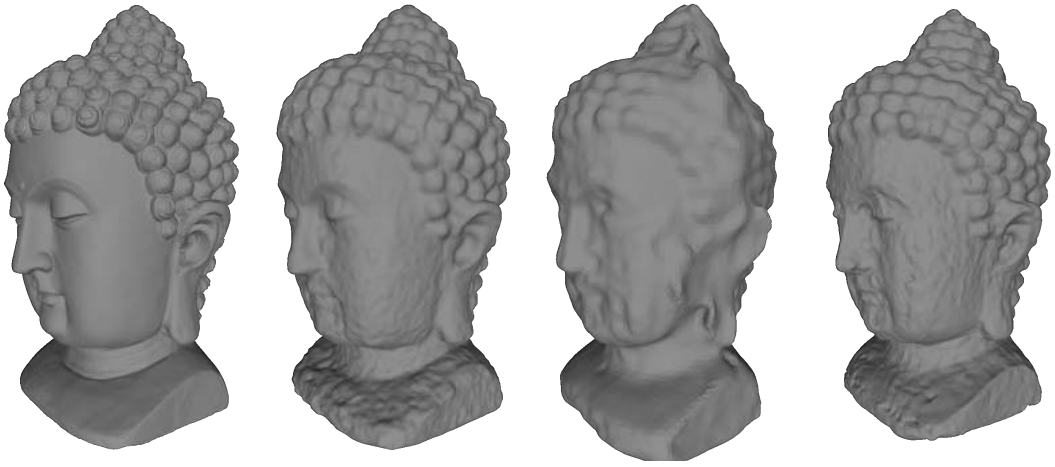


(a) Ground Truth (b) Ours (c) CMVS (d) MVE

Figure 5.9: BUDDHA HEAD specular dataset acquired with a telecentric camera, mesh comparison: ground truth (a), ours (b), CMVS (c), and MVE (d). Our circular light field approach outperforms the two multi-view algorithms.

(a) Ground Truth (b) Ours (c) CMVS (d) MVE

Figure 5.10: BUDDHA HEAD Lambertian dataset acquired with a perspective camera, mesh comparison: ground truth (a), ours (b), CMVS (c), and MVE (d). As in the telecentric case, our method looks slightly better than the other two.



(a) Ground Truth (b) Ours (c) CMVS (d) MVE

Figure 5.11: BUDDHA HEAD specular dataset acquired with a perspective camera, mesh comparison: ground truth (a), ours (b), CMVS (c), and MVE (d). Our circular light field approach outperforms the two multi-view algorithms.

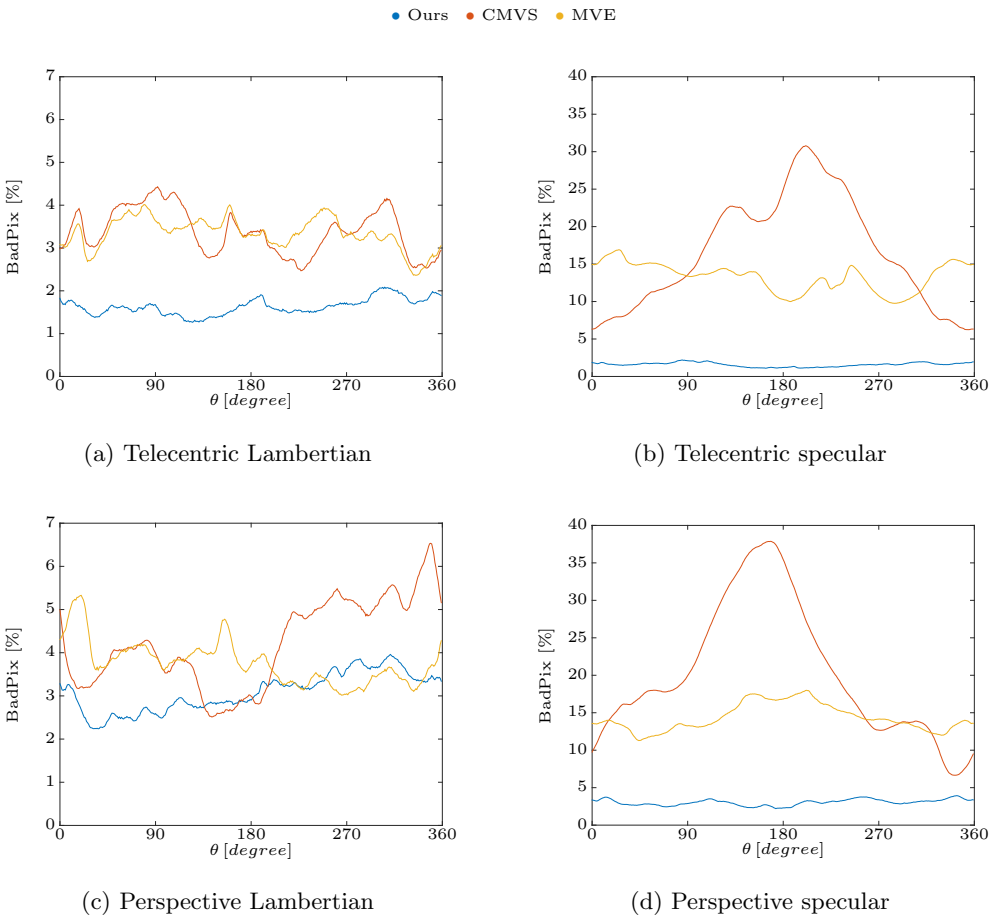
case. Here the three algorithms perform almost the same, but overall our method is still the best (even though now we use an approximated model, which ignores the y-shift). The potential of our Hough transform light field approach can be seen in Figures 5.12 (b) and 5.12 (d), where the BadPix for the two specular datasets (telecentric and perspective) are presented. The proposed approach dramatically outperforms the multi-view methods for all the views. It is interesting to notice that for both the specular datasets, CMVS has a large error around 150°. This angle corresponds to the back part of the head, which presents many structures and should be easier to reconstruct. However, the specular surface gives some problems to CMVS, which is failing in this area.

5.4.2 Real Datasets

Real datasets were acquired both with a telecentric lens (Zeiss Visionmes 105/11) for the orthographic case, and a standard lens (Zeiss Makro-planar 2/100 ZF.2) for the perspective one. The former [103] is a telecentric lens with a working distance of 121 [mm] and a magnification of 0.1. The latter [102] is a standard lens with focal length 100 [mm]. Calibration was performed for both the lenses, in order to remove distortion from the images and determine the correct rotation center. We used again the pco.edge 5.5 [71] camera with a resolution of 2560×2160 pixel and a pixel pitch of $\sigma = 6.5\,\mu$m. Test objects were placed on a high precision rotation stage (Owis DTM 130N [69]), and light fields composed of $N = 720$ images were acquired. The acquisition setup, as well as the rotation stage are shown in Figure 5.13. In the following, the results of two plastic animal figures as well as one highly specular metallic drill bit are presented.

5.4.2.1 Cat

The results of the orthographic case for the Cat dataset are showed in Figure 5.14. When looking at the mesh generated with CMVS, it can be seen that it lacks in details, and the cat's tail is lost. Also the mesh produced with MVE presents some issues, it is noisy and with many errors on the surface. Differently from the multi-view methods, our algorithm provides the a very good reconstruction. For the perspective case, presented in Figure 5.15, the MVE reconstruction is not available since the algorithm failed with



(a) Telecentric Lambertian

(b) Telecentric specular

(c) Perspective Lambertian

(d) Perspective specular

Figure 5.12: BUDDHA HEAD synthetic datasets: BadPix 0.05 [m] for all the 360° views. Comparison of our method with CMVS and MVE for four possible cases: telecentric Lambertian (a), telecentric specular (b), perspective Lambertian (c), and perspective specular (d). Our approach dramatically outperforms the others, especially for specular surfaces. Note that the scales of the Lambertian cases (a-c) and specular cases (b-d) plots are different.

(a) (b)

Figure 5.13: Circular light fields acquisition setup: the rotation stage OWIS DTM 130N (a), and the whole setup with the telecentric lens Zeiss Visionmes 105/11 (b).

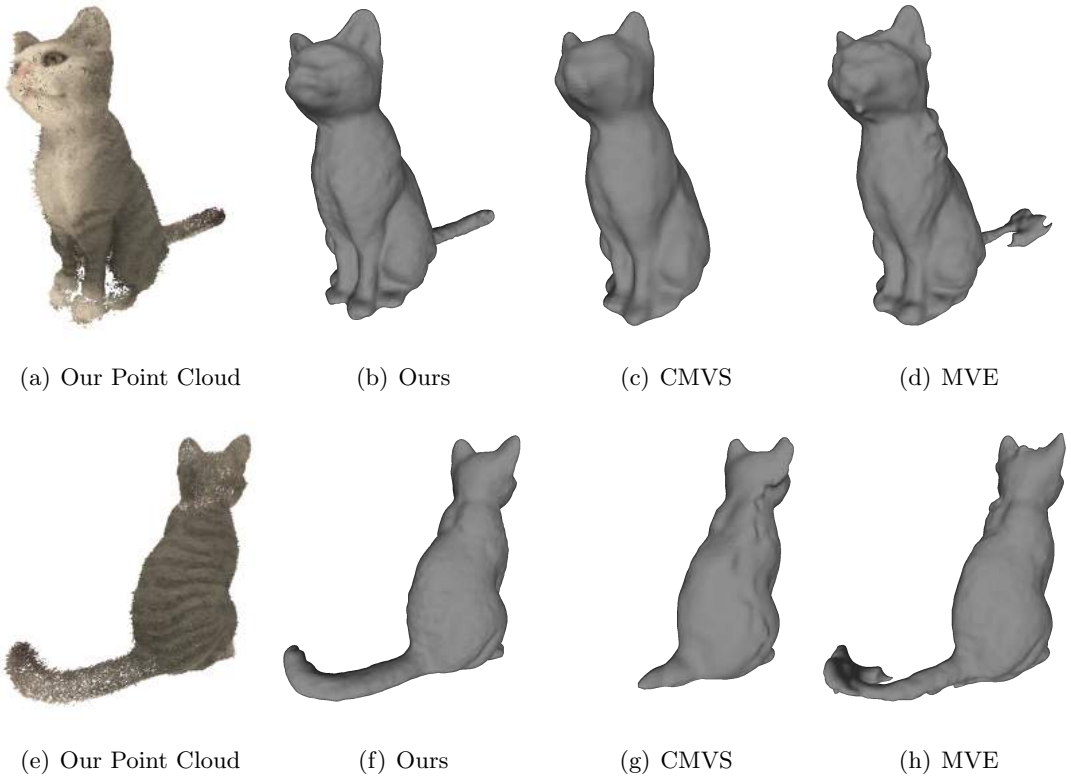
this dataset. Our approach produces a result comparable to CMVS, and it is difficult to determine which is the best reconstruction.

5.4.2.2 Seahorse

Similarly to the Cat dataset, also in the Seahorse dataset our approach outperforms the multi-view stereo algorithms in the orthographic case. From the images presented in Figure 5.16 it is clearly visible that the light field method produces a superior mesh. The comparison is more difficult in the reconstructions from data acquired with the standard lens, shown in Figure 5.17. Here the three results are very similar, with our reconstruction showing a more noisy surface. Once again we can say that our approximation is comparable to the other two algorithms.

