Dissertation

submitted

to the

Combined Faculty for the Natural Science

and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Science

submitted by

Dipl.-Ing. Maximilian Diebold

born in Karlsruhe

Oral examination: 18.04.2016

Light-Field Imaging
and
Heterogeneous Light Fields

Supervisor:	Prof. Dr. Bernd Jähne
Second Assessor:	Prof. Dr. Karl-Heinz Brenner

Zusammenfassung

Bei der traditionellen Lichtfeldauswertung hat jedes Bild gleichen spektralen Inhalt, was zu konstanten Intensitätswerten in der *Epipolar Plane Image* (EPI) Mannigfaltigkeit führt. Diese Art von Lichtfeld wird auch *homogenes Lichtfeld* genannt.

Im Gegensatz dazu unterscheiden sich *Heterogene Lichtfelder* darin, dass die einzelnen Bilder verschiedene Eigenschaften besitzen, wie beispielsweise unterschiedliche Luminanzen oder durch verschieden Spektralfilter unterschiedlichen spektralen Inhalt.

Um heterogene Lichtfelder auswerten zu können, wird eine entsprechende Methode zur Berechnung von Orientierungen in heterogenen EPIs benötigt. Dazu werden die einzelnen Komponenten der Strukturtensoranalyse in Bezug auf ihre Funktion analysiert und alternative Methoden zur Orientierungsanalyse wie beispielsweise die *singular value decomposition* analysiert. Schlussendlich führen diese Analysen zu neuen Konzepten, die den Strukturtensoransatz verbessern, wodurch dessen Genauigkeit gesteigert und eine Anwendbarkeit auf heterogene Lichtfelder möglich wird. Während der aktuelle Strukturtensor nur Orientierungen mit konstanter Pixelintensität entlang der Orientierungsrichtung schätzen kann, ist der neu entworfene Strukturtensor in der Lage auch Orientierungen mit sich ändernden Intensitätsstrukturen zu schätzen. Zusätzlich wird es aufgrund einer viel höheren Robustheit gegen Belichtungsschwankungen möglich, aufgenommene Lichtfelder mit einer viel höheren Zuverlässigkeit auszuwerten.

Um das volle Potential dieses verbesserten Stukturtensors zu nutzen, ist es wichtig das Lichtfeldkamerasetup so anzupassen, dass die Szene perfekt in den ±45° Orientierungbereich passt. Diese Voraussetzung führt zu einer direkten Verbindung zwischen dem Lichtfeldkamerasetup und dem wie ein Pyramidenstumpf geformten relevanten Messraumvolumen.

Wir zeigen, dass hochpräzise Tiefenkarten berechenbar werden, was einen positiven Einfluss auf die Güte nachfolgender Prozessierungen hat und besonders bei der sRGB Farbrekonstruktion in Lichtfeldern mit unterschiedlich spektral gefilterten Bereichen zu sehen ist. Zusätzlich wird ein *Global Shifting* entwickelt, was eine Überschreitung der zugrundeliegenden Limitierung des ±45° Orientierungsbereichs ermöglicht, um somit einen größeren Tiefenbereich schätzen zu können und eine zusätzliche Steigerung der Präzision zu erreichen. Hierdurch wird zudem ermöglicht spherische Lichtfelder zu untersuchen, bei denen der ±45° Orientierungsbereich regelmäßig überschritten wird. Die Forschung an spärischen Lichtfeldern war in enger Zusammenarbeit mit dem "German Center for Artificial Intelligence (DFKI)" in Kaiserslautern.

Summary

In traditional light-field analysis, images have matched spectral content which leads to constant intensity on *epipolar plane image* (EPI) manifolds. This kind of light field is termed *homogeneous light field.*

Heterogeneous light fields differ in that contributing images may have varying properties such as exposure selected or color filter applied. To be able to process heterogeneous light fields it is necessary to develop a computation method able to estimate orientations in heterogeneous EPI respectively. One alternative method to estimate orientation is the *singular value decomposition.* This analysis has resulted in new concepts for improving the structure tensor approach and yielded increased accuracy and greater applicability through exploitation of heterogeneous light fields. While the current structure tensor only estimates orientation with constant pixel intensity along the direction of orientation, the newly designed structure tensor is able to estimate orientations under changing intensity. Additionally, this improved structure tensor makes it possible to process acquired light fields with a higher reliability due to robustness against illumination changes.

In order to use this improved structure tensor approach, it is important to design the light-field camera setup that the target scene covers the ±45° orientation range perfectly. This requirement leads directly to a relationship between camera setup for light-field capture and the frustum-shaped volume of interest.

We show that higher-precision depth maps are achievable, which has a positive impact on the reliability of subsequent processing methods, especially for sRGB color reconstruction in color-filtered light fields.

Aside this, a *global shifting* process is designed to overcome the basic range limitation of ±45° to estimate larger distances and to increase additionally the achievable precision in light-field processing. That enables the possibility to research spherical light fields, since the orientation range of spherical light fields typically overcomes the ±45° limit. Research in spherically acquired light fields has been conducted in collaboration with the German Center for Artificial Intelligence (DFKI) in Kaiserslautern.

Acknowledgment

My very special thanks to Prof. Dr. B. Jähne for supervising my thesis and giving me the opportunity to do my PhD in image processing. Furthermore I owe particular thanks to Dr. A. Gatto who was supervising me and supporting me in the best possible way during my time as PhD student. But I also like to thank my colleagues at the University particularly Priv.-Doz. Dr. Garbe and Dr. Wanner.
Further thanks to B. Krolla for the cooperation and shared research in spherical light fields. Last but not least special thanks to Dr. H. Baker for all the informative discussions and for the proofreading of my thesis.

Contents

1 Introduction

Light-field imaging has established itself over the past few years as promising new research field in computer vision. Its areas of application are wide spread and reach from viewpoint interpolation [42, 60, 8] through super-resolution [90, 28] to refocusing [63]. Still, light-field imaging's most basic and perhaps most important application may be in depth estimation. Due to the steady improvements in modern computing, it is now possible to process the huge amount of data in a light field in seconds or less. Light-field imaging itself has a lot of advantages in comparison with other ranging methods. It provides the estimates of scene depth without invoking the matching that causes uncertainties in binocular and multi-view stereo algorithms. Because of this, it is faster in estimating depth and makes it possible to gather much more information from the captured scene.

We want to begin by introducing alternate methods for estimating scene geometry, highlighting the advantages and disadvantages of the different methods. Then, we will detail what of light-field analysis brings to the problem and explain the contributions of this thesis.

1.1 Alternate Ranging Methods

There are several major differing approaches to obtain depth information from a scene. They can be split into two main classes. The first class are passive methods using the environmental light to obtain depth information. Associated methods are binocular or multi-view stereo algorithms. The second class uses active illumination like a fringes

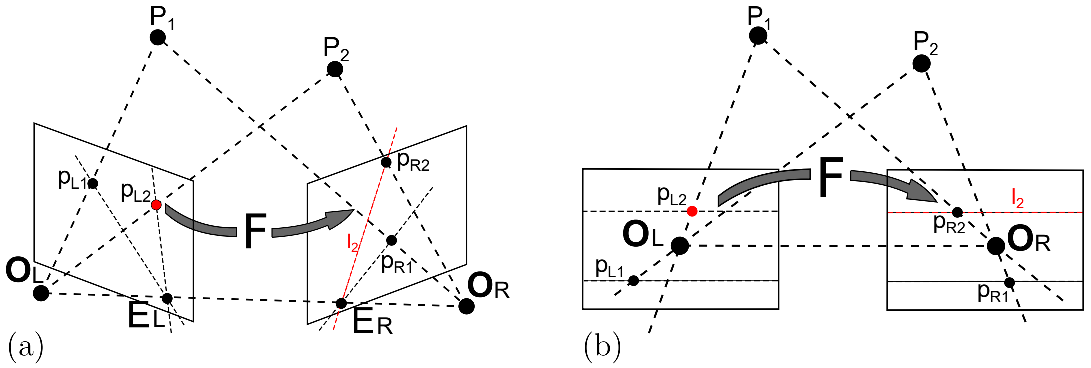


Figure 1.1: *(a) illustrates the epipolar geometry for verged cameras. The epipoles E_l and E_R are located inside the image. All epipolar lines intersect in the epipole. The epipolar lines can be determined by the fundamental matrix and a reference point in the other image. (b) shows a stereo camera setup with two parallel looking cameras. As one can see the epipolar lines are horizontally aligned. Thus the epipoles are located at infinity.*

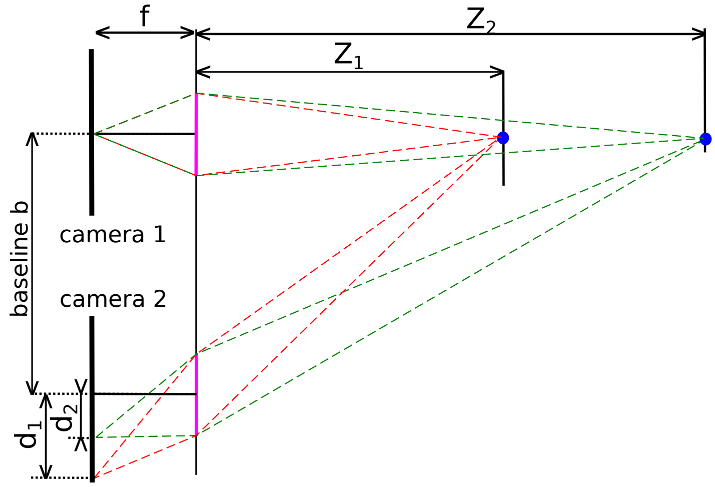


Figure 1.2: *Illustrates the imaging of two points located at different depths Z_1 and Z_2. In the first camera both points are imaged onto the same position. In the second camera both points are imaged onto a different position. The distance between both projection points is termed disparity d_1 and d_2.*

projector, laser light or coherent light. The most common methods for active and passive illumination are detailed in the following and its advantages and disadvantages are highlighted.

Stereo imaging

One of the first established methods to estimate range is termed stereo imaging or triangulation. This method is based on the binocular vision [36] of humans and other animals which use visual information derived from their two eyes to determine binocular disparities [9, 72] and hence depth. While the perception of binocular disparity occurs naturally when viewing a scene, this can also be obtained artificially through the separate presentation of two different images to each eye using stereoscopic display methods, such as in a Viewmaster or, more currently, in the Oculus Rift [21].
To estimate disparities, stereo triangulation methods exploit the epipolar geometry to efficiently determine correspondence of points in the two images. The parallax observed between corresponding points is the disparity, and this corresponds inversely to distance from the viewer. To find correspondences, the fundamental matrix F [57, 33] is used to compute epipolar lines in the second camera with respect to a reference point in the first camera, as illustrated in figure 1.1 (a).
All such determined scan lines intersect in a mathematical object termed the epipole, which is defined as the intersection of the line joining the two camera's centers of projection and the imaging plane. Knowing this epipole is essential to understand the camera geometry, and thus for doing image triangulation. A special stereo setup with parallel-directed cameras orthogonal to their baseline has its epipoles at infinity, as shown in figure 1.1 (b). All epipolar lines in this configuration are horizontally

Figure 1.3: *Shown are different models of time-of-flight cameras. (a) Argos 3D-P100 with up to 160 fps and* 160×120 *pixels [6], (b) PMDs CamCube with* 204×204 *pixels and SBI [67], (c) SwissRanger 4000 of MESA Imaging having* 176×144 *pixels [81] and (d) TOF-camera having* 176×144 *pixels*

aligned, and this simplifies the correspondence search to image rows only. Imagery in this form, after lens distortion correction, is termed *rectified*. In light-field imaging, rectified images are mandatory. To extract epipolar-plane images (EPIs) which are first defined by H. Baker [7], all cameras need to be located at a common camera plane having parallel viewing direction and epipoles at infinity. How to extract EPIs out of the lumigraph, we detail in section 2.1. To determine scene depth we define a stereo camera setup as shown in figure 1.2. Both cameras are assumed to have the same focal length f. The distance between the cameras is termed their baseline b. The difference of the relative projection of a world point P is termed disparity d. Thus resulting depth can be computed by

$$Z = \frac{fb}{d} \tag{1.1}$$

which shows that the disparity is inversely proportional to the distance Z of the object [38].

Stereo algorithms can estimate disparity fairly reliably in regions where there are no specularities or occlusions. In regions with low contrast or with high sensor noise, however, most implementation have difficulties. Additionally, many stereo implementations only possess a discretized resolution which results in visible depth steps in 3D reconstructions. To achieve a sub-pixel accuracy requires more effort and analysis.

Time-of-Flight cameras

Time-of-flight cameras (ToF cameras) are systems that measure distance based on the speed of light and the time taken for signal return. LIDAR (Light Detection And Ranging) is a form of ToF sensing, where one measures distance by illuminating a target with a laser and analyzing the reflected light. Sensors in ToF cameras are more complex than sensors in normal passive cameras. Each pixel needs to be able to determine independently the time from signal emission to reception. To accommodate this, the

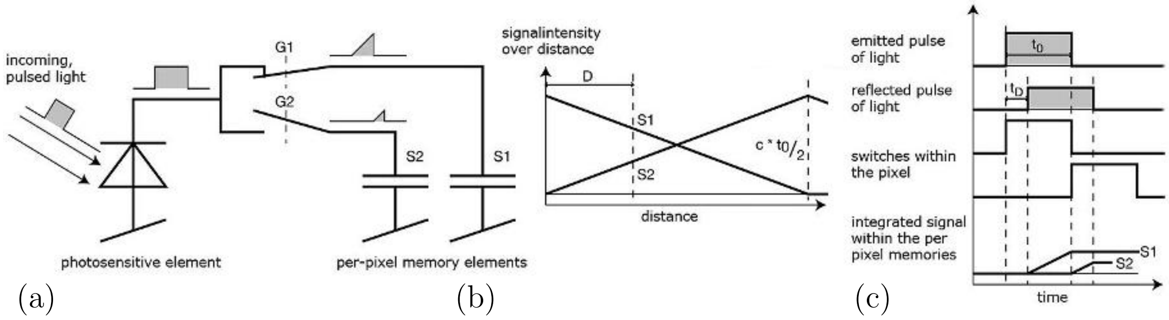


Figure 1.4: *Shows the distance measurement of time-of-flight cameras [84]. The switches G_1 and G_2 trigger the pixel memory elements S_1 and S_2 as shown in (a). While the reflected light arrives with a time delay t_D the memory elements fill with respect to the delay respectively (b). The ratio between S_1 and S_2 define the distance D of the object. The overall measurement is shown in (c).*

pixels of ToF cameras need to be larger – as much as 10 times the size in normal CCD cameras – often about 10 μm. Most commercially available ToF cameras achieve sensor resolutions of about 320 x 240 pixels [78] or less. Furthermore it is only possible to determine depth information of materials able to reflect the incident laser light frequency toward the emitting source (the scatter that this necessitates is part of the reason why ToF sensing elements must be large). Also multi path propagation or interference between two ToF cameras can lead to ghosting effects – ambiguity in distance arises since the time for signal return may be incorrect. ToF cameras are able to estimate distances from a few decimeters to about tens of meters with a depth resolution of about 1 cm. The main advantages of these systems are their high frame rate (about 160 frames per second) and lack of need for any sort of the signal matching involved with passive range correspondence-based ranging. To determine the depth, each ToF camera has two storage elements S_1 and S_2. These elements are alternately triggered at the same frequency as the emitted light pulse. Due to the time delay t_d of the returned light, the ratio between the collected signal in S_1 and S_2 changes with respect to the underlying distance as shown in figure 1.4. Thus the depth can be computed by

$$D = \frac{c_{\text{air}} t_0}{2} \cdot \frac{S_2}{S_1 + S_2} \tag{1.2}$$

where c_{air} is the speed of light in air.

Interferometry

This technique exploits the interference ability of light and extracts information about its wave character [17]. It is an important method in fields of fiber optics, optical metrology, remote sensing, surface profiling, velocimetry [65] and many more. Interferometry typically uses a single light source emitting coherent light. The emitted light enters a

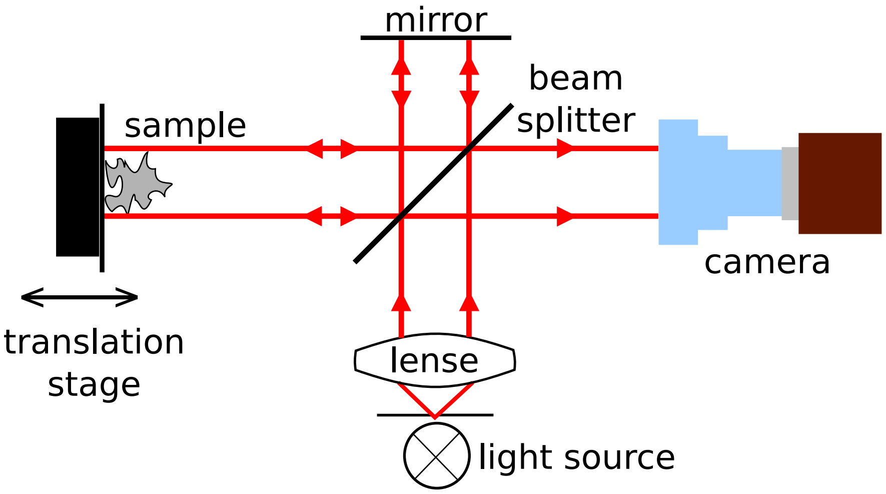


Figure 1.5: *Demonstrates the mechanism of interferometry. The emitted light enters a beam splitter which separates the wave in a reference wave, which gets reflected at a mirror and in a second wave, which hits the target scene. The reflected wavefronts of both directions interfere while entering the camera's lens. The camera captures the interference pattern.*

beam splitter which splits the wave into two identical beams as shown in figure 1.5. While one beam is directed toward a mirror to act as reference wave, the second beam is reflected toward the target surface which modifies the wavefront. After entering the camera the reflected wavefront undergoes constructive or destructive interference with the reference wave. Due to the appearing phase difference of both waves, the distance can be measured by the phase difference itself or by the resulting intensity deviation of the mixed signal. The depth resolution of this method depends on the wavelength λ employed, the image distance with respect to the back principal plane d and the diameter of the exit pupil l. Thus it can be expressed for air environments as

$$\delta_z = 4\,\frac{\lambda d^2}{l^2}. \tag{1.3}$$

In contrast the spatial resolution is limited by the Rayleigh criterion [73]. The minimal resolvable separation termed Rayleigh distance δ [29] is therefore

$$\delta_x = 1.22\,\frac{\lambda d}{l}. \tag{1.4}$$

It is not effective to estimate objects larger than the wavelength of the light employed. Additionally, thermal expansion and mechanical disturbances need to be considered.

Structured Light

This method uses active illumination to recover scene geometry. Here, the target is illuminated with a known pattern, such as fringes or grids. To project such patterns, two methods have been established. The first method uses laser interference and the

second uses non-coherent light, which means commercial video projectors can be used. Laser interference methods allow very fine patterns and have an unlimited depth of field, but there are difficulties in providing the ideal beam geometry. Additional problems appear due to speckle, noise or self interference with reflected beams.
In contrast, the projection method commonly uses fringe patterns which deform on striking the surface, and whose analysis allows to estimate depth and surface information. The basic principle of this method is shown in figure 1.6. Typical measuring devices consist of one stripe projector and at least one camera, while a second camera on the opposite side saves the calibration of the gamma curve of the projector and has been established as useful. To triangulate reliably, a linear mapping of the light intensities between camera and projector is necessary. Thus with two identical cameras a significant effort of the gamma calibration is saved. Unfortunately this method is very sensitive to ambient light, performing best in darkened rooms. Additionally it cannot be used to scan shiny surfaces or objects with many intricate details where the patterning becomes too complex for analysis.

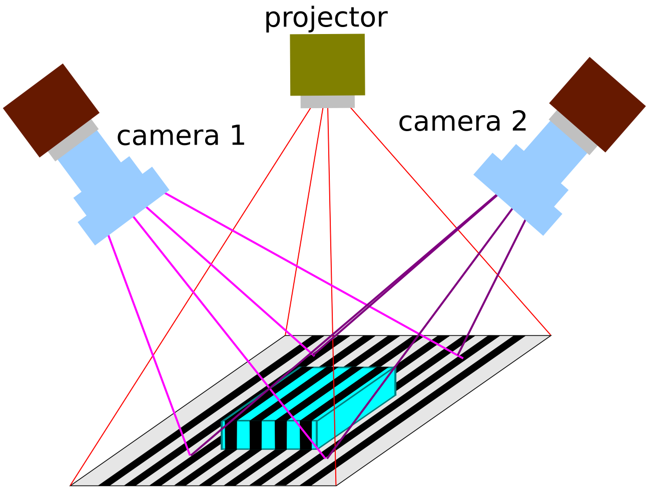


Figure 1.6: *Illustrates a structured light setup. It consists of two opposing cameras and one central fringe pattern projector. The depth information is encoded in the displacement of the fringes projected onto the surface of the objects.*

1.2 Motivation

In contrast to other depth sensing methods, light-field imaging provides a new class to analyze 3D geometry of a captured target scene. Light fields can be acquired by moving a single camera on a regular grid either horizontally, vertically or in both directions. Using a single camera restricts the acquisition to static scenes, while using camera arrays allow the capturing of dynamic scenes and light-field movies. In addition to spatial scene information, light fields capture angular information due to the different camera positions. That idea was first described in 1908 by Gabriel Lippmann who termed it

Integral Photography [52]. The term light field was introduced by A. Gershun [27] in 1936. After an additional 60 years, light-field imaging was reformulated as a display technology in computer graphics by Gortler *et al.* [30] and Levoy *et al.* [49]. The first applications in light-field imaging focused on view interpolation because light fields yield the possibility of generating novel views without explicitly knowing the 3D geometry [45]. Nevertheless light-field imaging provides the possibility to determine scene geometry due to the available angular information.

Additionally, light fields contains information about the Bidirectional Reflectance Distribution Function (BRDF). The BRDF describes the angular dependent intensity change of the reflected light on opaque surfaces with respect to the incidence angle and the surface normal. While in reality almost all materials have an angular dependent BRDF (these are termed Non-Lambertian surfaces), only a few materials exist with pure Lambertian properties (i.e. constant BRDF).

Stereo algorithms suffer from Non-Lambertian surfaces and are not able to find correspondences when the intensity of two corresponding points changes significantly due to the differing view perspectives. Fortunately, with just a small angular deviation between two viewpoints, the assumption of constant BRDF (the Lambertian assumption) is generally valid. In case of larger angular deviation it is possible to use multi-view stereo approaches which provide more correspondences and make it possible to analyze also Non-Lambertian surfaces, however algorithms used in this become more and more complex [39, 40].

In contrast, light fields not only allow estimation of the scene geometry but also provide the ability to analyze the BRDF. Aside from this, as is shown in this thesis, light fields are not restricted to image elements that retain color or intensity distribution across views. It is possible to capture light fields, while each image has slightly different properties such as different exposure values or applied color filters. This opens new possibilities in analyzing scenes: for example, hyper-spectral information can be extracted, or one-shot high-dynamic-range (HDR) images become achievable without applying complex algorithms. Due to the nearly continuous disparity space, we can not only attain higher quality image information, but also better surface normals for use in describing the underlying 3D geometry.

Contribution: The purpose of this thesis is to present an analysis of heterogeneous light fields. Thus an improved Structure Tensor is introduced which not only estimates homogeneous light field orientation with higher confidence, but is also applicable to the analysis of heterogeneous light fields. While the traditional Structure Tensor fails totally in processing heterogeneous light fields, the mechanism we adapt – the Structure Tensor – still achieves good results. To prove the robustness of the new Structure Tensor against changing illumination, we introduce and analyze illumination gradient light fields. Aside from this, we also process color-filtered light fields and use them to obtain dense disparity maps. The resulting disparity maps of color-filtered light fields are used to reconstruct a hyper-spectral image for each light field with respect to a

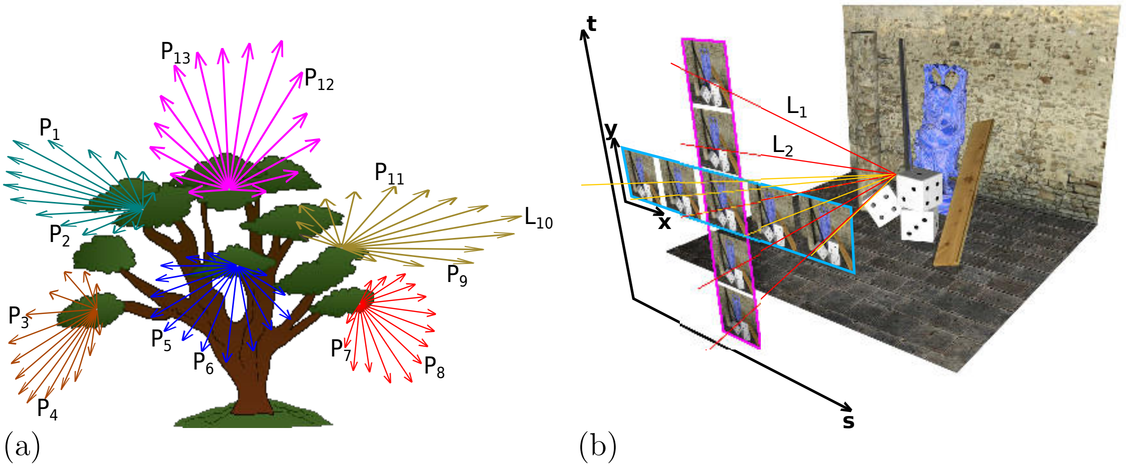


Figure 1.7: *(a) illustrates that incident light becomes scattered in all possible directions by hitting the surface of objects. Each reflected ray can be described by the plenoptic function P. (b) visualizes the lumigraph representation L of a light field.*

reference view. These hyper-spectral images are later mapped to the sRGB space.

In addition to the analysis of heterogeneous light fields, we perform an accuracy and precision analysis of the newly developed Structure Tensor. By these analyses it has become possible to determine requirements on the camera setup and the target scene for obtaining high quality depth reconstructions.

Further, we present analysis of spherical light fields, through a cooperation with the DFKI in Kaiserslautern. Initial steps are also presented in the analysis of temporal light fields.

1.3 Light-Field definition

In considering the light field, we first need to think about light within a ray representation. Light is filling space, in all directions, with rays of various intensities. These rays spread without interfering with each other while traveling independently though space. Light hitting an object surface at position (X_w, Y_w, Z_w) becomes scattered and reflects in a pencil of rays from the object surface. The reflection direction of each ray of the pencil as seen in figure 1.7 (a) is describable by (Θ, ϕ) . This model of light traveling through space is described by the plenoptic function.

The plenoptic function was introduced by Adelson and Bergen [2] in 1991. The resulting parametrization for a specific wavelength λ of the light to a given time t is described by

$$P = P(\Theta, \phi, \lambda, t, X_w, Y_w, Z_w) \tag{1.5}$$

This parametrization defines a light field in a very detailed manner, yet an even higher dimensional description is possible, including factors such as polarization and the light incident angle. This high dimensional representation is, unfortunately, not suitable for use in computational imaging. With modern cameras only capturing a discrete number of wavelengths in bands such as red, green and blue, or in a monochromatic integrated band, we may ignore wavelength. Furthermore, through considering only static scenes, we may ignore the temporal variation and consider the time component as constant. Thus the plenoptic function becomes a manageable version with reduced dimensionality.

$$P = P(\Theta, \phi, X_w, Y_w, Z_w) \tag{1.6}$$

This parametrization is simpler but still ill-suited for computer graphics. In computer graphics images are parametrized in the (x, y) image space. Thus the plenoptic function is represented by the lumigraph, as introduced by Gortler *et al.* [30]. The lumigraph parametrizes the light field with respect to camera position (s, t) and pixel location (x, y) as visualized in figure 1.7 (b). The light-field representation becomes

$$(\Theta, \phi, X_w, Y_w, Z_w) \to (s, t, x, y) \tag{1.7}$$
$$L(s, t, x, y) := P(\Theta, \phi, X_w, Y_w, Z_w). \tag{1.8}$$

With this assumption all cameras are considered as located on a common plane with parallel viewing direction. This implies that all epipoles are located at infinity, facilitating the extraction of epipolar-plane images [7] from the captured data. A detailed explanation of the Lumigraph and how to slice out EPIs is given in chapter 2.

1.4 Light-Field Acquisition

The lumigraph enables the acquisition of light fields using two different methods. The first method uses cameras having a micro lens array in front of the image sensor as realized by T. Georgiev [23], T. Lumsdaine [56] and C. Perwass [66]. These kinds of cameras are also termed plenoptic cameras. The principal behind plenoptic cameras

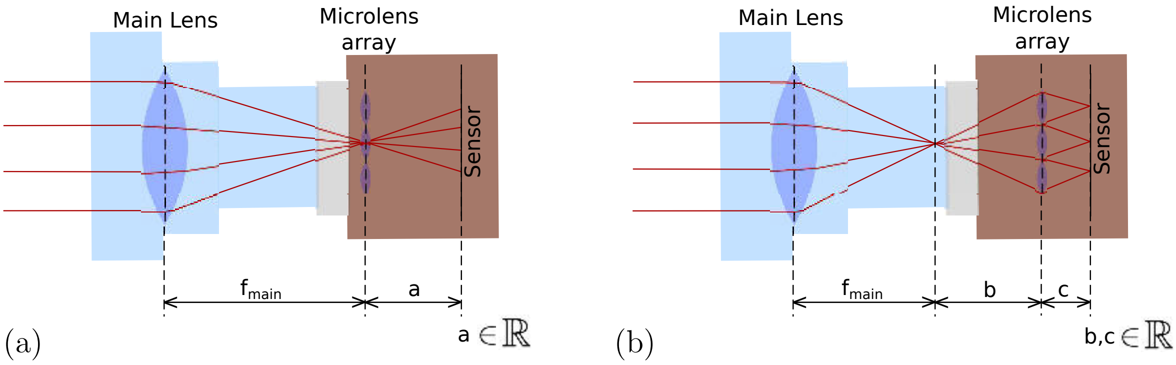


Figure 1.8: *(a) shows the principle of a plenoptic camera. The micro-lens array is located at the focus point of the main lens. (b) illustrates the focused plenoptic camera. The micro-lens array is located behind the focal point of the main lens. Thus, the resulting image contains hundreds of micro images.*

is that incident light becomes separated through positioning of a micro lens array before the image sensor, enabling the acquisition of angular information in addition to the obvious spatial chrominance information. A disadvantage of this sort of plenoptic camera is the loss of spatial resolution due to separation across angular distributions. The distinction behind the plenoptic camera [63] and what is called the focused plenoptic camera [24, 25] is shown in figure 1.8 (a). The plenoptic camera has its micro lens array directly at the focal point of the main lens with the micro lenses focused at infinity. Thus only the same surface point is split into its angular components which is spatially imaged onto the sensor chip. The final resolution is through this determined by the number of micro lenses. That means, the resulting image has a very low spatial resolution but possesses depth information. In contrast, the focused plenoptic camera has the micro lens array placed behind the main lens focus point as shown in figure 1.8 (b). The micro lenses now satisfy the lens equations and display focused micro images onto the sensor chip. This design allows to acquire images which trades angular resolution for higher spatial resolution. Thus the number of micro lenses and the number of achievable resolution is decoupled which is a significant improvement [55]. Unfortunately, the lower angular resolution can lead to undesired aliasing artifacts.