5.4.3 Drill Bit

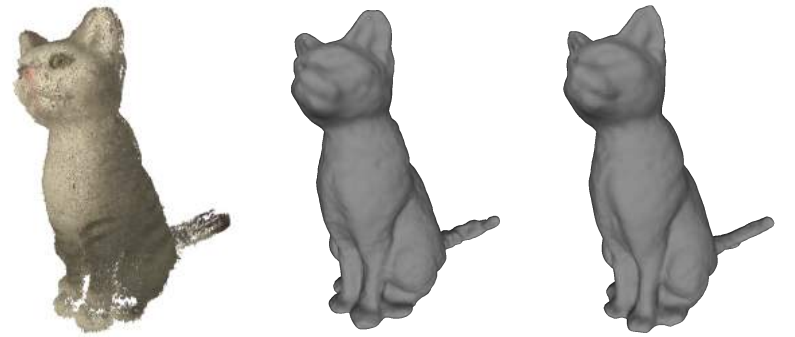
In the third real dataset, we tested the robustness of our algorithm against non-Lambertian surfaces by reconstructing a Drill Bit. This is a highly specular and challenging metallic part. Nevertheless, we can precisely reconstruct the object and correctly retrieve the EPI-trajectories even with strong intensity variations. The results comparison between our light field approach and the multi-view methods are reported in Figures 5.18 and 5.19 for the telecentric and perspective case, respectively. Also for this dataset, the MVE reconstruction algorithm failed in the reconstruction.



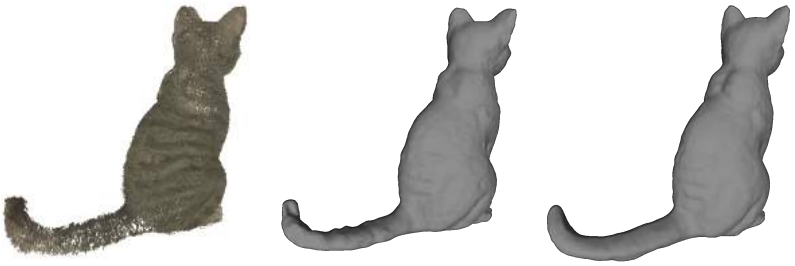
(a) Our Point Cloud (b) Ours (c) CMVS (d) MVE

(e) Our Point Cloud (f) Ours (g) CMVS (h) MVE

Figure 5.14: CAT dataset acquired with a telecentric camera. First view: our point cloud (a), our mesh (b), CMVS (c), and MVE (d). Second view: our point cloud (e), our mesh (f), CMVS (g), and MVE (h).

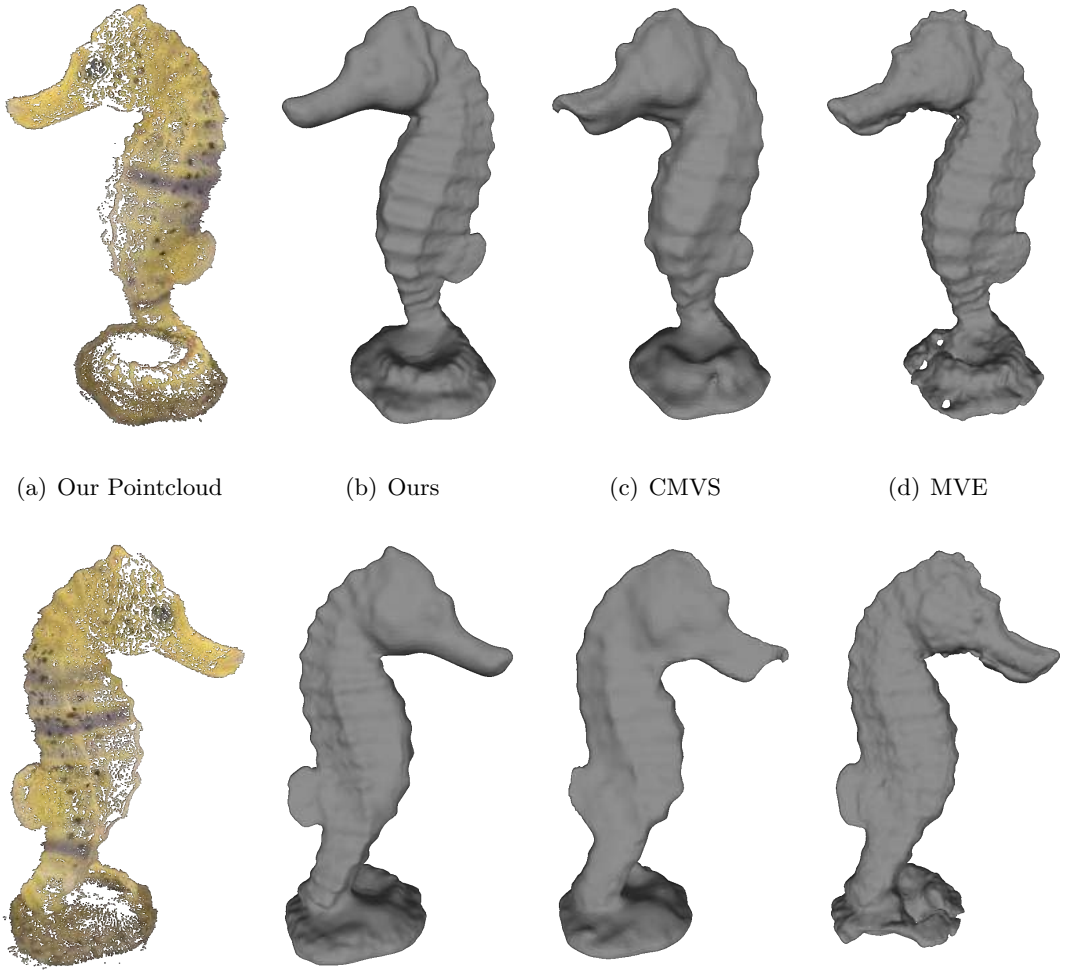


(a) Our Point Cloud (b) Ours (c) CMVS



(d) Our Point Cloud (e) Ours (f) CMVS

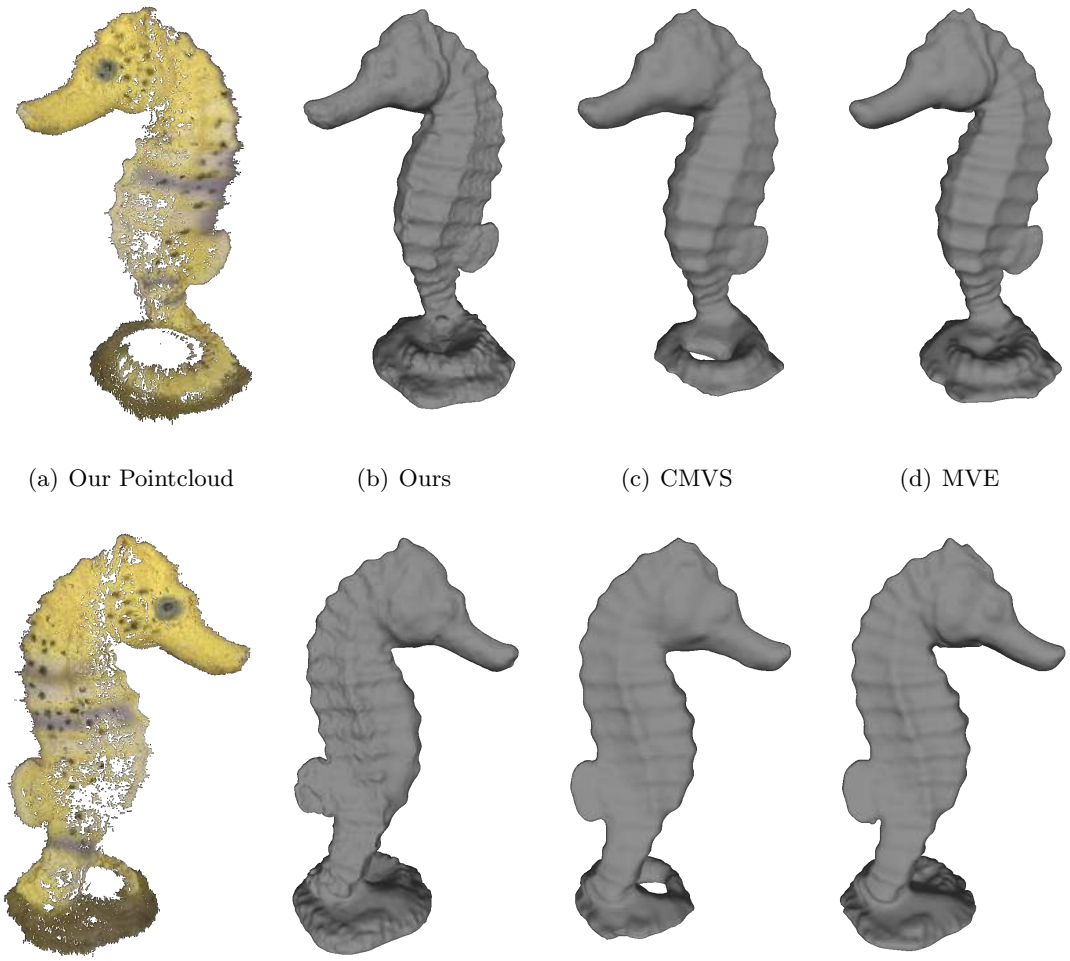
Figure 5.15: CAT dataset acquired with a perspective camera. First view: our point cloud (a), our mesh (b), CMVS (c). Second view: our point cloud (d), our mesh (e), CMVS (f). Note: for this dataset the result from MVE is not available.



(a) Our Pointcloud (b) Ours (c) CMVS (d) MVE

(e) Our Pointcloud (f) Ours (g) CMVS (h) MVE

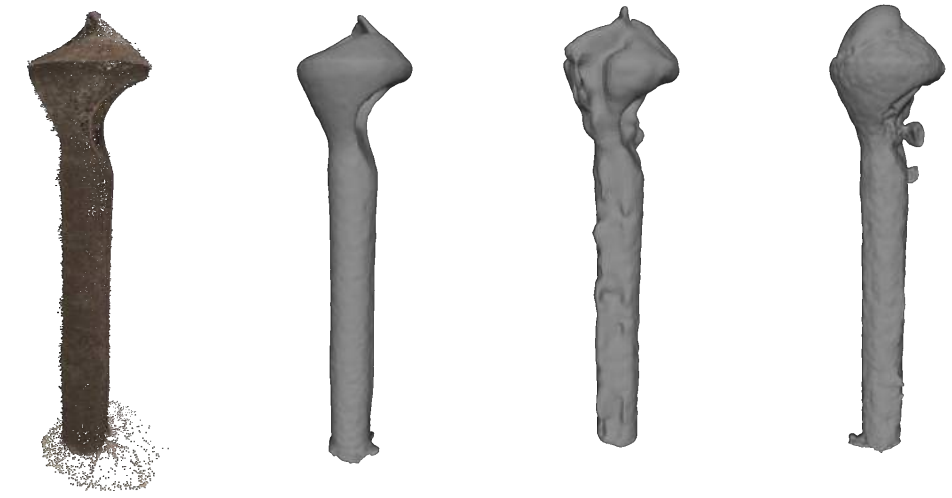
Figure 5.16: SEAHORSE dataset acquired with a telecentric camera. First view: our point cloud (a), our mesh (b), CMVS (c), and MVE (d). Second view: our point cloud (e), our mesh (f), CMVS (g), and MVE (h).



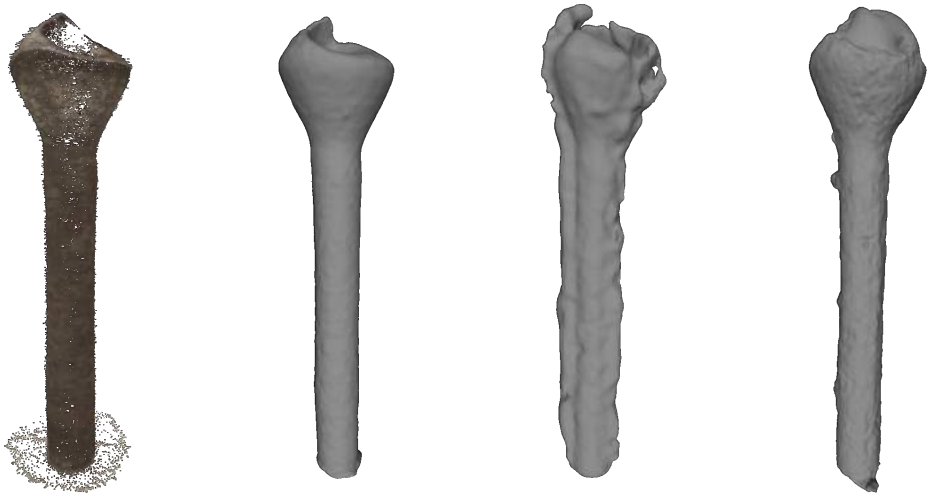
(a) Our Pointcloud (b) Ours (c) CMVS (d) MVE

(e) Our Pointcloud (f) Ours (g) CMVS (h) MVE

Figure 5.17: SEAHORSE dataset acquired with a perspective camera. First view: our point cloud (a), our mesh (b), CMVS (c), and MVE (d). Second view: our point cloud (e), our mesh (f), CMVS (g), and MVE (h).



(a) Ours Pointcloud (b) Ours (c) CMVS (d) MVE



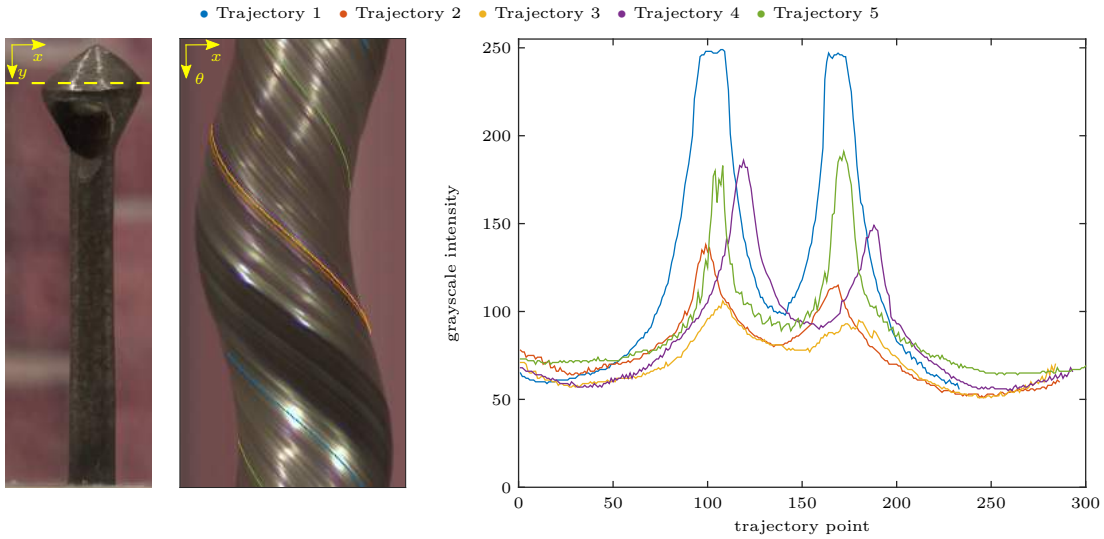
(e) Ours Pointcloud (f) Ours (g) CMVS (h) MVE

Figure 5.18: Drill Bit dataset acquired with a telecentric camera. First view: our point cloud (a), our mesh (b), CMVS (c), and MVE (d). Second view: our point cloud (e), our mesh (f), CMVS (g), and MVE (h).

(a) Our Point Cloud (b) Ours (c) CMVS

(d) Our Point Cloud (e) Ours (f) CMVS

Figure 5.19: Drill Bit dataset acquired with a perspective camera. First view: our point cloud (a), our mesh (b), CMVS (c). Second view: our point cloud (d), our mesh (e), CMVS (f). Note: for this dataset the result from MVE is not available.



(a) (b) (c)

Figure 5.20: Trajectories analysis for the DRILL BIT telecentric dataset: first frame of the light field (a), and the EPI corresponding to the y-coordinate highlighted by the yellow dashed line (the object rotates anticlockwise in front of the camera) (b). Five samples of the founded trajectories are highlighted with different colors. The correspondent grayscale intensity values of these trajectories are plotted in (c). It can be observed that these trajectories have similar intensity behaviors, with two specular peaks, due to the two main illumination units. In particular, *trajectory 1* passes through two strong peaks, which are clearly visible in the EPI, leading to saturation in those areas.

In order to further appreciate the robustness to specular surfaces of the proposed algorithm, Figure 5.20 shows a frame of the circular light field, as well as an EPI. In this image the non-Lambertian effects of the metallic surface are visible, with two clear specular peaks due to the two main lights used to illuminate the scene. This type of data cannot be correctly resolved by classical multi-view algorithms which try to find correspondences between the views, assuming color constancy. However, our approach is able to determine these correspondences in the form of trajectories. Some of these are highlighted in the EPI, and the correspondent intensity values are plotted in Figure 5.20 (c). For simplicity, we report the intensity of the grayscale EPI, rescaled between 0 and 255. Also from this plot the two specular peaks are clearly distinguishable.

5.5 Conclusion

We introduced a novel method to recover precise depth information from circular light fields. Two variants were presented: one for images acquired with telecentric cameras and the other for standard perspective lenses. Differently from classic linear light fields, with circular light fields it is possible to reconstruct the full 360° view of the target scene with just one continuous acquisition. Additionally, they allow retrieving depth even from image sequences acquired with telecentric lenses, a task which is not possible with simple linear motion. In this way, also setups that require telecentric optics can be used to make 3D reconstruction from images without having to change the lens or placing an additional perspective camera. Our method also overcomes the limitation to Lambertian surfaces, which many state-of-the-art algorithms have, by using the Hough transform of binarized EPIs. This leads to a very robust estimation of the EPI-trajectories, which can be found in presence of specular reflections, noise, or even wiggles due to some imprecisions in the calibration or in the rotation mechanism.

We demonstrated the quality of the reconstruction by comparing the resulting meshes of proposed method with two state-of-the-art multi-view algorithms. In case of synthetic datasets, the availability of the ground truth allowed to quantitatively evaluate the results. It was showed that our approach always exceeds the quality of multi-view algorithms, especially in case of non-Lambertian surfaces, were we dramatically outperform the others methods in terms of qualitative and quantitative results. Further experiments with real datasets demonstrated the quality of our reconstructions.

The robustness against specular surfaces makes our circular light field approach suitable to many tasks in industrial optical inspection, where 3D reconstruction of objects with non-Lambertian surface properties is often an issue. Besides 3D reconstruction, another application is material classification based on BRDF estimation. In fact, the intensity variation along each trajectory encodes the material properties of the surface. Therefore, the distribution of these intensities (such as the distributions showed in Figure 5.20 (c)) can be approximated by mathematical models and associated to a specific BRDF. Moreover, the presented setup with a fixed light and circular motion is particularly suited for this type of application. In fact, a rotation is better

than simple linear motion for BRDF reconstruction, since it leads to a larger angular variation between the object and the light source.

In order to further improve the algorithm, it would be worth investigating a possible combination with *photometric stereo* [98]. In fact, at the moment the algorithm is not computing, and therefore using, the surface normals. However, the possibility to handle specular surfaces suggests to exploit this non-Lambertian effects to jointly estimate the geometry and the slope of the surface, similarly to what was done in previous works such as [12, 65, 77], in order to get a denser and even more precise final reconstruction.

Chapter 6

Conclusion

In this thesis we introduced new approaches for 3D reconstruction from light fields video sequences acquired with two different types of motion. Differently from classical multi-view stereo systems, light fields allow to get rid of the correspondence problem, by exploiting the redundancy contained in a densely sampled image sequence, as well as the internal structures of the EPIs, deriving from each specific motion. In the case of linear motion, scene points correspond to lines in the EPI, where the slope of the trajectory encodes the point's distance to the cameras. On the other hand, we have seen that for circular motion the scene's points lead to curved trajectories in the EPI. Variations of the depth lead to sine shaped curves with different amplitudes and phase offsets.

We first described our acquisition setup, composed of a high precision motorized translation stage which is moving a camera with a predefined baseline, to easily acquire linear light fields. On the other hand, circular light fields were acquired by means of a rotation stage, which allows to rotate a target object 360° in front of a fixed camera. We acquired datasets with both a standard and a telecentric lens, to analyse respectively the two cases of perspective and orthographic projection. For both linear and circular light fields, additional synthetic datasets were generated with Blender.

After reviewing the state-of-the-art in linear light field 3D reconstruction algorithms, we introduced a new semi-global approach to estimate lines' orientation in EPIs. The proposed method combines the local slope estimation from the structure tensor with the global information deriving from the Hough transform of the EPI. A quality measure

(i.e. the line score) was introduced to estimate the reliability of the detected lines. We compared our method with different structure tensor approaches, showing that we can achieve better reconstructions, especially in presence of noisy data, where the Hough transform approach is extremely robust. Moreover, our approach leads to better results around depth discontinuities (avoiding edge blurring effects of the local tensor methods), and is able to preserve fine details in the final depth map. Therefore, this approach is particularly suited for datasets with many occlusions and complicate structures.

We then focused on overcoming the biggest limit of linear light fields, namely the fact that only one side of the target object can be reconstructed. To this end, we extended our Hough transform approach to circular light fields, which allow to reconstruct the full 3D object shape with just one continuous acquisition. An additional advantage of circular motion is the possibility to retrieve depth information even from sequences acquired with a telecentric lens. This is particularly useful in applications where this type of optic is needed (e.g. industrial optical inspection), but at the same time one wants to obtain a 3D reconstruction without placing an additional perspective camera. After providing a comprehensive mathematical description of the orthographic and perspective projection models, we present two variants of the algorithm, which can deal with video sequences acquired with respectively telecentric and standard perspective lenses. The output of the algorithms is a point cloud, from which we generate a mesh of the acquired object. The mesh is then compared with state-of-the-art multi-view techniques. In the evaluation we showed that our approach outperforms the multi-view stereo methods for both perspective and orthographic cases. The Hough transform leads to a very robust estimation of the EPI-trajectories, which can be correctly retrieved even in presence of specular reflections.