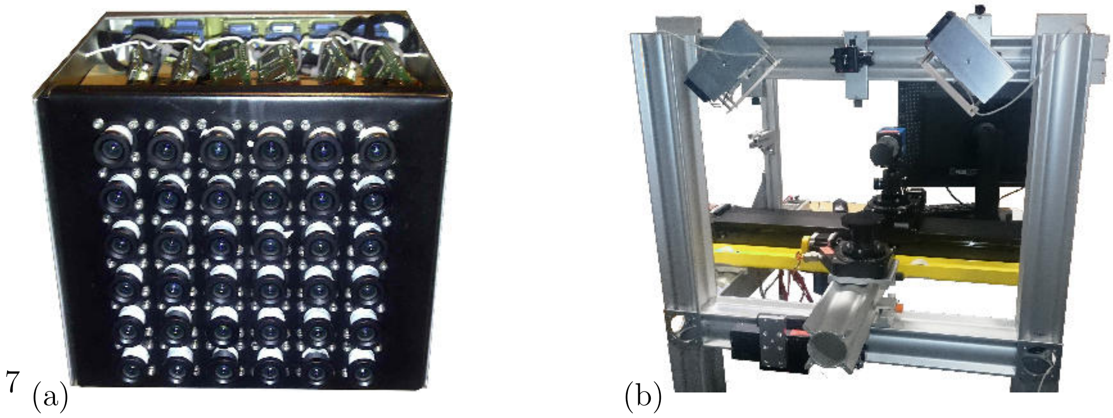


Figure 1.9: *(a) shows a camera array, having cameras located on a regular grid structure. It consists of* 36 *cameras, mounted in a* 6×6 *structure. (b) shows a translation-stage setup with mounted camera. This setup is well-suited to acquire high quality light fields of close objects.*

The second method to acquire light fields uses arrays of discrete cameras (camera arrays) to capture light fields as shown in figure 1.9 (a). An advantage of this method is that the single images have a much higher resolution than plenoptic cameras. Furthermore, no special optics are necessary which means that any commercially available camera is a suitable component. Willburn *et al.* [94] present a camera array consisting of 100 cameras mounted on a regular grid in a 10×10 structure to capture scenes from different directions in a single shot. With either approach – plenoptic capture or camera-array capture – high precision calibration of the system is indispensable. Additionally, the localization and consistency of the imaging elements is of great consequence, as described by Y.Xu *et al.* [95]. Cameras with large differences in their characteristics decrease the possibility of high precise measurements. The size of each single camera also influences the process, as it established the minimal separation possible across the array. Densely sampled light-field imaging of nearby objects is clearly a challenge due to the mechanical restrictions presented by current capture systems. To overcome this restriction in the density of imaging, it is possible to use a single camera mounted on a translation stage as demonstrated by V.Vaish *et al.* [89], C.Kim *et al.* [43] and J.Unger *et al.* [87]. Precision placement makes it possible to capture dense and high-precision light fields and reduces the complexity in calibrating the system – eliminating the need for extrinsic analysis (aside from modeled orientation errors), and leaving only the intrinsic of the single camera to be defined. All acquired light fields presented in this thesis were obtained with an Owis Limes 170 high-precision translation stage and an sCMOS PCO edge 5.5 USB 3.0 camera as shown in figure 1.9 (b). The digitally synthesized light fields were rendered with Blender [13] using the developed light-field toolbox described in the Appendix D.

2 Orientation Estimation

2.1 The lumigraph light-field representation

A light field is defined, as in the lumigraph [30], by two parallel planes Π and Ω. The Ω-plane addresses the coordinates $(x, y) \in \Omega$ of image observations and the Π-plane defines the location of the focal points $(s, t) \in \Pi$ of each camera. A 4D color light field can thus be defined as

$$L : \Omega \times \Pi \to \mathbb{R} \qquad (s, t, x, y) \mapsto L(s, t, x, y), \tag{2.1}$$

where $L(s, t, x, y)$ defines the pixel intensity value of the ray defined (x, y) in the image plane and (s, t) in the focal plane. 2D slices from L are termed epipolar-plane images [14] (EPIs) Σ. In a 4D light field two different epipolar-plane images can be extracted as illustrated in figure 2.2. The first relates to the horizontal camera direction and the second to the vertical camera direction. Epipolar-plane images related to the horizontal camera direction are described by the equation

$$S_{t^*,y^*} : \Sigma_{t^*,y^*} \to \mathbb{R} \tag{2.2}$$
$$(x, s) \mapsto S_{t^*,y^*}(x, s) \quad := \quad L(s, t^*, x, y^*) \tag{2.3}$$

where t and y take the values t^* and y^* to obtain the EPI. In contrast, EPIs related to the vertical camera direction s and x take the values s^* and x^*. The addressed EPIs are described by the equation

$$S_{s^*,x^*} : \Sigma_{s^*,x^*} \to \mathbb{R} \tag{2.4}$$
$$(y, t) \mapsto S_{s^*,x^*}(y, t) \quad := \quad L(s^*, t, x^*, y). \tag{2.5}$$

In the following computations, vertical and horizontal EPIs are computed identically. For ease of review, all further equations will be expressed with respect to horizontal EPIs S_{t^*,y^*}, abbreviated with S.

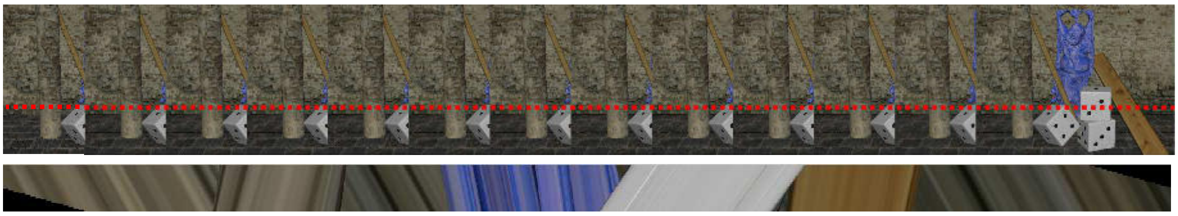


Figure 2.1: *Shows a light field consist of* 13 *images. An EPI, addressed at the red horizontal line, is shown below. It contains depth dependent orientation information of each captured object.*

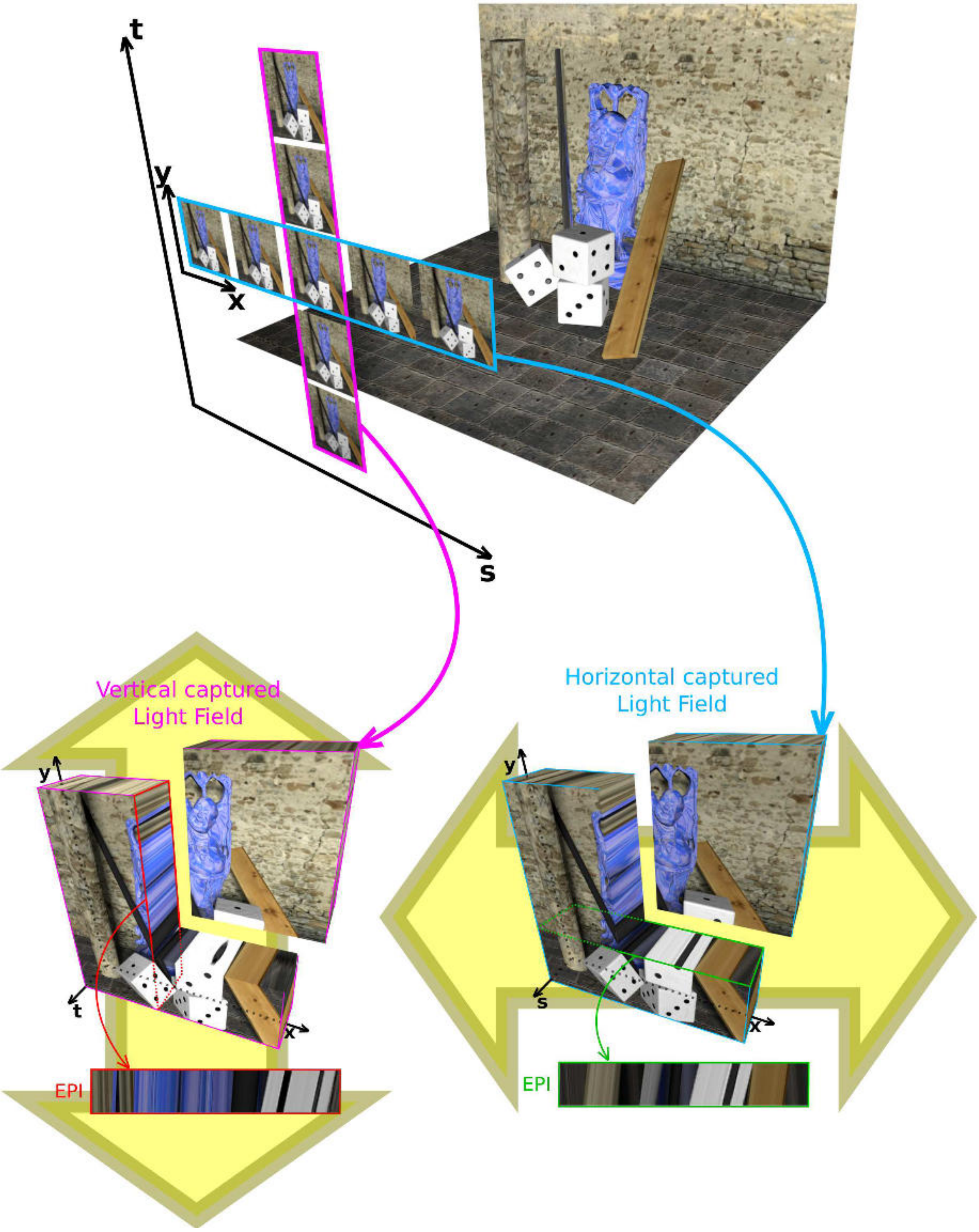


Figure 2.2: *Representation of a cross light field and its representation as 3D image volumes with respect to the capture direction. In a horizontal light field case, EPIs are achieved by slicing horizontally the image volume. In the vertical case EPIs are achieved by slicing vertically the image volume.*

2.2 2D Structure Tensor

To estimate the disparity maps of a given scene using EPIs, one has to estimate the underlying orientations that relate to the scene geometry. Thus estimating orientations For this, we use the 2D Structure Tensor, which determines the underlying orientation in the 2D epipolar-plane images, see figure 2.1. The 2D Structure Tensor J is defined as:

$$J = \tau * \begin{pmatrix} \left(\frac{\partial \hat{S}}{\partial x}\right)^2 & \frac{\partial \hat{S}}{\partial x} \cdot \frac{\partial \hat{S}}{\partial s} \\ \frac{\partial \hat{S}}{\partial s} \cdot \frac{\partial \hat{S}}{\partial x} & \left(\frac{\partial \hat{S}}{\partial s}\right)^2 \end{pmatrix} =: \begin{pmatrix} J_{xx} & J_{xs} \\ J_{xs} & J_{ss} \end{pmatrix} \tag{2.6}$$

with the abbreviation

$$\hat{S} := \sigma * S, \tag{2.7}$$

where σ defines the inner Gaussian smoothing of the EPI and τ the outer Gaussian smoothing, applied on the Structure Tensor components. To achieve a disparity map d all EPIs in the light field must be processed. The Structure Tensor is applied on each EPI independently, as shown in figure 2.3 (a). The disparity itself is computed by

$$d = \tan\left(\frac{1}{2}\arctan\left(\frac{2J_{xs}}{J_{xx} - J_{ss}}\right)\right) \tag{2.8}$$

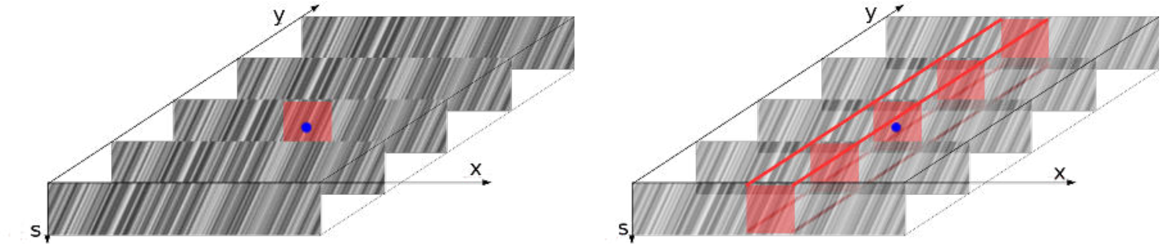
as given in Wanner *et al.* [92], where only the Structure Tensor components are necessary to obtain the disparity.

In a 4D light field, two disparity maps are computable, one with respect to the horizontal light field d_{hori} and one with respect to the vertical light field d_{vert}. The final disparity map d_{final} is achieved by merging the horizontal and vertical disparity map solutions with respect to a reliability measure termed the coherence c. Coherence, computed for both the vertical c_{vert} and horizontal c_{hori} light fields, is represented by the equation introduced in Bigun *et al.* [11]

$$c := \sqrt{\frac{(J_{xx} - J_{ss})^2 + 4(J_{xs})^2}{(J_{xx} + J_{ss})^2}}. \tag{2.9}$$

The merging process between the vertical and horizontal light fields can be described by:

$$d_{\text{final}} = \begin{cases} d_{\text{hori}} & c_{\text{hori}} > c_{\text{vert}} \\ d_{\text{vert}} & c_{\text{vert}} > c_{\text{hori}} \\ 0 & \text{else} \end{cases} \tag{2.10}$$



(a) (b)

Figure 2.3: *(a) visualizes the sphere of influence of the Gaussian filter, to estimate the orientation of the blue center point. This scenario is used in the 2D Structure Tensor. (b) represents the sphere of influence for the 2.5D Structure Tensor. Here also neighboring EPIs are influencing the orientation estimation of the underlying blue center point.*

2.3 2.5D Structure Tensor

While the 2D Structure Tensor uses information from just a single 2D EPI to compute the orientation, the 2.5D Structure Tensor incorporates information from the local 3D image environment which interconnects the local orientations across EPIs, see image 2.3 (b). Its computation for the horizontal and the vertical light fields is achieved by the same Structure Tensor as introduced in equation 2.6. The only difference is the shape of the inner and outer Gaussian smoothing filters. The 2.5D Structure Tensor extends the smoothing range from the 2D EPI in the (s, x) domain to the 3D light field volume (s, x, y) which then involves also the local image information of neighboring EPIs. Due to the transfer of the smoothing of the inner Gaussian and outer Gaussian into the 3D domain, there is more global support for the local orientation computation. Thus smaller kernels can be used to achieve results having precision similar to those of the 2D Structure Tensor. The merge of the vertical and horizontal light fields also depends on the coherence 2.9 as a reliability measure. An advantage of the 2.5D Structure Tensor is that the same kernel size attains higher precision than for the 2D case due to its inclusion of more data. A disadvantage is that by increasing the support it reduces discrimination at sharp transitions between objects at different depths – the standard boundary definition problem.