6.1 Limitations and Future Work

The new Hough transform based method for linear light fields has proved to give better reconstructions than the local structure tensor approaches. It is also robust to noise and allows to obtain very precise depth maps, especially around depth discontinuities. One of the limitations of this algorithm is that the detected lines are based on the EPI-edge map generated with the Canny edge detector. After the reconstruction, this leads to a sparse disparity map. This disparity map could be post-processed by using a

global optimization scheme, such as the second order total variation method proposed by Diebold [28], or by applying bilateral filtering approaches [90, 96]. Moreover, the Hough transform parameter space leads to a discretisation of the disparity values. For this reason the resulting depth maps are composed of many fronto-parallel surfaces, one for each depth level. Also for this issue a filtering procedure to get continuous depth values would be a possible solution. Another improvement could be a better handling of the occlusions, which are currently determined by comparing the slope of the line with the local structure tensor orientation. Instead of this, we could compute the exact occlusions' locations by determining where the EPI-lines intersect. Similar considerations can be applied to the circular light field approach, which is also providing sparse, discretised, but very accurate depth maps. In Chapter 5 it was shown that the Hough transform approach is able to deal with specular objects. These type of surfaces provide an extra information about the shape of the objects. A further development of the algorithm could be exploiting the intensity variation along the trajectories, showed for example in Figure 5.20, and hence combine the surface slope (i.e. surface normals) and geometry (i.e. depth) information in order to generate denser and more precise reconstructions.

Possible applications of these algorithm are all the tasks where an accurate 3D reconstruction is required. One example is industrial optical inspection, where the exact geometry of a production part can be used to decide weather the part is good or defected. Moreover, the robustness to specular surfaces of the Hough transform opens up many possible applications and further developments such as the extraction of the material's BRDF information by analysing the intensity variations along each EPI-trajectory. Another possible application scenario is the movie industry, where accurate depth maps are a fundamental prerequisite for post-production tasks such as background-foreground segmentation.

Appendices

Appendix A

Linear Light Fields

A.1 Hough Transform Parameters

Here we describe the parameters for the Hough transform algorithm for linear light fields:

- Edge scale: the standard deviation of the derivative of Gaussian filter, defined by Equation 3.24, that computes the gradient used by the Canny edge detector.
- Accumulator threshold thr: the value above which an accumulator cell (ρ_i, θ_i) has to lie in order for the corresponding line to be detected.
- Maximum gap $[px]$: the maximum length of gaps in a line. If the gap between two collinear line segments does not exceed this value, then the segments will be merged to a single line.
- Minimum line length $[px]$: lines shorter than this value are discarded.
- Minimum line score: lines having a score below this value are discarded. The line score is computed through Equation 4.5.
- Coherence threshold c_{th}: edge points whose coherence lies below this threshold vote over the entire θ range. Otherwise, they vote over a restricted region, whose size is inversely proportional to the coherence, as defined in Equation 4.4.

A.2 RMSE and BadPix

Here we report the RMSE and BadPix values for the linear light field datasets. The RMSE contains essentially the same information of PSNR and is computed through Equation (4.8). Differently, the BadPix, defined in Equation 5.11, provides the percentage of pixels for which the absolute difference between computed disparity and ground truth is greater than a threshold δ, which we set to $0.05\,px$. Both metrics are scale-dependent, which in our case means that they depend on the disparity range. For this reason, we scale both estimated and ground truth disparity maps so that the range of the ground truth amounts to $2\,px$, i.e. a disparity map is multiplied by $2/(d_+(GT) - d_-(GT))$. This range allows a direct comparison of the RMSE with the theoretical values obtained in Section 4.3.2. The parameters are those used in Section 4.4. For the structure tensor results, the asterisk (*) marks that a coherence threshold of 0.9 has been applied.

Synthetic Buddha

RMSE [px]	Disparity Range [px]		
	1.2	2	4
Classic ST Gauss. Grad.	0.10	0.09	0.08
Classic ST Gauss. Grad.*	0.07	0.06	0.05
Classic ST Scharr*	0.07	0.06	0.05
Classic ST 2.5D*	0.07	0.06	**0.05**
Modified ST*	0.08	0.07	0.07
Ours (41 views)	0.09	0.07	0.07
Ours (101 views)	**0.06**	**0.06**	0.07

BadPix(0.05)	Disparity Range [px]		
	1.2	2	4
Classic ST Gauss. Grad.	7.83	5.68	4.05
Classic ST Gauss. Grad.*	3.02	2.11	1.34
Classic ST Scharr*	3.83	2.53	1.44
Classic ST 2.5D*	1.83	1.48	1.08
Modified ST*	1.86	1.53	1.33
Ours (41 views)	10.67	3.94	2.41
Ours (101 views)	**0.98**	**0.90**	**1.02**

Bronze Man

RMSE $[px]$	Disparity Range $[1.8\,px]$
Classic ST Gauss. Grad.	0.07
Classic ST Gauss. Grad.*	0.05
Classic ST Scharr*	0.04
Classic ST 2.5D*	0.07
Modified ST*	**0.03**
Ours (41 views)	0.04

BadPix(0.05)	Disparity Range $[1.8\,px]$
Classic ST Gauss. Grad.	27.59
Classic ST Gauss. Grad.*	20.60
Classic ST Scharr*	29.34
Classic ST 2.5D*	15.96
Modified ST*	**3.79**
Ours (41 views)	4.55

Clutter

RMSE [px]	Disparity Range [px]	
	2	4
Classic ST Gauss. Grad.	0.17	0.15
Classic ST Gauss. Grad.*	0.11	0.09
Classic ST Scharr*	0.12	0.10
Classic ST 2.5D*	0.10	**0.08**
Modified ST*	0.12	0.10
Ours (41 views)	0.12	0.12
Ours (101 views)	**0.08**	-

BadPix(0.05)	Disparity Range [px]	
	2	4
Classic ST Gauss. Grad.	37.12	28.15
Classic ST Gauss. Grad.*	18.62	12.80
Classic ST Scharr*	21.67	14.17
Classic ST 2.5D*	14.21	**10.91**
Modified ST*	17.78	12.99
Ours (41 views)	14.21	12.49
Ours (101 views)	**8.07**	-

Buddha Head

RMSE [px]	Disparity Range [px]	
	1.7	2.6
Classic ST Gauss. Grad.	0.12	0.11
Classic ST Gauss. Grad.*	0.08	0.07
Classic ST Scharr*	0.09	0.08
Classic ST 2.5D*	0.07	0.06
Modified ST*	**0.06**	**0.05**
Ours (41 views)	0.08	0.07
Ours (101 views)	0.06	-

BadPix(0.05)	Disparity Range [px]	
	1.7	2.6
Classic ST Gauss. Grad.	50.10	47.28
Classic ST Gauss. Grad.*	42.93	38.36
Classic ST Scharr*	47.61	43.11
Classic ST 2.5D*	31.69	28.02
Modified ST*	**29.05**	**24.39**
Ours (41 views)	45.96	40.10
Ours (101 views)	36.65	-

A.3 Disparity Maps

In this section additional disparity maps for the linear light field datasets are presented.

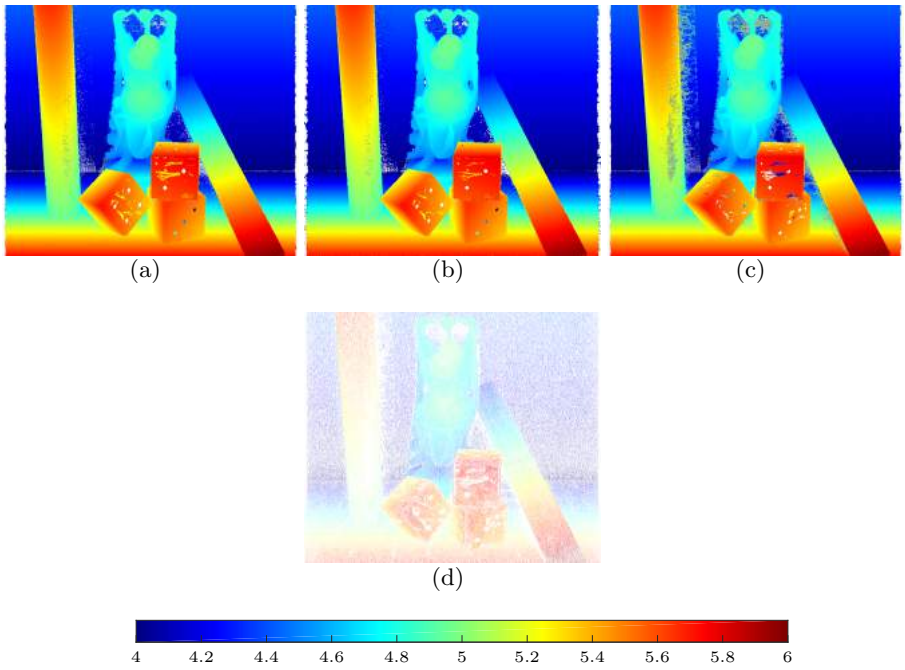


Figure A.1: SYNTHETIC BUDDHA dataset with $\Delta d = 2\,px$: classic structure tensor with Scharr derivative filter (a), classic structure tensor 2.5D (b), modified structure tensor (c), and Hough transform for 101 views (d).

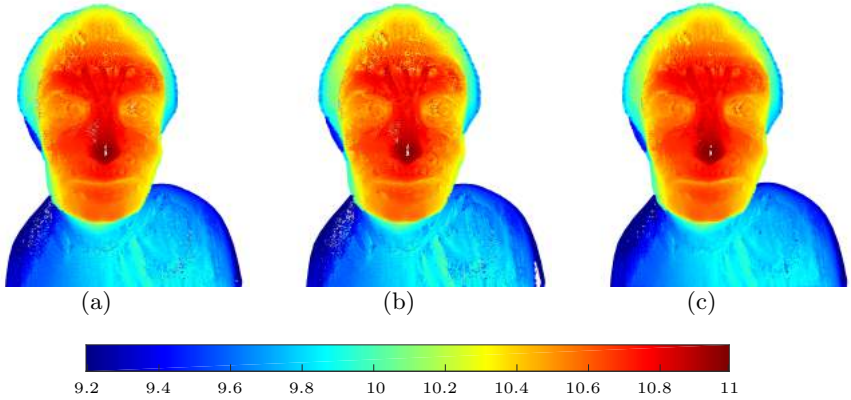


Figure A.2: BRONZE MAN dataset with $\Delta d = 1.8\,px$: classic structure tensor (a), classic structure tensor with Scharr derivative filter (b), and classic structure tensor 2.5D (c).

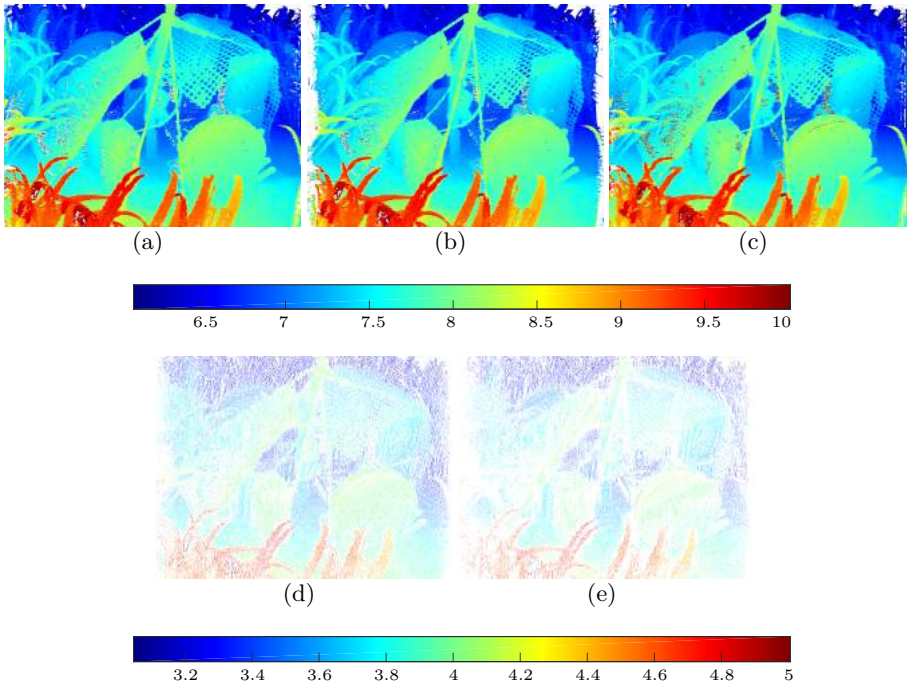


Figure A.3: CLUTTER dataset with $\Delta d = 4\,px$: classic structure tensor (a), classic structure tensor with Scharr derivative filter (b), and modified structure tensor (c). Dataset with $\Delta d = 2\,px$: Hough transform for 41 views (d) and Hough transform for 101 views (e).

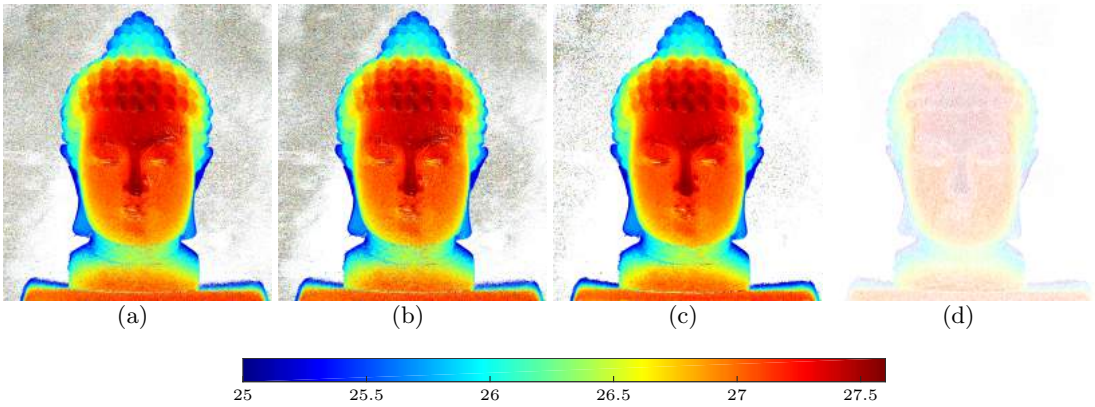


Figure A.4: BUDDHA HEAD dataset with $\Delta d = 2.6\,px$: classic structure tensor (a), classic structure tensor with Scharr derivative filter (b), classic structure tensor 2.5D (c), and Hough transform for 41 views (d).

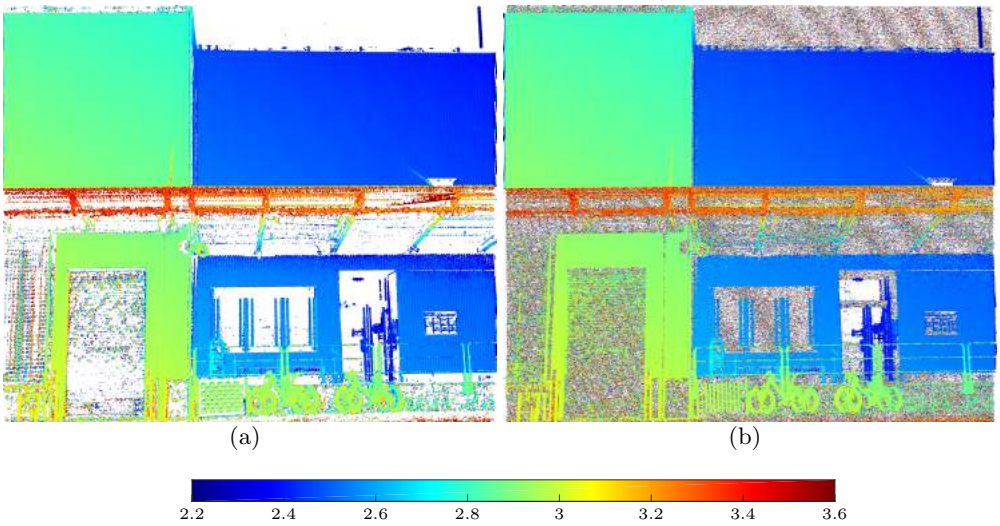


Figure A.5: BACKYARD dataset with $\Delta d = 1.4\,px$: classic structure tensor 2.5D (a) and modified structure tensor (b).

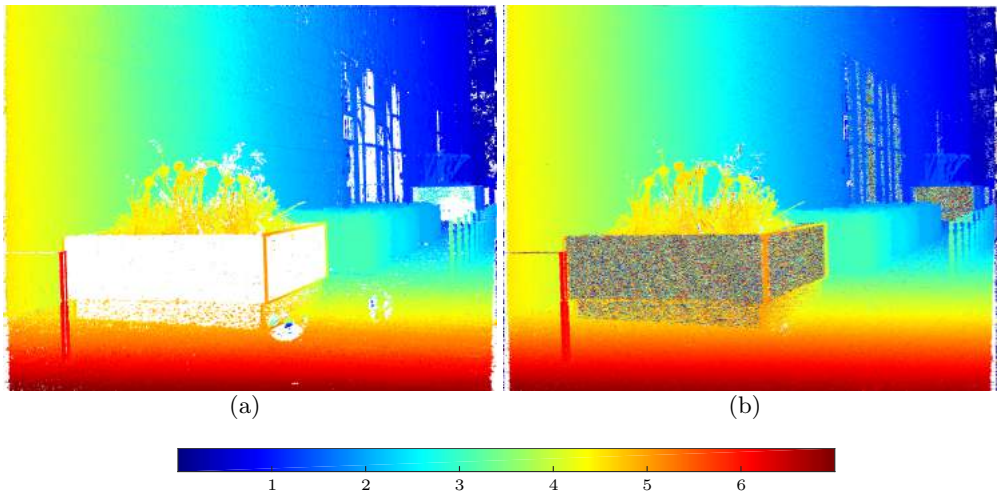


Figure A.6: MATHEMATIKON dataset with $\Delta d = 6.4\,px$: classic structure tensor 2.5D (a) and modified structure tensor (b).

A.4 Point Clouds

To better show the differences between local structure tensor methods and the proposed Hough transform approach, additional point clouds of the reconstructed scenes are presented in the following figures.

(a) (b) (c)

Figure A.7: BRONZE MAN dataset with $\Delta d = 1.8\,px$: classic structure tensor (a), modified structure tensor (b), and hough transform (c). Coherence threshold 0.9 for all the tensor methods.

(a) (b) (c)

Figure A.8: CLUTTER dataset with $\Delta d = 4\,px$: classic structure tensor 2.5D (a), modified structure tensor (b), and hough transform for 41 views (c). Coherence threshold 0.9 for all the tensor methods.

Appendix B

Circular Light Fields

B.1 RMSE and BadPix

In this section we report the RMSE values for the BUDDHA HEAD circular light field dataset. Additionally, we report also the also the BadPix, computed with an error tolerance $\delta = 0.05$ [m].

Dataset	RMSE [%]		
	Ours	CMVS	MVE
Telecentric Lambertian	**0.49**	0.59	0.56
Telecentric specular	**0.45**	0.91	0.72
Perspective Lambertian	**0.54**	0.62	0.59
Perspective specular	**0.56**	0.91	0.81

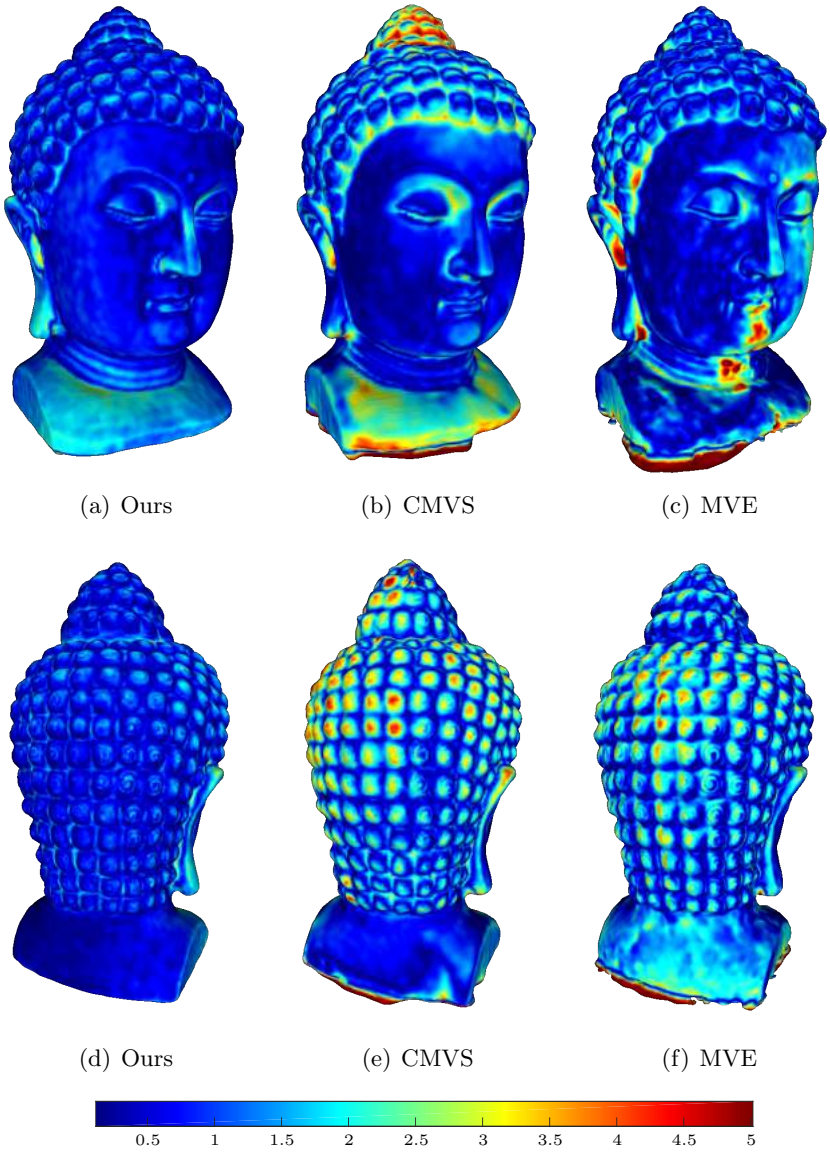
Table B.1: BUDDHA HEAD synthetic datasets: RMSE of the reconstructed meshes in percentage, normalized by the extent of the bounding box.