2.4 3D Structure Tensor

In contrast to the 2D and the 2.5D Structure Tensors, where only derivatives in x and s directions are analyzed, the 3D case involves components in the y direction as well. The 3D Structure Tensor, as described in Muehlich *et al.* [62], is defined by:

$$J = \tau * \begin{pmatrix} \left(\frac{\partial \hat{S}}{\partial x}\right)^2 & \frac{\partial \hat{S}}{\partial x} \cdot \frac{\partial \hat{S}}{\partial s} & \frac{\partial \hat{S}}{\partial x} \cdot \frac{\partial \hat{S}}{\partial y} \\ \frac{\partial \hat{S}}{\partial s} \cdot \frac{\partial \hat{S}}{\partial x} & \left(\frac{\partial \hat{S}}{\partial s}\right)^2 & \frac{\partial \hat{S}}{\partial s} \cdot \frac{\partial \hat{S}}{\partial y} \\ \frac{\partial \hat{S}}{\partial y} \cdot \frac{\partial \hat{S}}{\partial x} & \frac{\partial \hat{S}}{\partial y} \cdot \frac{\partial \hat{S}}{\partial s} & \left(\frac{\partial \hat{S}}{\partial y}\right)^2 \end{pmatrix} =: \begin{pmatrix} J_{xx} & J_{xs} & J_{xy} \\ J_{xs} & J_{ss} & J_{ys} \\ J_{yx} & J_{ys} & J_{ss} \end{pmatrix} \tag{2.11}$$

which is a symmetric 3×3 matrix. The given 3D Structure Tensor represents a positive-semi definite covariance matrix. Thus, it has only real positive eigenvalues related to an orthogonal eigenvector system. These properties make it possible to compute the eigenvectors directly from the given Structure Tensor representation. To know which eigenvector is pointing in the direction of the underlying orientation, a closer look at the eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ is necessary, as described in [38]. There are four different cases to consider:

- 3D-Cube: Gray values do not change in any dimensions, see image 2.4 (d)
  $\lambda_1 \approx 0, \lambda_2 \approx 0, \lambda_3 \approx 0$
  $\rightarrow$ Orientation estimation is not possible

- 2D-Plane: Gray values changes in 1 dimension only, see image 2.4 (c)
  $\lambda_1 > 0, \lambda_2 \approx 0, \lambda_3 \approx 0$
  $\rightarrow$ e_1 is the eigenvector that points stable in normal direction of the orientation.

- 1D-Line: Gray values are constant in 2 dimension, see image 2.4 (b)
  $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 \approx 0$
  $\rightarrow$ e_3 is the eigenvector that points stable in orientation direction.

- 0D-Spot: Gray values change in all dimensions, see image 2.4 (a)
  $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0$
  $\rightarrow$ Disparity estimation is not possible.

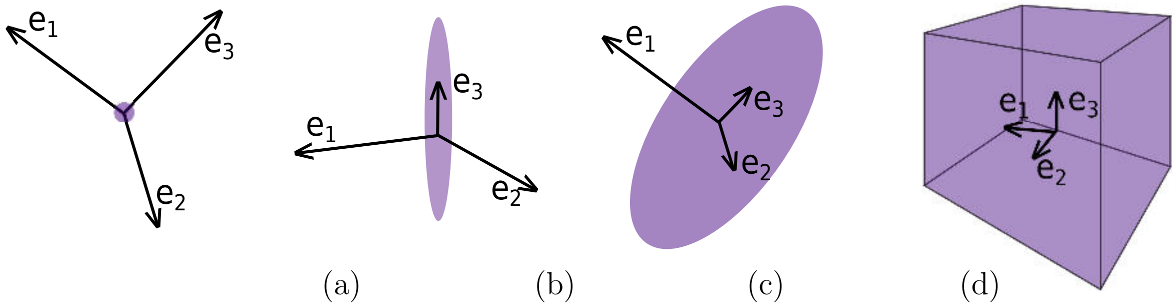


Figure 2.4: *(a) The eigenvectors for a 0D-Spot. (b) 1D line orientation in 3D light field volume. The orientation is represented by the eigenvector with the smallest eigenvalue. (c) Eigenvectors for a 2D plane orientation occurring in the light field volume evaluation. The eigenvector with the largest eigenvalue points stable in normal direction of the orientation. (d) 3D cube, all eigenvalues are small.*

For the final orientation analysis the second and the third cases have to be taken into account. Due to the fact that two eigenvectors are needed for a stable orientation estimation, a reliability measure is necessary to uniquely distinguish them. Thus two reliability measures are defined:

$$c_{12} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \quad c_{12} \in [0,1] \tag{2.12}$$

$$c_{23} = \frac{\lambda_2 - \lambda_3}{\lambda_2 + \lambda_3} \quad c_{23} \in [0,1] \tag{2.13}$$

This two measures are opposed, and the perfect indicator to distinguish the two cases are explained above

$$c_{23} < c_{12} \rightarrow e_1 \tag{2.14}$$

$$c_{12} < c_{23} \rightarrow e_3 \tag{2.15}$$

For the two remaining cases both reliability measures are close to zero and indicate a poor orientation estimate. To compute the final disparity value the computed eigenvectors are interpreted as normal vectors in the case of a 2D plane and as direction vectors in case of a 1D line. In both cases the first two eigenvector components point either in the normal direction for the 2D plane or in the orientation direction for the 1D line.

2.5 4D Structure Tensor

The separation into horizontal and vertical light fields as employed in other Structure Tensor methods is not required in the 4D Structure Tensor approach. To achieve a 4D light-field representation we first consider the 2D light-field approach. In the 2D light field the Structure Tensor is given by the equation

$$J = \tau * \begin{pmatrix} \left(\frac{\partial \hat{S}}{\partial x}\right)^2 & \frac{\partial \hat{S}}{\partial x} \cdot \frac{\partial \hat{S}}{\partial s} \\ \frac{\partial \hat{S}}{\partial s} \cdot \frac{\partial \hat{S}}{\partial x} & \left(\frac{\partial \hat{S}}{\partial s}\right)^2 \end{pmatrix} =: \begin{pmatrix} J_{xx} & J_{xs} \\ J_{xs} & J_{ss} \end{pmatrix}. \tag{2.16}$$

as previously illustrated. This Structure Tensor can be associated with the optical flow for one-dimensional horizontal movements given by the equation

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial s} \quad = \quad 0. \tag{2.17}$$

The transfer of this optical flow to describe vertical and horizontal movements of a light field leads to the following optical flow equation

$$\frac{\partial I}{\partial x} V_s + \frac{\partial I}{\partial y} V_t + \frac{\partial I}{\partial s} + \frac{\partial I}{\partial t} \quad = \quad 0. \tag{2.18}$$

This optical flow approach combines both light-field directions. Considering an equal baseline of the vertical and horizontal light-field directions, the resulting estimated velocity for both directions become $V_s = V_t = V$. Thus the optical flow equation can be expressed by

$$\left(\tfrac{\partial I}{\partial y} + \tfrac{\partial I}{\partial y}\right) V + \frac{\partial I}{\partial s} + \frac{\partial I}{\partial t} \quad = \quad 0 \tag{2.19}$$

Transferring the achieved optical flow approach back to a Structure Tensor formulation, the 4D Structure Tensor approach becomes

$$J = \tau * \begin{pmatrix} \left(\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right)^2 & \left(\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right)\left(\frac{\partial I}{\partial s} + \frac{\partial I}{\partial t}\right) \\ \left(\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right)\left(\frac{\partial I}{\partial s} + \frac{\partial I}{\partial t}\right) & \left(\frac{\partial I}{\partial s} + \frac{\partial I}{\partial t}\right)^2 \end{pmatrix} =: \begin{pmatrix} J_{xyxy} & J_{xyst} \\ J_{xyst} & J_{stst} \end{pmatrix}. \tag{2.20}$$

To obtain the final disparity map we use the previously introduced equation 2.10.

2.6 Benchmarking Results

To compare the reliability of each introduced Structure Tensor, we analyze four different synthetically generated test scenes, shown in Appendix B. Onto the Structure Tensor results, only a coherence thresholded ($c > 0.9$) is applied, but aside this, no further post processing. To compare the different Structure Tensor implementations we use the peak-signal-to-noise ratio (PSNR). For the evaluation of disparity maps we define the PSNR as

$$\mathrm{PSNR} = 10 \log_{10} \frac{\mathrm{MAX}}{\mathrm{MSE}} \tag{2.21}$$

where MSE defines the mean squared error, relative to known ground truth values and MAX the maximal disparity value, which is set to 25 px. The PSNR for different test scenes can be seen in table 2.1. For the applied Structure Tensors we set the inner Gaussian smoothing to $\sigma_{[5\times5]} = 0.5$ and the outer Gaussian smoothing to $\tau_{[9\times9]} = 1.3$. The Scharr filter and the Gaussian derivative filter was used to compute the desired derivatives. A detailed explanation about the inner and outer Gaussian filter and the optimal selections of the filter values σ and τ is discussed in section 6. The resulting disparity maps and the measured mean relative error maps are shown in the Appendix B.

PSNR [%]	Gaussian derivative filter				Scharr filter			
Scene	**2D**	**2.5D**	**3D**	**4D**	**2D**	**2.5D**	**3D**	**4D**
Buddha	24.77	25.01	23.07	22.98	24.71	24.88	24.15	22.90
StillLife	21.92	23.13	19.16	20.50	22.98	23.72	24.77	20.62
MonaRoom	21.69	24.05	20.16	22.98	23.66	24.84	20.98	24.03
Papillion	20.73	24.10	19.42	21.14	22.82	25.89	24.12	23.44

Table 2.1: *Comparison of the peak-signal-to-noise ratio (PSNR) for different synthetically rendered scenes (which provides ground truth). The selected inner Gaussian smoothing is $\sigma_{[5\times5]} = 0.5$ and the selected outer Gaussian smoothing is $\tau_{[9\times9]} = 1.3$. The best Structure Tensor results are highlighted in green.*

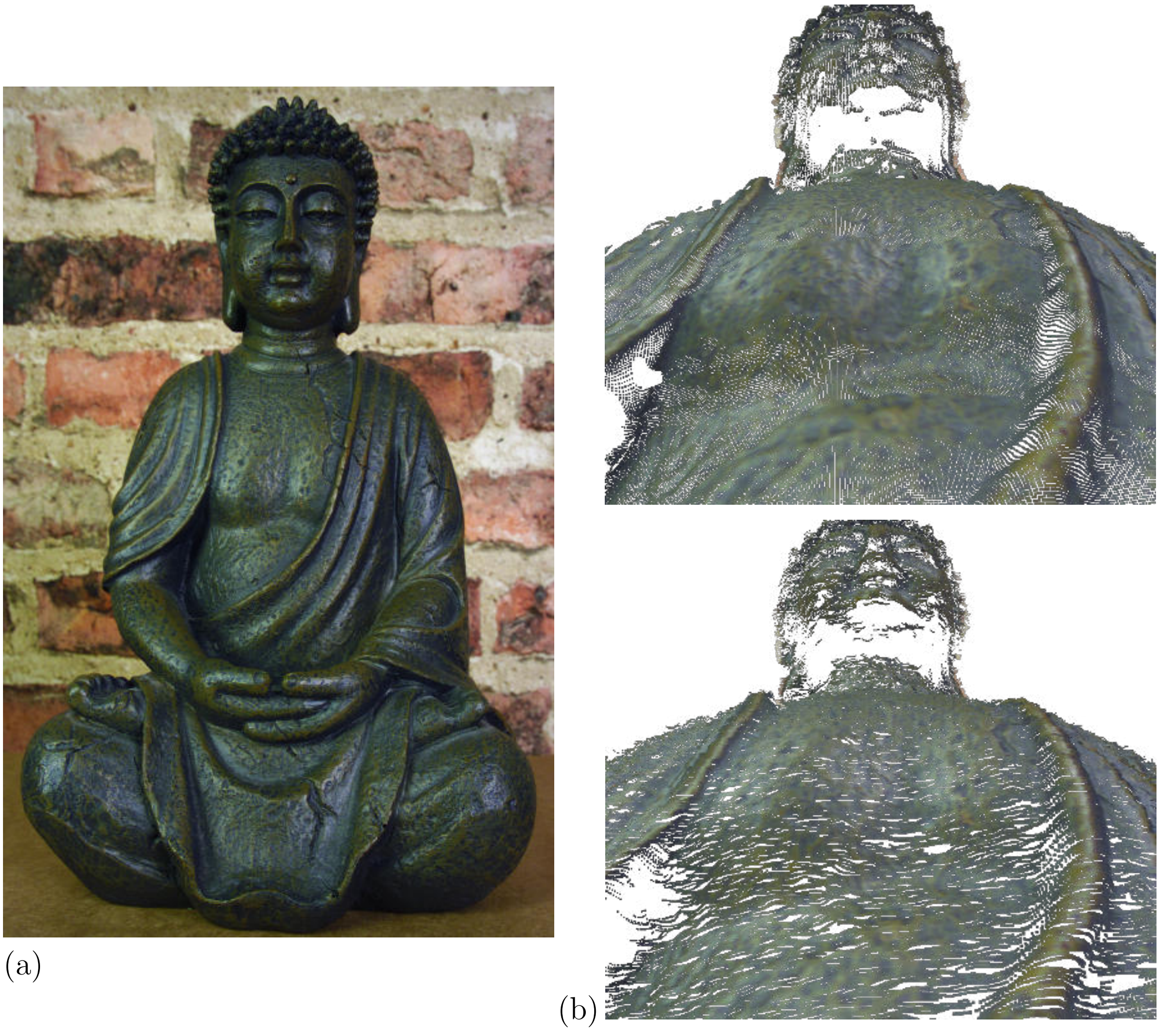


Figure 2.5: *(a) shows the center view image of the processed light field which consists of* 21 *images. (b) shows the point clouds of the 2.5D Structure Tensor on the top in contrast to the 2D Structure Tensor at the bottom. It illustrates that the 2.5D Structure Tensor interconnects the EPI rows (horizontal direction), while the 2D Structure Tensor keeps them independent. Thus much smoother surfaces are possible.*

2.7 Conclusion

The overview in table 2.1 shows that the PSNR values are similar. In general, the 3D Structure Tensor has the worst PSNR and its computation time is high in comparison with the other methods. This cost can be seen as the requirement to compute eigenvalues and eigenvectors for each pixel in a defined evaluation window around each pixel estimate. Following the estimation, the eigenvector representing the correct orientation has to be identified by a reliability measure. This computation is more expensive than that for the other Structure Tensor implementations where only two equations are solved in obtaining the disparity and coherence maps.

The performance of the 4D Structure Tensor is similar to that of the 2D Structure

Tensor. It produces smoother estimates at pixels having the same flow orientation in both directions (i.e., the same depth), but is weaker at object boundaries, where the contrast in orientations causes problems, as also observed in the 2.5D case. The 2D Structure Tensor shows an overall acceptable disparity map, with boundaries appearing to be better detected than in the other methods. Object surfaces seem smoothest with the 2.5D Structure Tensor. This accounts for the 2.5D method having the best PSNR measures, although its smooth surfaces come at the cost of reduced sharpness at object boundaries. For general applications the 2D Structure Tensor may be most recommended. If there is a preference for smooth-surface results, the 2.5D Structure Tensor is the better choice, as shown in figure 2.5.