Dataset	BadPix [%]		
	Ours	CMVS	MVE
Telecentric Lambertian	**1.63**	3.39	3.35
Telecentric specular	**1.58**	16.58	13.29
Perspective Lambertian	**3.04**	4.22	3.76
Perspective specular	**3.08**	20.24	14.34

Table B.2: BUDDHA HEAD synthetic datasets: BadPix 0.05 [m] of the reconstructed meshes.

B.2 Reconstruction Errors

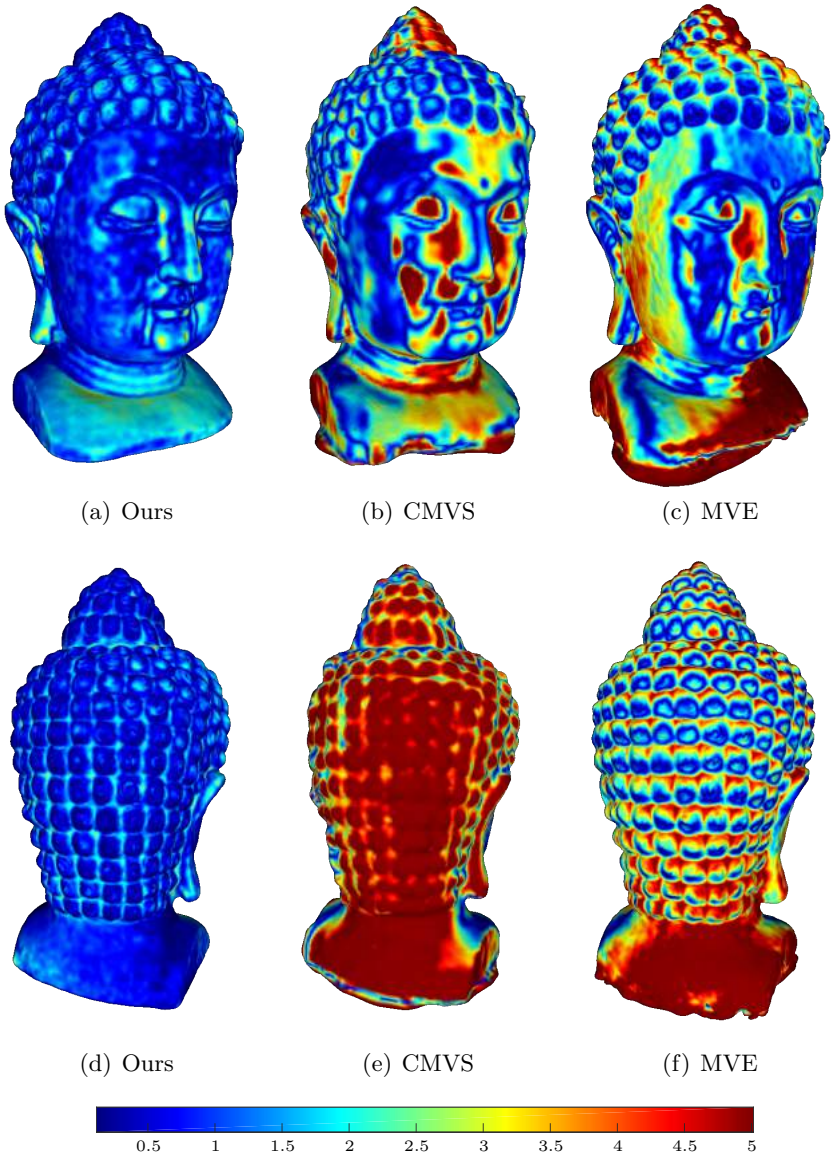
In the following we report a comparison of the reconstruction of the BUDDHA HEAD circular light field dataset. The two the multi-view methods CMVS and MVE are compared with the Hough transform approach by means of meshes where the color highlights the error on millimeters with respect to the structured light ground truth.



(a) Ours (b) CMVS (c) MVE

(d) Ours (e) CMVS (f) MVE

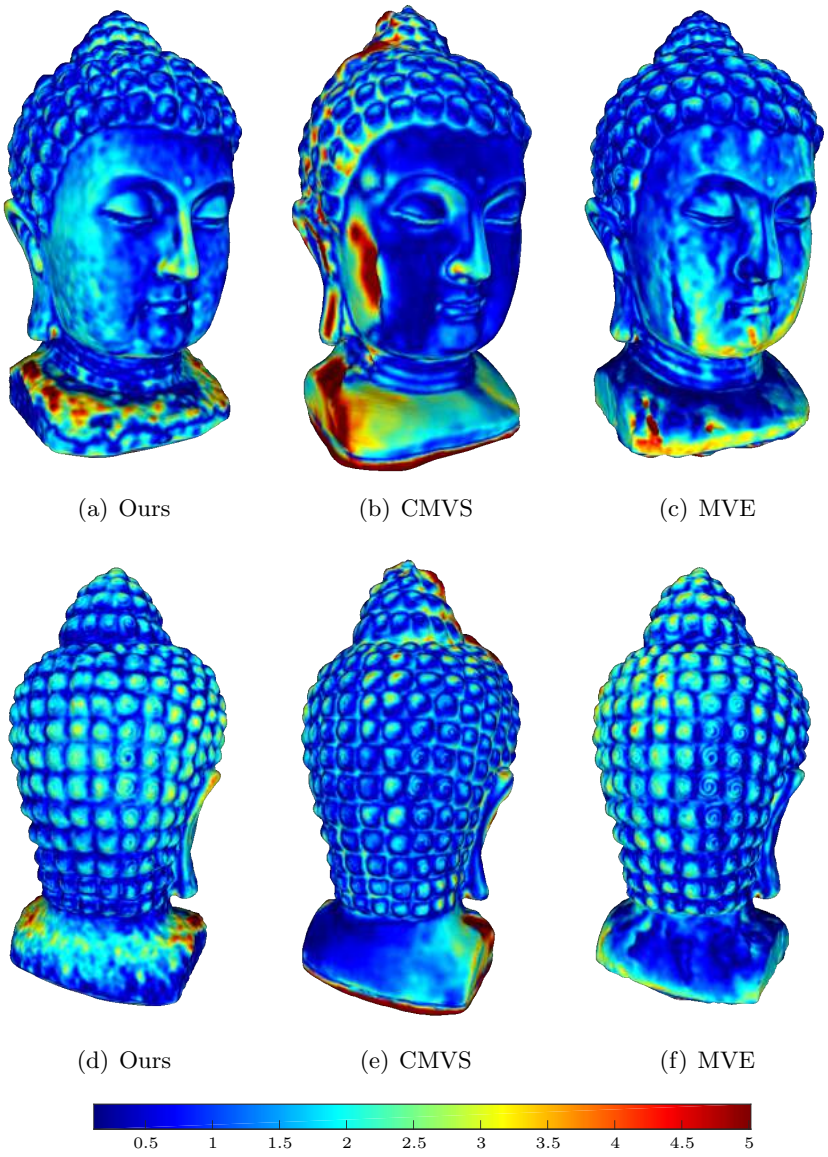
Figure B.1: BUDDHA HEAD Lambertian dataset acquired with a telecentric camera. Comparison of the reconstruction errors in millimeters: ours (a), CMVS (b), MVE (c).



(a) Ours (b) CMVS (c) MVE

(d) Ours (e) CMVS (f) MVE

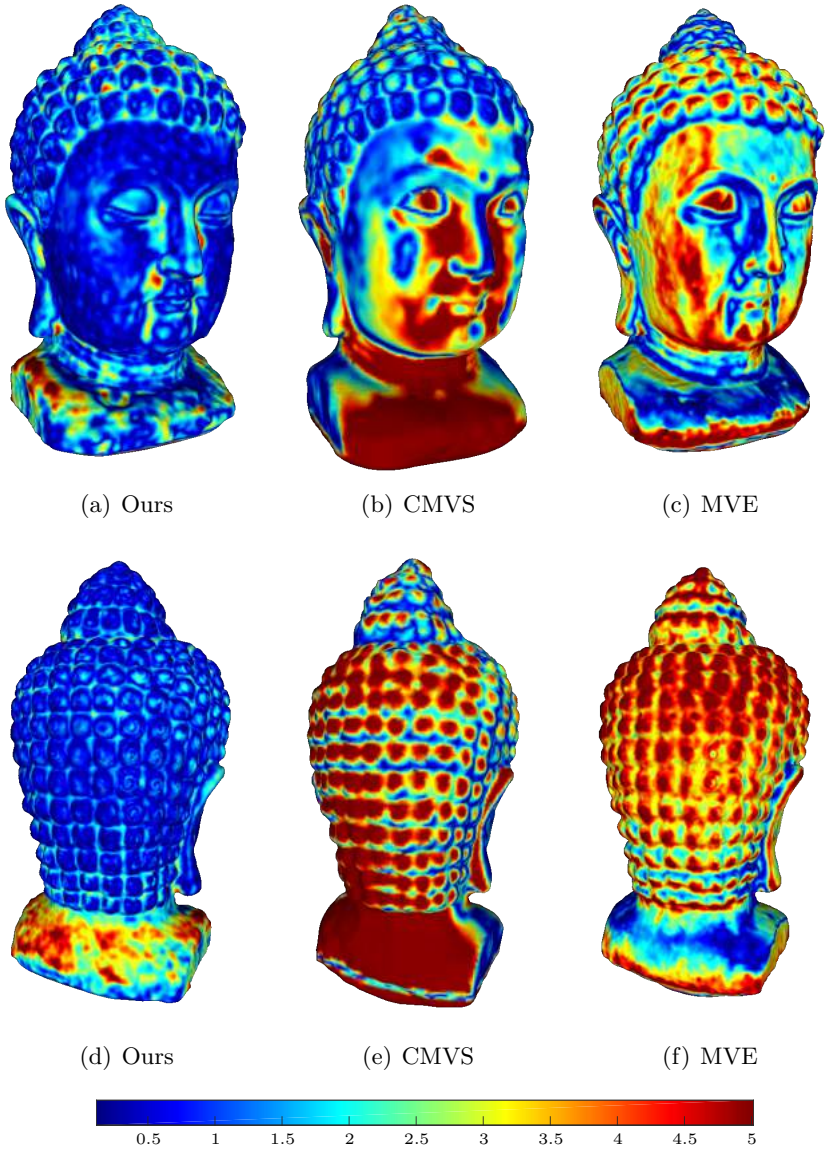
Figure B.2: BUDDHA HEAD specular dataset acquired with a telecentric camera. Comparison of the reconstruction errors in millimeters: ours (a), CMVS (b), MVE (c).