3 Light-Field Camera Design

3.1 Introduction

Capturing a light field without knowledge of the required setup can lead to unusable image data and poor disparity estimation. Thus it is important to consider how to design a light-field acquisition system, and this involves understanding what we term the *bounded frustum* that encloses the scene volume of interest. With a properly defined setup, high quality light fields can be acquired that enable reaching the full potential of the Structure Tensor ranging analysis methodology. In this chapter we introduce the bounded frustum and discuss, over a variety of defined light-field setups, the precision attainable in each through a precomputation analysis. For the application, we designed a light-field camera configuration program that maps between setup parameters and system quantitative performance. This was introduced in C. This chapter is edited from a publication in SPIE2015 Videmetrics Range Imaging and Application [19]

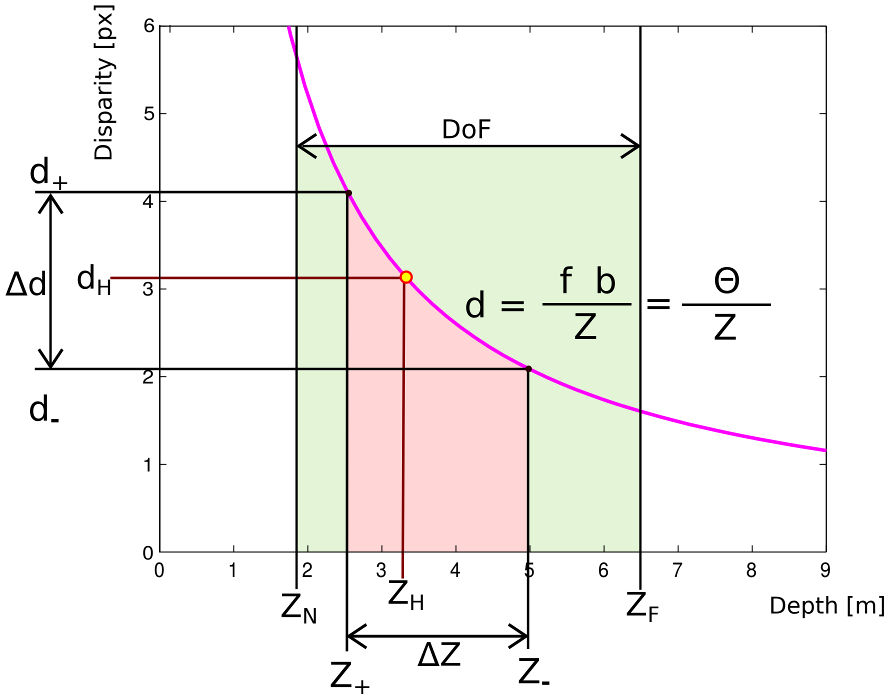


Figure 3.1: *This image illustrates the generalization of the mapping from depth space into the disparity space for arbitrary camera array setups which are definable by its specific Θ value. The reddish region represents the bounded frustum and the greenish region represents the depth of field.*

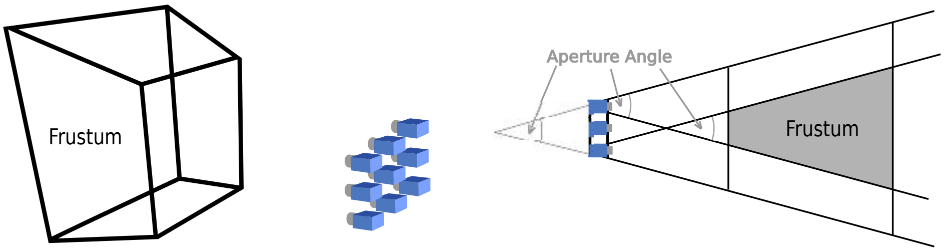


Figure 3.2: *The frustum depends from a designed light-field camera setup. For all objects inside this frustum it is possible to determine the depth information but it also guarantees that all objects are imaged onto the wanted disparity range Δd around the horopter disparity.*

3.2 Optimal estimation range determination

The light-field setup can be defined by its principal parameters of focal length $f_{[m]}$, baseline $b_{[m]}$ and pixel pitch $p_{[m]}$. These parameters determine not only the light-field camera design but also the relationship between depth $Z_{[m]}$ and disparity $d_{[px]}$ through the equation

$$Z_{[m]} = \frac{f_{[px]} b_{[m]}}{d_{[px]}} \tag{3.1}$$

with

$$f_{[px]} = \frac{f_{[m]}}{p_{[m]}}. \tag{3.2}$$

To compute orientation using the Structure Tensor, disparity must lie in a symmetric 2px range ($|\Delta d| = 2\text{px}$) around a given horopter disparity d_H, see figure 3.1. This information leads to

$$d_+ = d_H + \frac{\Delta d}{2} \qquad \text{and} \qquad d_- = d_H - \frac{\Delta d}{2} \tag{3.3}$$

which defines a depth range (equation 3.1),

$$\Delta Z = Z_- - Z_+ = f_{[px]} b_{[m]} \left(\frac{1}{d_+} - \frac{1}{d_-} \right). \tag{3.4}$$

where Z_- defines the far distance limit and Z_+ the near distance limit, as shown in figure 3.1. When the underlying scene has a larger depth range than defined by ΔZ, a global shift must be applied to extend the measurable range (as introduced in chapter 4). Otherwise, as discussed in the previous section, a new horopter disparity d_H and/or the principal parameters have to be adapted to adjust the depth range accordingly. In contrast, if the target scene is more shallow than the designed frustum then the disparity resolution decreases for the given target scene, and this waste of attainable resolution should also be avoided. In addition to the depth constraint, it is important to consider a specific field of view F within the measure distance to ensure that all parts of the target scene are observed by all cameras. The field of view (FOV) can be defined either by that of the reference camera or by the joint scene content seen by all cameras, which then becomes

$$F(Z_i) = \frac{SZ_i}{f_{[m]}} - b_{[m]}C \qquad Z_i \in \{Z_+, ..., Z_-\} \tag{3.5}$$

where S denotes the sensor size and C the number of cameras in the relevant direction. In contrast to a shrinking field of view with increasing number of cameras at a selected depth, the aperture angle remains constant and is defined as

$$FoV(p_{[m]}, R, f_{[m]}) = \frac{180°}{\pi} \cdot 2 \cdot tan^{-1}\left(\frac{p_{[m]}R}{2f_{[m]}}\right) \tag{3.6}$$

where R is image resolution. The introduced depth range together with the frame size define a frustum which determines the volume where the entire scene content should be located, as shown in figure 3.2. These boundary conditions, plus the fact that a finite number of objectives and cameras are available, limit a possible setup configuration in its minimal and maximal achievable baseline $b_{[m]}$ and focal length $f_{[px]}$. Thus, with light-field cameras, it is important to define the bounded frustum with respect to operational requirements. The complexity of all control parameters and the dependence of the bounded frustum on the scene's depth of field make it challenging to design a setup satisfying all constraints. Nevertheless, this effort is important for light-field imaging, as it can help to prevent unintended estimation failures. Thus we define

$$\Theta_{[px\, m]}(f, b, p) = f_{[px]}b_{[m]} = \frac{f_{[m]}b_{[m]}}{p_{[m]}} \tag{3.7}$$

which combines chosen baseline, focal length and pixel pitch into a single conceptual parameter $\boldsymbol{\Theta}$ – a rule of thumb. This $\boldsymbol{\Theta}$ can also be derived directly by the bounded frustum depth constraint:

$$\Theta_{[px\, m]}(\Delta d, \Delta Z, Z_H) = \frac{\Delta d Z^2 + \Delta d Z_H\sqrt{Z_H^2 + \Delta Z^2}}{2\Delta Z}, \tag{3.8}$$

where Z_H denotes the horopter depth and Δd the disparity range of the selected setup. Alternatively, $\boldsymbol{\Theta}$ can be derived by using the equivalent horopter disparity d_H instead of Z_H:

$$\Theta_{[px\, m]}(\Delta d, \Delta Z, d_H) = \frac{\Delta Z}{\Delta d} d_H^2 - \frac{\Delta Z \Delta d}{4}. \tag{3.9}$$

This combination of the frustum with the principal camera parameters facilitates designing a light-field camera setup. In addition, having a Depth of Field (DoF) enclosing the bounded frustum as shown in figure 3.1, sharp images are captured.

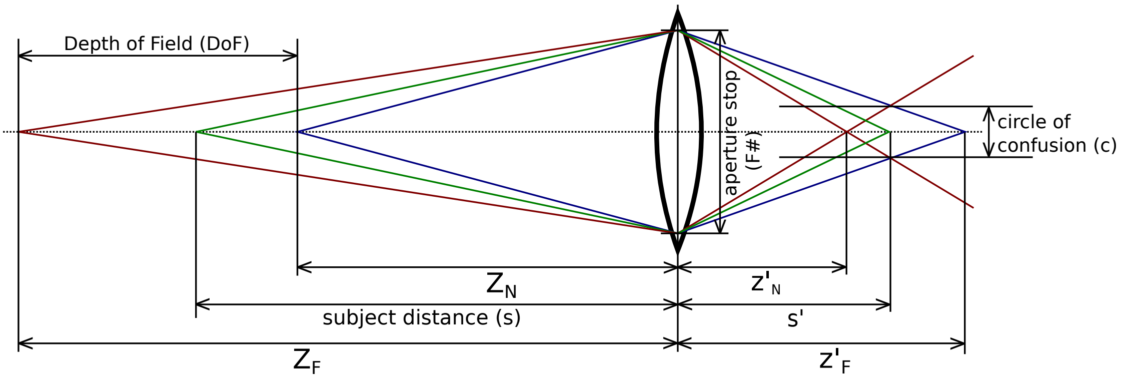


Figure 3.3: *Shows the relation between the Depth of Field and the circle of confusion size.*

3.2.1 The Depth of Field (DoF)

The depth of field, as shown in figure C.4, is with respect to the principal parameters, dependent on focal length and pixel pitch, which limits the size of the circle of confusion c. This brings the advantage of baseline-independent determination of the depth of field. Thus it is possible to use baseline adjustments to ensure the frustum lies within a defined depth of field, delivering consistently sharp images. To determine the depth of field we use the following equations

$$H = \frac{f_{[m]}^2}{c \cdot F} + f_{[m]} \tag{3.10}$$

$$Z_N = \frac{\left(H - f_{[m]}\right)s}{H + \left(s - 2f_{[m]}\right)} \tag{3.11}$$

$$Z_F = \frac{\left(H - f_{[m]}\right)s}{H - s} \tag{3.12}$$

$$\text{DoF} = Z_F - Z_N \tag{3.13}$$

where H defines the hyper-focal distance, Z_N the near, Z_F the far distance limits, and F the aperture stop. But the depth of field is not the only factor limiting sharpness. A small aperture stop introduces diffraction artifacts, which also must be avoided. These artifacts become visible when the cutoff frequency is less than twice the reciprocal of the pixel pitch

$$F = \frac{\lambda}{2p_{[m]}} \tag{3.14}$$

where λ is the wavelength of the incident light, $p_{[m]}$ the pixel pitch and F the aperture stop with respect to the exit pupil. In simple pin-hole camera approximations the entrance and exit pupils coincide, while in real camera systems they differ. The aperture stop of the objective is related to the entrance pupil while the diffraction limit is related to the exit pupil, thus the obtained value has to be converted into the entrance pupil accordingly. To convert one into the other one needs the pupil ratio, which is the diameter of the exit pupil divided by the diameter of the entrance pupil. The pupil ratio should be known by the objective's manufacturer.

3.3 Accuracy and Precision

Accuracy describes the difference between an estimated value and its reference value. Assuming a distribution of measured values, accuracy is the distance between the mean value of the estimation distribution and the reference value, as seen in figure 3.4. This means that the estimated disparities within a precision range $\sigma_{d_{st}}$ cannot be separated reliably. Precision represents a measure of the distance between two neighboring orientations, making it possible to assign both uniquely to the related reference value. In this section the geometric as well as the Structure Tensor based accuracy and precision are computed and evaluated for the Sobel and the Scharr filters. For this purpose we generate synthetic EPIs having orientations from $-1px$ to $+1px$ as shown in figure 3.5.

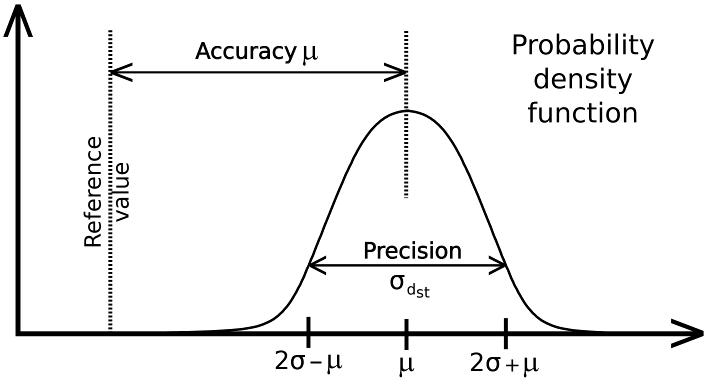


Figure 3.4: *The accuracy is defined by the distance between a reference value and a measured mean value μ of a value distribution. The precision $\sigma_{d_{st}}$ describes the distribution of the measured values around the mean value.*

3.3.1 Precision computation

The precision to separate two orientations is computed by taking M different EPIs and evaluating the estimated orientation at N different uniformly distributed positions. Thus we assume for each orientation $i \in N$ a normally distributed estimation $\mathcal{N}(\mu_i, \sigma_i)$ in the valid orientation range from $-1px$ and $+1px$. The precision is then defined as the $2\sigma_i$ environment around the mean value μ_i covering 95.4% of all estimations of the assumed Gaussian-distributed orientation measures. The resulting overall precision

Figure 3.5: *This figure shows synthetic EPIs to support analysis of Structure Tensor precision. Two examples are shown, where each EPI has different colors and textures. The EPIs contain orientations from $-1px$ tp $1px$ to evaluate the full possible orientation range.*

becomes

$$\sigma_{d_{st}} = \sqrt{\frac{1}{N}\sum_{i}^{N}(\mu_i)^2 + \frac{4}{N}\sum_{i}^{N}(\sigma_i)^2}. \tag{3.15}$$

Aside from the $2\sigma_i$ orientation precision, this formula also considers the systematic error μ_i, which occurs when using rotational-asymmetric derivative filters such as the Sobel filter. The systematic error represents the achieved accuracy, as one can see in figure 3.8. Due to the fact that the achieved accuracy is much larger than the achieved precision we could also neglect the accuracy for the overall precision computation.

3.4 Accuracy and precision in light-field estimation

We assume a light-field setup consisting of a camera array with each camera having different characteristics such as tolerance in the baseline σ_b and tolerance in the focal length σ_f. These inaccuracies affect the orientation estimation. An inaccurate baseline has the effect that linear orientation features in the EPI take on jitter, as seen in figure 3.6. A closer look at the shown orientations reveals a depth-dependent jitter where close objects are more affected. Scene objects at the same depth Z display the same jitter ratio, while closer objects have a larger jitter than objects farther away.
For an accurate depth reconstruction, a bias in the baseline between two cameras can cause a geometrical depth inaccuracy $\sigma_{Z_{geo}}$. Thus we define the depth inaccuracy dependent on the baseline jitter, which becomes

$$\sigma_{Z_{geo}} = \frac{Z}{b_0}\sigma_b, \tag{3.16}$$

where b_0 represents the underlying ground truth baseline.
A focal length deviation, without considering distortion effects, has an effect similar to the baseline deviation, as one can see in figure 3.7. There is also a depth dependent jitter observable in the EPI, but with a different distribution. This happens because of the object's projection onto the image sensor. The more an object moves away from the lens center, the more the perspective projection influences its observed shape and size.

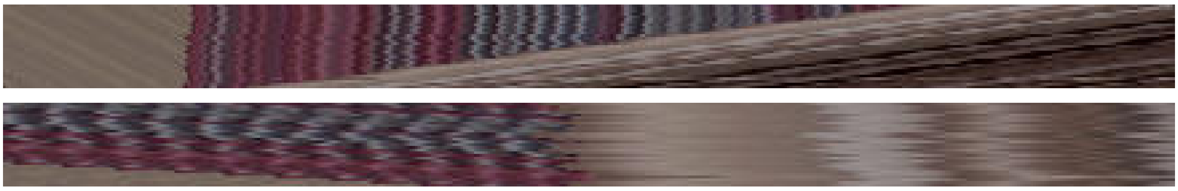


Figure 3.6: *This shows the effect of inaccuracy in the baseline, which causes a depth-dependent jitter in the epipolar-plane image. The same base EPI is shown in both cases, with a horopter placed first at a far distance and then at a close distance. Baseline jitter has increasing influence for objects closer to the camera.*

Thus, EPI regions having scene content close to the lens border show a larger jitter effect. Lens distortion will further increase this effect for real lenses. The resulting geometrical depth inaccuracy $\sigma_{Z_{geo}}$ occurring due to a focal length inaccuracy σ_f can then be defined as

$$\sigma_{Z_{geo}} = \frac{Z}{f_0}\sigma_f \tag{3.17}$$

where f_0 denotes the ground truth focal length. In a resulting light-field camera setup these effects are superimposed and are visible as depth-dependent orientation jitter in the EPI domain. Considering a light-field setup with a deviation in baseline and focal length, the combined depth inaccuracy becomes

$$\sigma_{Z_{geo}} = \sqrt{\left(\frac{Z}{b_0}\sigma_b\right)^2 + \left(\frac{Z}{f_0}\sigma_f\right)^2}. \tag{3.18}$$

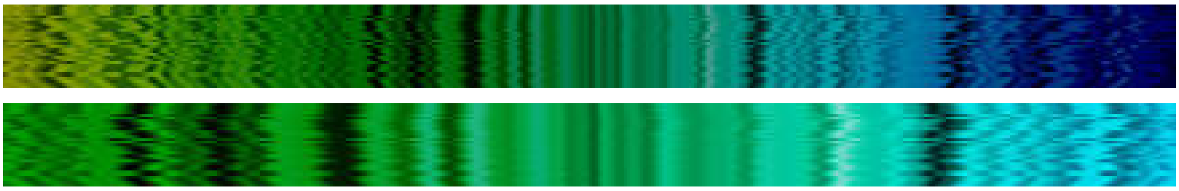


__Figure 3.7:__ Inaccuracy in camera focal length causes jitter with a parabolic distribution in EPI space. The top image shows the jitter distribution of a plane located at a distant position of the camera array, and the bottom shows the same plane close to the camera. Jitter distribution looks like a zoomed part of the top image which means that the jitter ratio increases.

3.4.1 Structure Tensor Accuracy and Precision

To determine the resulting precision of the Structure Tensor orientation estimation, an orientation jitter value σ_j must be defined to address the amount of jitter in the disparity space. This makes it possible to generate synthetic scenes having a certain amount of baseline or focal length jitter while knowing the related orientation jitter value σ_j in disparity space. This transfer of the jitter values σ_b and σ_f into disparity space has the advantage that they become depth independent which means that different jitter values can lead to different depths Z at the same orientation jitter σ_j. The transfer of baseline jitter σ_b and focal length jitter σ_f can be described by the equation

$$\sigma_{px} = \frac{f_0 b_0}{Z}\sqrt{\left(\frac{\sigma_b}{b_0}\right)^2 + \left(\frac{\sigma_f}{f_0}\right)^2}, \tag{3.19}$$

where f_0 denotes the focal length and b_0 the baseline without jitter. Next we compute the Structure Tensor precision related to the defined σ_{px} value and are able to use the resulting Structure Tensor precision values $\sigma_{d_{st}}$ as shown in table 3.1 to directly connect them with an underlying baseline and focal length jitter. The final Structure Tensor dependent depth precision $\sigma_{Z_{st}}$ can now be referred to scene dependent focal length σ_f and baseline jitter σ_b values and can be computed with the formula

$$\sigma_Z = \frac{Z^2}{f_0 b_0}\sigma_{d_{st}}(\sigma_{px}), \tag{3.20}$$

where Z is the depth at which we want to estimate the precision. The combination of the geometrical depth inaccuracy and the Structure Tensor depth inaccuracy gives an overall depth precision for the entire setup and can be computed by the formula

$$\sigma_Z = \sqrt{\left(\frac{Z^2}{f_0 b_0}\sigma_{d_{st}}(\sigma_{px})\right)^2 + \left(\frac{Z}{b_0}\sigma_b\right)^2 + \left(\frac{Z}{f_0}\sigma_f\right)^2}. \tag{3.21}$$

σ_{px}	10^{-5}	$2*10^{-5}$	$3*10^{-5}$	$4*10^{-5}$	$5*10^{-5}$	$6*10^{-5}$	$7*10^{-5}$
Sobel	0.0615	0.0603	0.0612	0.0062	0.0625	0.0614	0.0619
Scharr	0.0077	0.0080	0.0087	0.0093	0.0103	0.0111	0.0118
σ_{px}	$8*10^{-5}$	$9*10^{-5}$	10^{-4}	$2*10^{-4}$	$3*10^{-4}$	$4*10^{-4}$	$5*10^{-4}$
Sobel	0.0621	0.0611	0.0610	0.0691	0.0727	0.0774	0.0845
Scharr	0.0131	0.0147	0.0148	0.0276	0.0410	0.0443	0.0678
σ_{px}	$6*10^{-4}$	$7*10^{-4}$	$8*10^{-4}$	$9*10^{-4}$	10^{-3}		
Sobel	0.1015	0.1136	0.1244	0.1491	0.1668		
Scharr	0.0734	0.1016	0.1104	0.1166	0.1232		

Table 3.1: *The table shows the resulting precision $\sigma_{d_{st}}$ of the Structure Tensor with respect to the used derivative filter and the amount of disparity jitter σ_{px}. A visualization of the values are shown in figure 3.9*

3.5 Results

3.5.1 Precision distribution analysis

Since the precision distribution is imaged with box-whisker diagrams, it is possible to analyze the underlying distribution and get a feeling for the achievable precision. The red horizontal line of a box-whisker diagram in the center defines the median value of the distribution. 50% of all estimations starting from 25% of all measured values to 75% of all measured values is covered with the blue region. The whiskers cover 99.3% of all measurements covering 0.35% through 99.65%. Measured values outside of the whisker region are called outliers and are marked as red crosses in the diagram. The shapes of the distributions, as seen in figure 3.10, gives information about not only the rotation

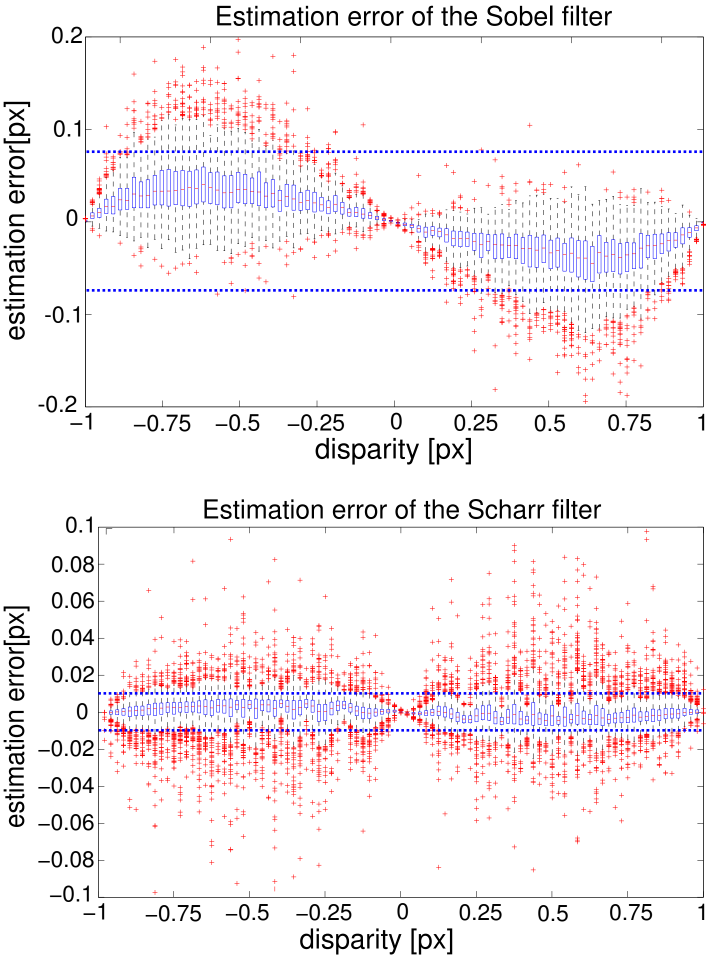


Figure 3.8: *The upper graph shows precision with the Sobel filter, evaluated at EPIs shown in figure 3.5, and the lower graph shows precision for the same EPIs using the Scharr filter. The blue horizontal dashed line illustrate the resulting precision. One can see the rotation asymmetry, the poorer precision for the Sobel filter, and the superiority of the Scharr filter.*

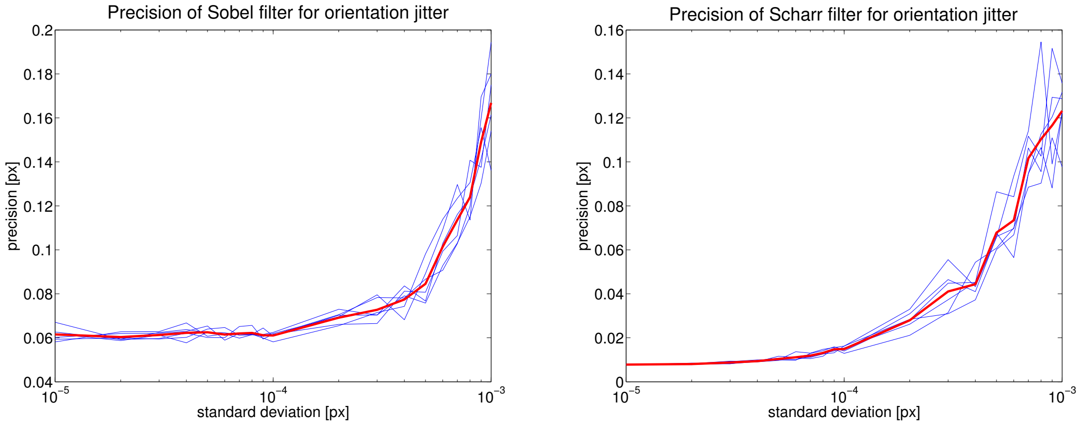


Figure 3.9: *The images show the resulting precision $\Delta d_{st}(\sigma_{px})$ for a different disparity jitter σ_{px} in the EPI. The first image shows the result for the Sobel filter and the second for the Scharr filter. The blue curves shows independent measurements, while the red curve illustrates the average value.*

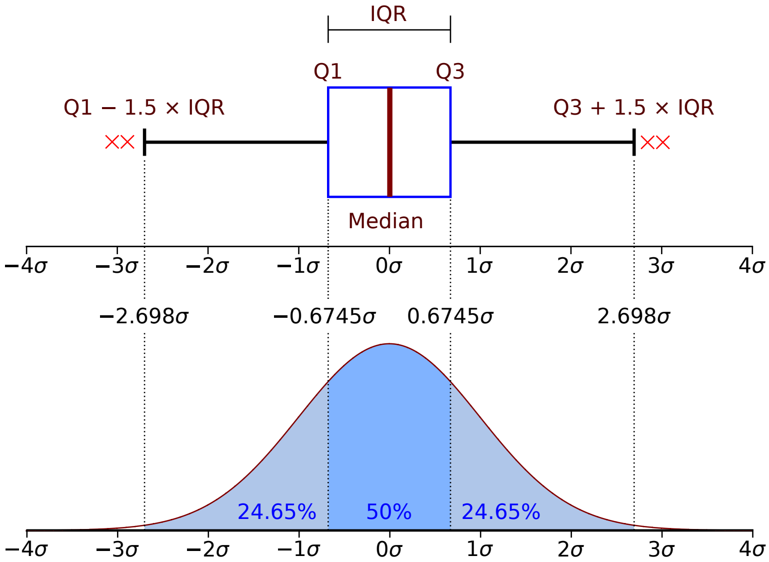


Figure 3.10: *This picture illustrated how a Gaussian distribution is imaged with a box-whisker diagram [15]. The center box (blue) covers* 50% *of all points. Almost all other points are in case of a underlying Gaussian distribution are covered with the whiskers on each side (black lines). When there are still some outliers they become displayed as red crosses outside the whiskers.*

symmetry of each derivative filter but also the underlying estimation distribution. The distribution of the mean values in each box-whisker plot shows the systematic error of each derivative filter. This error is caused by the rotation symmetry. Non-symmetric filters such as Sobel show a large sine-curve progression while optimized filters such as Scharr minimize this error. Another important fact that one can observe from the huge amount of outliers in the measurements is that the distribution is not Gaussian. The underlying distribution must be peaked around the median value having a large flat noise value environment as it is sketched in figure 3.11 by the green distribution. This kind of distribution causes the whiskers to shorten and the outlier range to increase. For such a distribution the 2σ range defines a good upper limit estimation precision.

3.5.2 The optimal measurement

Using the introduced equations for Θ we are able to define a bounded frustum with respect to a given camera setup, and vice versa. To simulate a cross light-field configuration, as shown in figure 2.2, we present a Blender scene with 11 horizontal and 11 vertical cameras sharing a center view. Further, we position the scene content completely inside the defined frustum.

After capturing the first set of views, we twice decrease the baseline, each time by a factor of two, to obtain a larger frustum over the Buddha scene. Here we ensure that, after the baseline reduction, the scene content remains inside the bounded frustum. On applying the Structure Tensor computations to the rendered data for center view depth maps, we triangulate the resulting estimates, forming 3D point clouds. This is

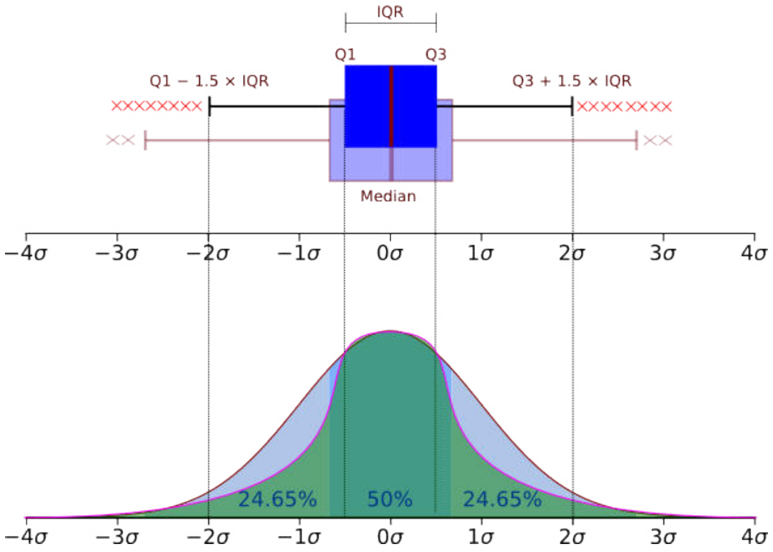


Figure 3.11: *The high number of outliers (red crosses), for each analyzed orientation as shown in figure 3.8, a non Gaussian distribution is assumed. Thus the underlying distribution must have a shape, proposed by the green curvature.*

a useful representation for comparing estimated depths. The resulting point clouds, as shown in figure 3.14, illustrate the reduction in depth resolution due to decreasing the baseline. This happens because the frustum depth (ΔZ) increases with decreasing baseline. Thus the scene depth reduces with respect to the frustum depth range and depth resolution decreases, resulting in a higher inaccuracy as seen in figure 3.14.
The same straight-forward procedure to obtain a bounded frustum is now applied on a real scene. The scene was captured with an IDS UI-1240ML-C-HQ camera utilizing a Kowa $f = 5mm/F1.8$ objective. The camera was mounted on a 2D gantry able to move with high accuracy ($< 5\mu m$) in vertical and horizontal directions. This captured light field also consists of 11 images in each direction.