(a) Ours (b) CMVS (c) MVE

(d) Ours (e) CMVS (f) MVE

Figure B.3: BUDDHA HEAD Lambertian dataset acquired with a perspective came. Comparison of the reconstruction errors in millimeters: ours (a), CMVS (b), MVE (c).



(a) Ours (b) CMVS (c) MVE

(d) Ours (e) CMVS (f) MVE

Figure B.4: BUDDHA HEAD Lambertian dataset acquired with a perspective camera. Comparison of the reconstruction errors in millimeters: ours (a), CMVS (b), MVE (c).

Bibliography

[1] *A comparative survey on invisible structured light*, **5303**, 2004. Available from: https://doi.org/10.1117/12.525369. 8

[2] E. H. Adelson and J. R. Bergen. **The plenoptic function and the elements of early vision**. *Computational models of visual processing*, **1**, 1991. 2, 19

[3] E. H. Adelson and J. Y. A. Wang. **Single Lens Stereo with a Plenoptic Camera**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**:99–106, 1992. Available from: https://doi.org/10.1109/34.121783. 23

[4] M. A. Albota, R. M. Heinrichs, D. G. Kocher, D. G. Fouche, B. E. Player, M. E. O'Brien, B. F. Aull, J. J. Zayhowski, J. Mooney, B. C. Willard, and R. R. Carlson. **Three-dimensional imaging laser radar with a photon-counting avalanche photodiode array and microchip laser**. *Appl. Opt.*, **41**(36):7671–7678, December 2002. Available from: https://doi.org/10.1364/ao.41.007671. 10

[5] **Action Recognition and Tracking based on Time-of-Flight Sensors** [online]. March 2017. Available from: http://www.artts.eu/. 9

[6] B. M. Ayyub and R. H. McCuen. *Probability, Statistics, and Reliability for Engineers and Scientists*. CRC Press, 2011. 45

[7] **Basler Time-of-Flight Camera** [online]. March 2017. Available from: https://goo.gl/OyKGnW. 9, 10

[8] P. Besl and N. McKay. **A method for registration of 3D shapes**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**:239–256, 1992. Available from: https://doi.org/10.1109/34.121791. 82

[9] J. Bigün. *Vision with Direction*. Springer, 2006. Available from: https://doi.org/10.1007/b138918. 26, 28

[10] J. Bigün and G. H. Granlund. **Optimal orientation detection of linear symmetry**. In *Proc. International Conference on Computer Vision*, pages 433–438, 1987. 26

[11] E. Binaghi, I. Gallo, G. Marino, and M. Raspanti. **Neural adaptive stereo matching**. *Pattern Recognition Letters*, **25**(15):1743–1758, 2004. Available from: https://doi.org/10.1016/j.patrec.2004.07.001. 14

[12] N. Birkbeck, D. Cobzas, P. Sturm, and M. Jägersand. **Variational Shape and Reflectance Estimation under Changing Light and Viewpoints**. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *9th European Conference on Computer Vision, ECCV 2006, May, 2006*, pages 536–549, Graz, Autriche, May 2006. Lecture Notes in Computer Science, Springer. Available from: https://doi.org/10.1007/11744023_42. 99

[13] T. E. Bishop, S. Zanetti, and P. Favaro. **Light field superresolution**. In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, April 2009. Available from: https://doi.org/10.1109/iccphot.2009.5559010. 2

[14] Blender Foundation. **Blender**. http://www.blender.org/, 2014. 25

[15] R. C. Bolles, H. H. Baker, and D. H. Marimont. **Epipolar-plane image analysis: An approach to determining structure from motion**. *International Journal of Computer Vision*, **1**(1):7–55, 1987. Available from: https://doi.org/10.1007/bf00128525. 3, 21

[16] Y. Boykov and V. Kolmogorov. **An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision**. *IEEE transactions on pattern analysis and machine intelligence*, **26**(9):1124–1137, 2004. Available from: https://doi.org/10.1007/3-540-44745-8_24. 16

[17] Y. Boykov, O. Veksler, and R. Zabih. **Fast Approximate Energy Minimization via Graph Cuts**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(11):1222–1239, 2001. Available from: https://doi.org/10.1109/34.969114. 16

[18] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. **Unstructured Lumigraph Rendering**. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 425–432, New York, NY, USA, 2001. ACM. Available from: https://doi.org/10.1145/383259.383309. 2

[19] B. Büttgen, T. Oggier, M. Lehmann, R. Kaufmann, and F. Lustenberger. **CCD/CMOS Lock-in pixel for range imaging: challenges, limitations and state-of-the-art**. In *In Proceedings of 1st Range Imaging Research Day*, pages 21–32, 2005. 9

[20] J. Canny. **A computational approach to edge detection**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**:679–698, 1986. Available from: https://doi.org/10.1016/b978-0-08-051581-6.50024-6. 36, 76

[21] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. **MeshLab: an Open-Source Mesh Processing Tool**. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. Available from: http://dx.doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136. 83

[22] Wikimedia Commons, 2007. File: Epipolar geometry.svg. Available from: https://goo.gl/IoXFC3. 13

[23] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. **Extracting layers and analyzing their specular properties using epipolar-plane-image analysis**. *Computer vision and image understanding*, **97**(1):51–85, 2005. Available from: https://doi.org/10.1016/j.cviu.2004.06.001. 3, 37

[24] D. Crispell, D. Lanman, P. G. Sibley, Y. Zhao, and G. Taubin. **Beyond Silhouettes: Surface Reconstruction Using Multi-Flash Photography**. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 405–412, jun 2006. Available from: https://doi.org/10.1109/3dpvt.2006.37. 4

[25] L. Da Vinci. **Codex atlanticus**. *Biblioteca Ambrosiana, Milan*, **26**, 1894. 1

[26] A. Davis, M. Levoy, and F. Durand. **Unstructured Light Fields**. *Comput. Graph. Forum*, **31**(2pt1):305–314, May 2012. Available from: https://doi.org/10.1111/j.1467-8659.2012.03009.x. 2, 4

[27] L. Di Stefano, M. Marchionni, and S. Mattoccia. **A fast area-based stereo matching algorithm**. *Image and vision computing*, **22**(12):983–1005, 2004. Available from: https://doi.org/10.1016/j.imavis.2004.03.009. 14

[28] M. Diebold. *Light-Field Imaging and Heterogeneous Light Fields*. PhD thesis, Heidelberg University (Germany), 2016. Available from: https://doi.org/10.11588/heidok.00020560. 22, 32, 55, 103

[29] M. Diebold and B. Goldlücke. **Epipolar Plane Image Refocusing for Improved Depth Estimation and Occlusion Handling**. In *Vision, Modeling and Visualization Workshop VMV*, 2013. Available from: https://doi.org/10.2312/PE.VMV.VMV13.145-152. 33

[30] M. Diebold, B. Jähne, and A. Gatto. **Heterogeneous Light Fields**. In *CVPR*, 2016. Available from: https://doi.org/10.1109/cvpr.2016.193. 3, 32, 46

[31] D. Donatsch, S. A. Bigdeli, P. Robert, and M. Zwicker. **Hand-held 3D Light Field Photography and Applications**. *Vis. Comput.*, **30**(6-8):897–907, June 2014. Available from: https://doi.org/10.1007/s00371-014-0979-5. 4

[32] I. Feldmann, P. Eisert, and P. Kauff. **Extension of Epipolar Image Analysis to Circular Camera Movements**. In *Int. Conf. on Image Processing*, pages 697–700, 2003. Available from: https://doi.org/10.1109/icip.2003.1247340. 4, 72, 74, 81

[33] I. Feldmann, P. Kauff, and P. Eisert. **Optimized Space Sampling for Circular Image Cube Trajectory Analysis**. In *Int. Conf. on Image Processing*, pages 1947–1950, October 2004. Available from: https://doi.org/10.1109/icip.2004.1421461. 4, 81

[34] S. Foix, G. Alenya, and C. Torras. **Lock-in Time-of-Flight (ToF) Cameras: A Survey**. *IEEE Sensors Journal*, **11**(9):1917–1926, Sept 2011. Available from: https://doi.org/10.1109/jsen.2010.2101060. 9, 10

[35] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. **Towards internet-scale multi-view stereo**. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1434–1441. IEEE, 2010. Available from: https://doi.org/10.1109/cvpr.2010.5539802. 17

[36] Y. Furukawa and J. Ponce. **Accurate, Dense, and Robust Multi-View Stereopsis**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. Available from: https://doi.org/10.1109/cvpr.2007.383246. 17, 82

[37] C. Galambos, J. Matas, and J. Kittler. **Progressive Probabilistic Hough Transform for line detection**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, **1**, pages 554–560. IEEE, 1999. Available from: https://doi.org/10.1109/cvpr.1999.786993. 38

[38] A. Gershun. **The Light Field**. *J. Math. and Physics*, **18**:51–151, 1936. Available from: https://doi.org/10.1002/sapm193918151. 2

[39] M. Goesele, N. Snavely, B. Cureless, H. Hoppe, and S. M. Seitz. **Multi-View Stereo for Community Photo Collections**. In *Proc. International Conference on Computer Vision*, pages 265–270, 2007. Available from: https://doi.org/10.1109/iccv.2007.4408933. 17, 82

[40] B. Goldlücke and D. Cremers. **Superresolution Texture Maps for Multiview Reconstruction**. In *Proc. International Conference on Computer Vision*, 2009. Available from: https://doi.org/10.1109/iccv.2009.5459378. 2

[41] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. **The Lumigraph**. In *Proc. SIGGRAPH*, pages 43–54, 1996. Available from: https://doi.org/10.1145/237170.237200. 2, 20

[42] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav. **Depth keying**. *Proc. SPIE*, pages 564–574, 2003. Available from: https://doi.org/10.1117/12.474052. 10

[43] **Halcon** [online]. March 2017. Available from: http://www.mvtec.com/products/halcon/. 26

[44] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Available from: https://doi.org/10.1017/cbo9780511811685.012. 11

[45] H. Hirschmüller. **Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, **2**, pages 807–814. IEEE, 2005. Available from: https://doi.org/10.1109/cvpr.2005.56. 16

[46] P. V. C. Hough. **Method and means for recognizing complex patterns**, December 18 1962. US Patent 3,069,654. 36

[47] **Programming Language C++**. Standard, International Organization for Standardization (ISO), Geneva, Switzerland, 2011. 41

[48] B. Jähne. *Digitale Bildverarbeitung und Bildgewinnung*. Springer, 2012. Available from: https://doi.org/10.1007/978-3-642-04952-1. 11, 29, 30

[49] S. B. Kang and R. Szeliski. **Extracting view-dependent depth maps from a collection of images**. *International Journal of Computer Vision*, **58**(2):139–163, 2004. Available from: https://doi.org/10.1023/b:visi.0000015917.35451.df. 17

[50] M. Kazhdan, M. Bolitho, and H. Hoppe. **Poisson Surface Reconstruction**. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pages 61–70, 2006. 82

[51] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. **Scene Reconstruction from High Spatio-Angular Resolution Light Field**. In *Proc. SIGGRAPH*, 2013. Available from: https://doi.org/10.1145/2461912.2461926. 3

[52] A. Kolb, E. Barth, and R. Koch. **ToF-sensors: New dimensions for realism and interactivity**. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, June 2008. 10