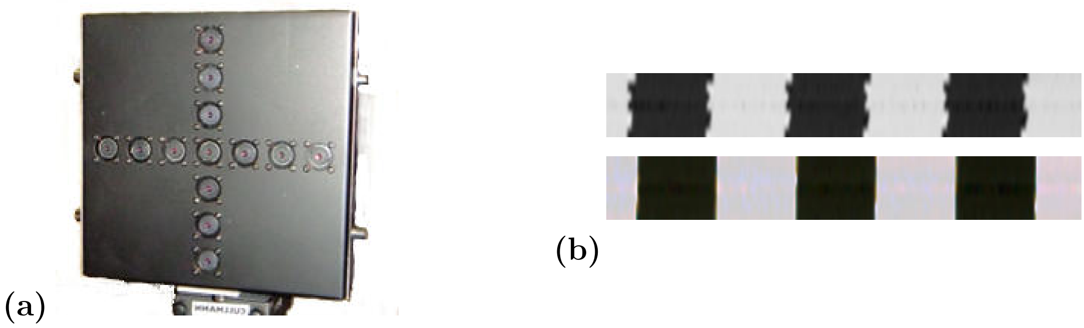


Figure 3.12: *(a)Shows the camera array used to capture a chessboard calibration pattern. (b) Shows the EPIs of a chessboard calibration target. The upper image shows EPI resampling using OpenCV's camera calibration, and the lower shows the result using a specialized light-field calibration method (Kurillo* et al. *[46]). The advantage of light-field calibration over the standard methods is evident.*

3.5.3 Synthetic data

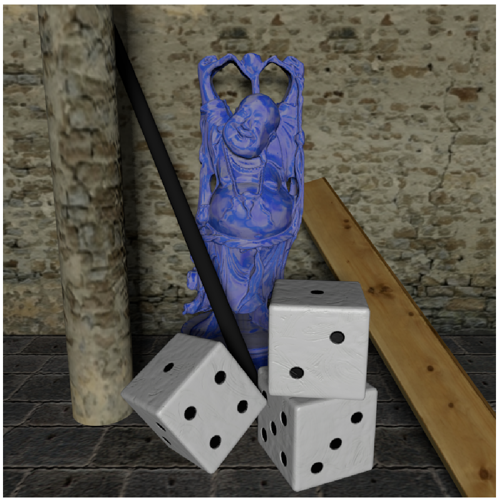


Figure 3.13: *Shown is one of the center-view images of the analyzed cross light fields. The light field is synthetically rendered.*

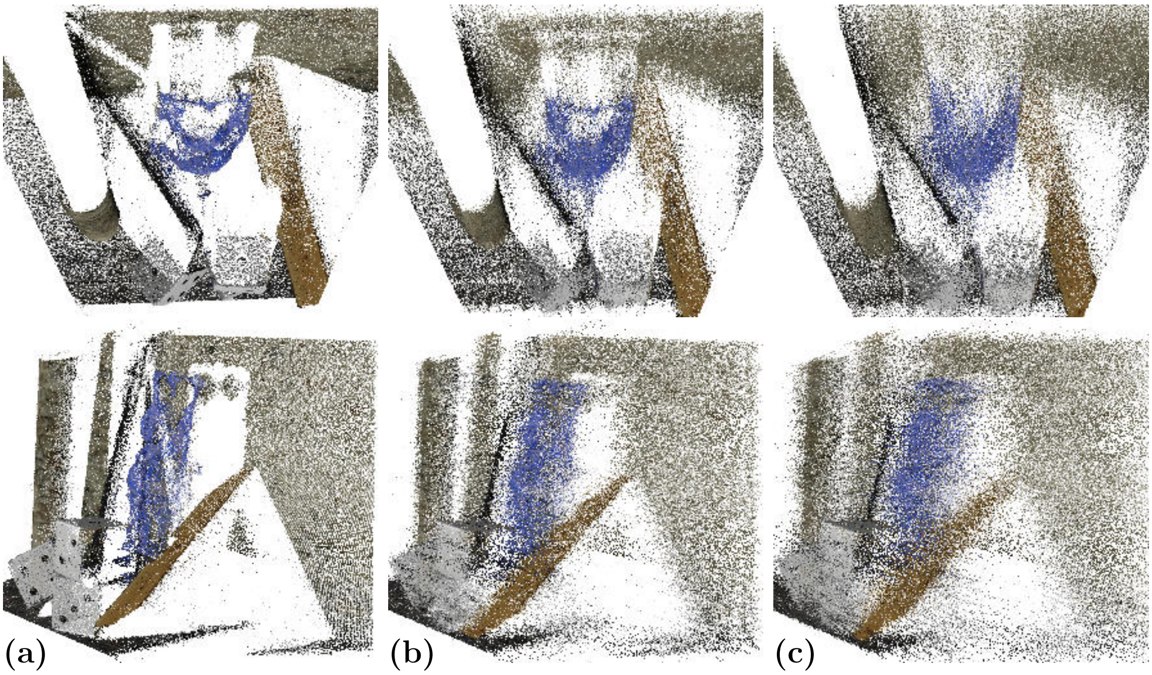


Figure 3.14: *A cross light-field dataset, as shown in figure 2.2, is generated in Blender using a virtual camera of resolution* 1280 × 960 *px and focal length of* 10 *mm. Column (a) has a baseline of* 4*mm, (b)* 2 *mm, and (c)* 1 *mm. Top- and side-view point clouds are shown. The images illustrate the decrease of precision with a decreasing baseline which shows that the frustum becomes larger than the target scene and the disparity resolution decreases.*

3.5.4 Real data

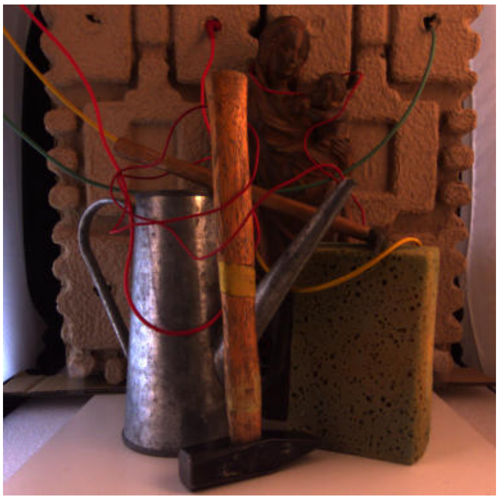


Figure 3.15: *Shown is one of the center-view images of the analyzed cross light fields. The light field is real captured.*

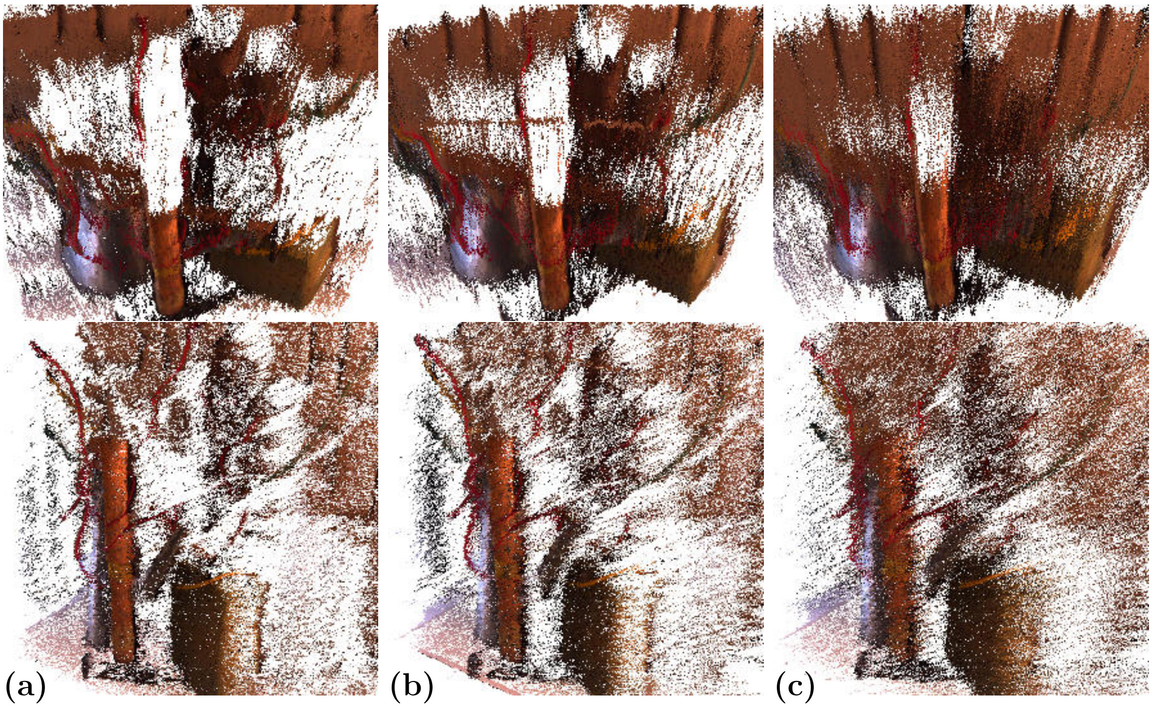


Figure 3.16: *Acquired is a cross light-field dataset, as shown in figure 2.2, using an IDS UI-1240ML-C-HQ camera with* 1280 × 960 *px resolution and a* 5 mm *Kowa objective. Column (a) has a decremental baseline of* 4 mm*, (b)* 2 mm*, and (c)* 1 mm*. Top- and side-view point clouds are shown. As observed with synthetic data, the images illustrate the decrease of precision with a decreasing baseline which shows that the frustum becomes larger than the target scene and the disparity resolution decreases.*

3.6 Conclusion

Due to the fact that variations in the focal length and in the baseline cause a rapid decrease in precision, it is necessary to calibrate the acquisition setup with an accurate calibration tool. Unfortunately most of the common calibration toolboxes are not sufficiently accurate to be used in light-field imaging, as shown in figure 3.12. Access to a calibration tool designed for EPI-based light-field imaging (such as that introduced by Kurillo *et al.* [46]) is indispensable for discrete-imager light-field camera studies. Nonetheless, even a highly accurate optimization is only able to minimize the error, and not to remove it. This is because of the depth dependency of the error and the impracticality of constructing a camera with ideal positioning and ideal optical characteristics. An image taken by an acquisition system, having an erroneous focal length and having an erroneous baseline contains occlusions valid for this position and object sizes valid for this focal length. No calibration can correct the parallax shift or the size of an image without knowing the scene depth – which is what we want to estimate (the proverbial chicken-and-egg problem). To overcome many of the problems associated with camera arrays, a high-precision translation stage can be seen as a good alternative in capturing light fields where scene dynamics are not an issue. Here, the calibration simplifies to a single camera's intrinsics, and known relative position is precise enough that one may neglect the baseline calibration. Thus is it possible to align the defined frustum perfectly to the scene, and due to the high-precise and constant translation it results in the exploitation of the maximal precision possible with the Structure Tensor based orientation estimation within the $2px$ range.

4 Sparse light field evaluation

In this chapter, we remove the restriction of Structure Tensor's 2 px disparity range by performing a global shift on the EPIs/Images to align about selected *horopters*. Through the use of several such horopters, we can obtain independent estimates of depth centered around a selection of distances. These results are merged based on their computed coherence values. The final combined disparity map represents the global range solution, as shown in figure 4.5. This use of shifting to horopter values makes it possible to analyze light fields acquired with fewer cameras and having greater inter-camera separations, with the disparity ranges are considerably larger than structure-tensor's expected 2 px.

4.1 The Horopter, human vision perception

In the eleventh century, Ibn al-Haytham discovered the horopter, building on the binocular vision work of Ptolemy. He discovered that objects lying at the fixation point of a binocular lens system result in a single image, whereas objects off the fixation point result in double images [37]. The phenomena was given the name *horopter* by Belgian mathematician Franciscus Aguilonius in 1613. In light-field imaging, the horopter describes a global disparity shift of each image along the epipolar line (see figure 4.2). Objects located at the horopter are imaged onto the same pixel in each final shifted image – they exhibit zero disparity. In contrast to human eyeball vision where the horopter is a sphere, in light-field imaging with planar sensors the horopter is a plane, and lies parallel to the camera plane as shown in figure 3.4.1.