[53] R. E. Krebs. *Groundbreaking scientific experiments, inventions, and discoveries of the Middle Ages and the Renaissance*. Greenwood Publishing Group, 2004. 1

[54] D. Lanman, D. C. Hauagge, and G. Taubin. **Shape from depth discontinuities under orthographic projection**. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1550–1557, Sept 2009. Available from: https://doi.org/10.1109/iccvw.2009.5457427. 4

[55] Z. Lee, J. Juang, and T. Q. Nguyen. **Local disparity estimation with three-moded cross census and advanced support weight**. *IEEE Transactions on Multimedia*, **15**(8):1855–1864, 2013. Available from: https://doi.org/10.1109/tmm.2013.2270456. 14

[56] F. Lenzen, F. Becker, and J. Lellmann. **Adaptive Second-Order Total Variation: An Approach Aware of Slope Discontinuities**. In *Proceedings of the 4th International Conference on Scale Space and Variational Methods in Computer Vision SSVM*, **7893** of *LNCS*, pages 61–73. Springer, 2013. 1. Available from: https://doi.org/10.1007/978-3-642-38267-3_6. 16

[57] F. Lenzen, H. Schäfer, and C. Garbe. **Denoising Time-Of-Flight Data with Adaptive Total Variation**. In *Proceedings ISVC*, pages 337–346. Springer, 2011. Available from: https://doi.org/10.1007/978-3-642-24028-7_31. 16

[58] M. Levoy and P. Hanrahan. **Light field rendering**. In *Proc. SIGGRAPH*, pages 31–42, 1996. Available from: https://doi.org/10.1145/237170.237199. 2

[59] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, Tokyo, 1995. Available from: https://doi.org/10.1007/978-4-431-66933-3. 16

[60] G. Lippmann. **Épreuves réversibles donnant la sensation du relief**. In *J. Phys. Theor. Appl.*, pages 821–825, 1908. Available from: https://doi.org/10.1051/jphystap:019080070082100. 23

[61] **Lytro Inc.** [online]. March 2017. Available from: https://store.lytro.com/. 24

[62] G. Manfredi. *Depth Estimation from 3D Light Fields*. Master's thesis, Heidelberg University, 2016. 35

[63] **MESA Imaging SR4000** [online]. March 2017. Available from: http://hptg.com/industrial/. 9

[64] S. K. Nayar and M. Gupta. **Diffuse structured light**. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, April 2012. Available from: https://doi.org/10.1109/iccphot.2012.6215216. 9

[65] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. **Efficiently Combining Positions and Normals for Precise 3D Geometry**. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 536–543, New York, NY, USA, 2005. ACM. Available from: http://doi.acm.org/10.1145/1186822.1073226. 99

[66] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. **Light field photography with a hand-held plenoptic camera**. Technical Report CSTR 2005-02, Stanford University, 2005. 2, 23

[67] A. S. Ogale and Y. Aloimonos. **Shape and the stereo correspondence problem**. *International Journal of Computer Vision*, **65**(3):147–162, 2005. Available from: https://doi.org/10.1007/s11263-005-3672-3. 14

[68] **Owis Limes 170-600-HiSM** [online]. March 2017. Available from: https://goo.gl/pZwWd5. 26

[69] **Owis DTM 130N** [online]. March 2017. Available from: https://goo.gl/IDMrfi. 26, 88

[70] K. Papafitsoros and C. B. Schönlieb. **A Combined First and Second Order Variational Approach for Image Reconstruction**. *J. Math. Imaging Vis.*, **48**(2):308–338, February 2014. Available from: https://doi.org/10.1007/s10851-013-0445-4. 16

[71] **pco.edge 5.5** [online]. March 2017. Available from: https://goo.gl/ltkn93. 26, 88

[72] D. Piatti. *Time-of-Flight cameras: tests, calibration and multi-frame registration for automatic 3D object reconstruction.* PhD thesis, Polytechnic University of Turin (Italy), 2010. 10

[73] **PMD CamCube 3.0** [online]. March 2017. Available from: https://goo.gl/F9nT6P. 9

[74] **Raytrix GmbH** [online]. March 2017. Available from: http://www.raytrix.de/. 24

[75] A. Rochas, M. Gosch, A. Serov, P. A. Besse, R.S. Popovic, T. Lasser, and R. Rigler. **First fully integrated 2-D array of single-photon detectors in standard CMOS technology**. *IEEE Photonics Technology Letters*, **15**(7):963–965, 2003. Available from: https://doi.org/10.1109/lpt.2003.813387. 10

[76] W. Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance.* Springer, 1996. Available from: https://doi.org/10.1007/bfb0015091. 83

[77] R. Sabzevari, V. Murino, and A. Del Bue. **PiMPeR: Piecewise Dense 3D Reconstruction from Multi-View and Multi-Illumination Images**. *CoRR*, **abs/1503.04598**, 2015. Available from: http://arxiv.org/abs/1503.04598. 99

[78] J. Salvi, J. Pags, and J. Batlle. **Pattern Codification Strategies in Structured Light Systems**. *Pattern Recognition*, **37**:827–849, 2004. Available from: https://doi.org/10.1016/j.patcog.2003.10.002. 8

[79] B. Schäling. *The Boost C++ Libraries*. XML Press, 2011. 41

[80] H. Scharr. *Optimale Operatoren in der Digitalen Bildverarbeitung*. PhD thesis, Heidelberg University, 2000. Available from: https://doi.org/10.11588/heidok.00000962. 31

[81] D. Scharstein, R. Szeliski, and R. Zabih. **A taxonomy and evaluation of dense two-frame stereo correspondence algorithms**. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001. Available from: https://doi.org/10.1109/smbv.2001.988771. 83

[82] M. Schmidt and B. Jähne. **A Physical Model of Time-of-Flight 3D Imaging Systems, Including Suppression of Ambient Light**. In *Proceedings of the DAGM 2009 Workshop on Dynamic 3D Imaging*, Dyn3D '09, pages 1–15, Berlin, Heidelberg, 2009. Springer-Verlag. Available from: http://dx.doi.org/10.1007/978-3-642-03778-8_1. 9, 10

[83] S. Seitz and C. Dyer. **Photorealistic scene reconstruction by voxel coloring**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, June 1997. Available from: https://doi.org/10.1109/cvpr.1997.609462. 17

[84] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. **A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, **1**, pages 519–528, 2006. Available from: https://doi.org/10.1109/cvpr.2006.19. 17

[85] N. Snavely, S. M. Seitz, and R. Szeliski. **Modeling the World from Internet Photo Collections**. *Int. J. Comput. Vision*, **80**(2):189–210, 2008. Available from: https://doi.org/10.1007/s11263-007-0107-3. 17

[86] J. Sun, N.-N. Zheng, and H.-Y. Shum. **Stereo matching using belief propagation**. *IEEE Transactions on pattern analysis and machine intelligence*, **25**(7):787–800, 2003. Available from: https://doi.org/10.1109/tpami.2003.1206509. 16

[87] S. Sylwan. **The application of vision algorithms to visual effects production**. In *Asian Conference on Computer Vision*, pages 189–199. Springer, 2010. Available from: http://dl.acm.org/citation.cfm?id=1964320.1964340. 16

[88] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. **Using plane + parallax for calibrating dense camera arrays**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2004. Available from: https://doi.org/10.1109/cvpr.2004.1315006. 24

[89] A. Vianello. *Depth Super-Resolution with Hybrid Camera System*. Master's thesis, Department of Information Engineering, University of Padova (Italy), 2013. Available from: https://goo.gl/JbOSOL. 10

[90] A. Vianello, F. Michielin, G. Calvagno, P. Sartor, and O. Erdler. **Depth images super-resolution: An iterative approach**. In *Int. Conf. on Image Processing*, pages 3778–3782. IEEE, Oct 2014. Available from: https://doi.org/10.1109/icip.2014.7025767. 10, 103

[91] S. Štolc, D. Soukup, B. Holländer, and R. Huber-Mörk. **Depth and all-in-focus imaging by a multi-line-scan light-field camera**. *Journal of Electronic Imaging*, **23**(5):053020–053020, 2014. Available from: https://doi.org/10.1117/1.jei.23.5.053020. 3

[92] S. Wanner. *Orientation Analysis in 4D Light Fields*. PhD thesis, Heidelberg University (Germany), 2014. Available from: https://doi.org/10.11588/heidok.00016439. 21, 33, 46

[93] S. Wanner, J. Fehr, and B. Jähne. **Generating EPI representations of 4D light fields with a single lens focused plenoptic camera**. *Advances in Visual Computing*, pages 90–101, 2011. Available from: https://doi.org/10.1007/978-3-642-24028-7_9. 3

[94] S. Wanner and B. Goldlücke. **Globally consistent depth labeling of 4D light fields**. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 41–48. IEEE, 2012. Available from: https://doi.org/10.1109/cvpr.2012.6247656. 31, 32

[95] S. Wanner and B. Goldlücke. **Spatial and angular variational super-resolution of 4D light fields**. In *Proc. European Conference on Computer Vision*, 2012. Available from: https://doi.org/10.1007/978-3-642-33715-4_44. 2

[96] O. Wasenmüller, G. Bleser, and D. Stricker. **Combined Bilateral Filter for Enhanced Real-time Upsampling of Depth Images.** In *VISAPP (1)*, pages 5–12, 2015. Available from: https://doi.org/10.5220/0005234800050012. 103

[97] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. **High performance imaging using large camera arrays**. *ACM Transactions on Graphics*, **24**:765–776, July 2005. Available from: http://doi.acm.org/10.1145/1186822.1073259. 24, 25

[98] R. J. Woodham. **Photometric method for determining surface orientation from multiple images**. *Optical engineering*, **19**(1):191139–191139, 1980. Available from: https://doi.org/10.1117/12.7972479. 99

[99] K. Yücer, C. Kim, A. Sorkine-Hornung, and O. Sorkine-Hornung. **Depth from Gradients in Dense Light Fields for Object Reconstruction**. In *Proceedings of International Conference on 3D Vision (3DV)*, 2016. Available from: https://doi.org/10.1109/3dv.2016.33. 4

[100] K. Yücer, A. Sorkine-Hornung, O. Wang, and O. Sorkine-Hornung. **Efficient 3D Object Segmentation from Densely Sampled Light Fields with Applications to 3D Reconstruction**. *ACM Transactions on Graphics*, pages 22:1–22:15, March 2016. Available from: https://doi.org/10.1145/2876504. 4

[101] C. Zach, K. Karner, and H. Bischof. **Hierarchical disparity estimation with programmable 3D Hardware**. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, **22**(12):275–282, 2004. 14

[102] **Zeiss Makro-planar 2/100 ZF.2** [online]. March 2017. Available from: https://goo.gl/Odbb2K. 88

[103] **Zeiss Visionmes 105/11** [online]. March 2017. Available from: https://goo.gl/q5jjyB. 88

[104] A. Zelinsky. **Learning OpenCV—Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf]**. *IEEE Robotics Automation Magazine*, **16**(3):100–100, September 2009. Available from: https://doi.org/10.1109/mra.2009.933612. 15

[105] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. **High-quality Video View Interpolation Using a Layered Representation**. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 600–608. ACM, 2004. Available from: https://doi.org/10.1145/1015706.1015766. 17