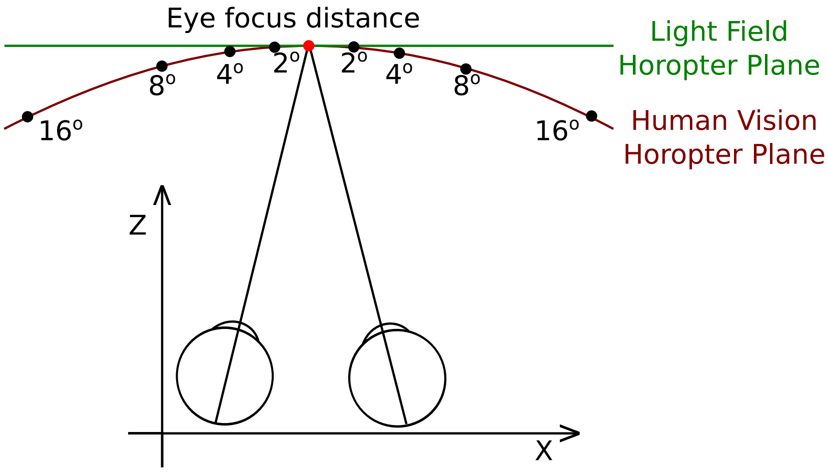


Figure 4.1: *The horopter in human eye perception defines a curvature around the human head. Its curvature changes for different eye focus distances. In light-field imaging the horopter is a parallel plane with respect to the camera sensor orientations.*

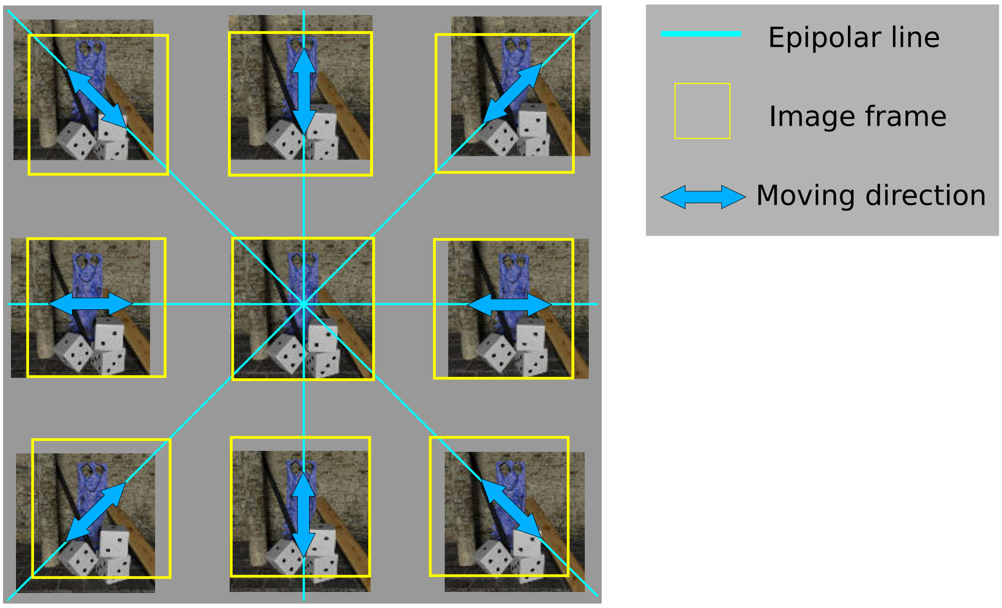


Figure 4.2: *This figure shows the global shifting in the image domain. Each image is individually shifted along the epipolar line. The amount of shifting is related to a defined horopter depth. All objects in the image located at the horopter depth then have zero disparity in the resulting shifted light field.*

4.2 Image representation of global shift

On capturing a set of light-field images, it is usually not practical to simply apply the Structure Tensor and compute a disparity map. An initial shifting will be mandatory to align images at the desired zero disparity to ensure there is a measurable range of ± 1 pixel is attained, meeting the requirements of the Structure Tensor. The amount of initial shifting depends on the light-field setup as well as on the defined frustum, as described in chapter 3. This two-pixel range arises because orientations larger than this ± 1 pixel will appear discontinuous and be fragmented to the filter. To make the structure-tensor-based orientation estimation applicable on light fields exhibiting extreme orientation variations, or where the content is very near and the ”vergence point” of the camera system is far away (at infinity, for example), a global disparity shift must be applied to the input images to, effectively, bring that distant vergence to the central area of the scene content. This is akin to focusing. Such a global disparity shift, as introduced in [20], can be applied several times, selecting different horopter distances for each, effectively resampling the scene range in units of 2-pixel disparity to

$$D_{H_i} \in \{D_{H_1}, ..., D_{H_N} | D_{H_1} < ... < D_{H_N}\}, \tag{4.1}$$

where N denotes the number of positions of the horopter. In depth space, horopter distances are depth dependent and thus a non-linear series of displacements should be applied to avoid regions being redundantly estimated (see figure 4.3). To obtain linear

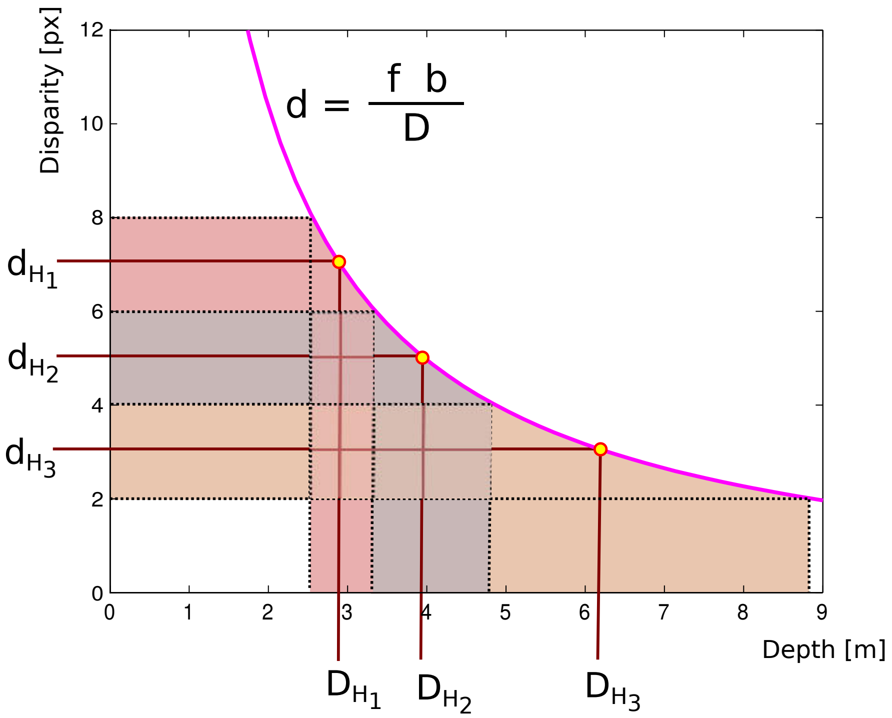


Figure 4.3: *Illustrates the mapping between the horopter depth values D_{H_i} and the horopter disparity values d_{H_i}. While the global shifts, represented by the translated horopter position, are linear in the disparity domain, in the depth domain it has a nonlinear relation which is defined by the camera setup parameters.*

shifts, the global shifting is transferred to the depth independent disparity space by using the equation

$$d_H = \frac{fb}{D_H}, \tag{4.2}$$

where f denotes the focal length and b the baseline of the camera system. As seen in figure 4.3 the global shifting is now linearized and depth independent. For a closed disparity map computation, a maximal possible global shift of 2 px is possible. The global shifting process in the disparity domain is defined by the equation

$$d_{H_i} = 2n \qquad n \in \mathbb{Z}. \tag{4.3}$$

This 2 px shift distance is deduced from the maximal possible orientation range, but may also be chosen smaller. Thus no overlap appears and the entire scene is computed in an efficient way.

Considering a 4D light-field representation given by the equation

$$L : \Omega \times \Pi \to \mathbb{R} \qquad (s,t,x,y) \mapsto L(s,t,x,y), \tag{4.4}$$

the resulting shifted images $I_{s,t}$ are described by the equation

$$\hat{I}^i_{s,t} : I_{s,t} \to \mathbb{R} \tag{4.5}$$

$$(x,y) \mapsto \hat{I}^i_{s,t}(x,y) := L(s,t,x+\Delta x(s), y+\Delta y(t)) \tag{4.6}$$

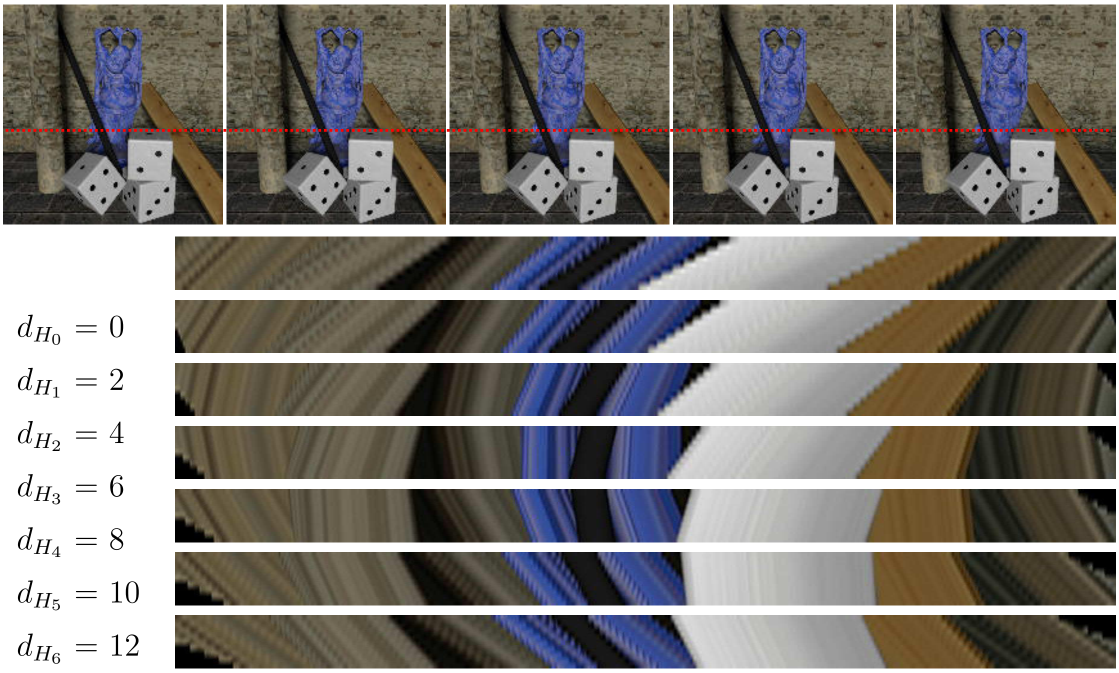


Figure 4.4: *This image shows the global shifting for the Buddha scene, where different horopter disparities d_{H_i} are applied. The shown EPIs are related to the upper image row at the red crossing horizontal line.*

with

$$\Delta x(s) := \frac{(s_{ref} - s)}{s} d_{H_i}, \quad \text{and} \quad \Delta y(t) := \frac{(t_{ref} - t)}{t} d_{H_i}, \tag{4.7}$$

where s_{ref} and t_{ref} define a reference image, mostly the center view, to which the global shifting relates.

4.3 EPI representation of global shifting

Global shifting is applied on horizontal Σ_{t^*,y^*} and vertical Σ_{s^*,x^*} EPIs and illustrated in figure 4.4. As earlier, we will only describe the horizontal direction of processing in the following description. A shifted EPI $\hat{S}^i_{t^*,y^*}$ at a defined horopter disparity d_{H_i} becomes

$$\hat{S}^i_{t^*,y^*} : \Sigma_{t^*,y^*} \to \mathbb{R} \tag{4.8}$$

$$(x, s) \mapsto \hat{S}^i_{t^*,y^*}(x, s) := L(s, t^*, x + \Delta x(s), y^*) \tag{4.9}$$

with

$$\Delta x(s) := \frac{(s_{ref} - s)}{s} d_H, \tag{4.10}$$

which defines the amount of displacement $\Delta x(s)$ for each row in an EPI to satisfy the horopter definition.

4.4 Global disparity map

After applying the global shifts, a tuple of disparity maps is computed,

$$d_i(x,y) \in \{d_1(x,y), ..., d_N(x,y)\}, \tag{4.11}$$

where each relates to a given horopter displacement d_{H_i}, as shown in figure 4.5(a). In addition to the disparity maps, the coherence maps are additionally computed

$$c_i(x,y) \in \{c_1(x,y), ..., c_N(x,y)\} \tag{4.12}$$

as shown in figure 4.6 (a). For the final disparity map $d(x,y)$ and its related coherence map $c(x,y)$, a filtering based on coherence is applied, described by the formula

$$d(x,y) \quad = d_I(x,y) + d_{H_i} \tag{4.13}$$

$$c(x,y) \qquad = c_I(x,y), \tag{4.14}$$

where

$$I(x,y) = \arg\max_i \{c_i(x,s)\} \tag{4.15}$$

indexes the highest reliability at each coordinate. This disparity merge is shown in figure 4.5 (b) with coherence in figure 4.6 (b).

4.5 Conclusion

Global shifting makes it possible to arbitrarily extend the 2 px range of the Structure Tensor. A global solution is achieved through merging disparity layers of size 2 px using their computed coherence values. With this, it becomes possible to use this light-field approach in analyzing scenes of much greater disparity range, such as those typical of the Middlebury data sets (see figure 4.7). In addition, this permits extending the accuracies and precisions achievable to these multiples of 2 px disparity ranges.

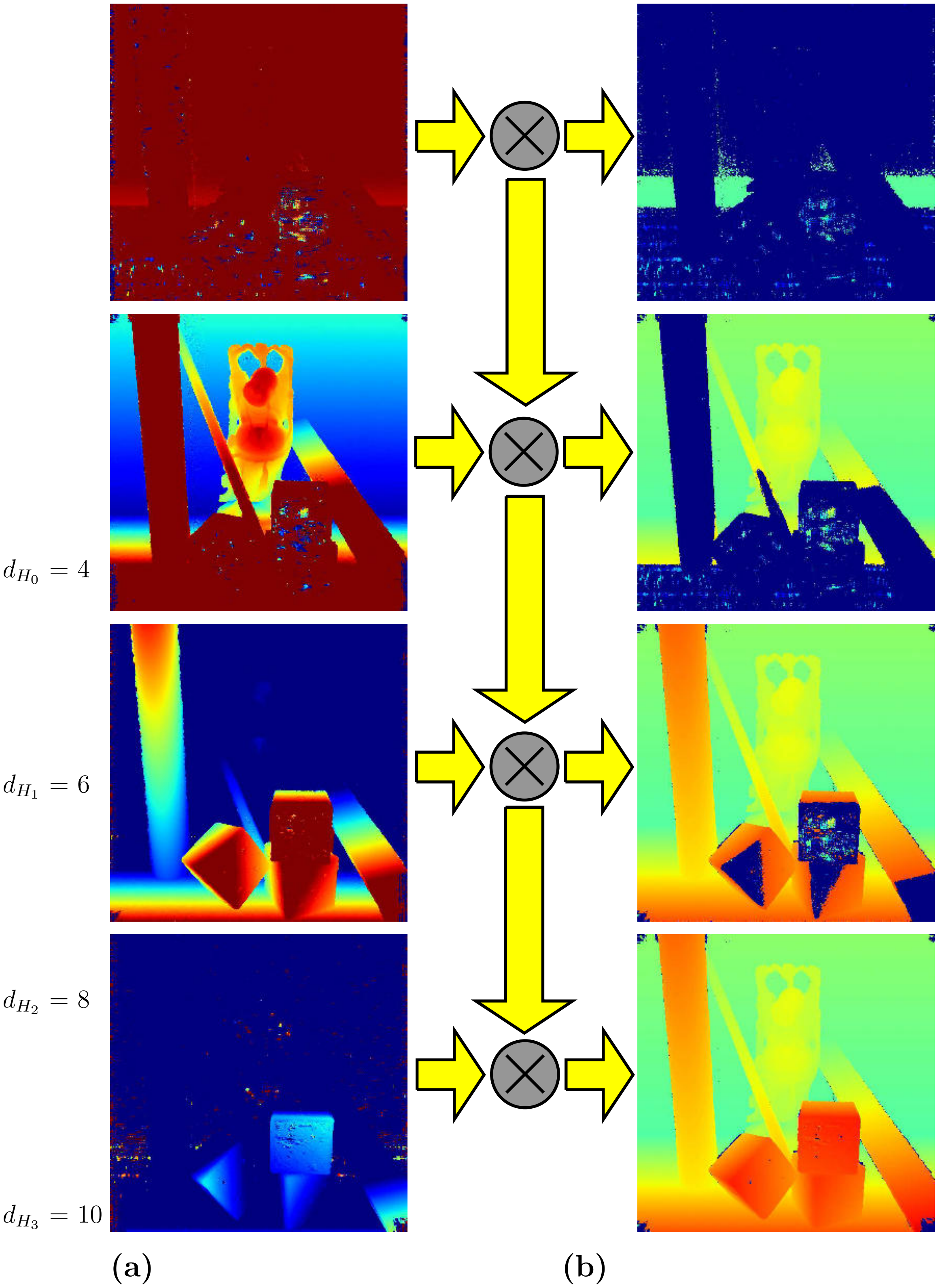


Figure 4.5: *Illustrated is the merge of local disparity maps $d_{H_i}(x, y)$ processed at different horopters H_i. For the superimposition the additional computed coherence maps are used. (a) shows the local disparity maps which relate to different global shifts. (b) displays the merged disparity maps, shown for each iteration step.*

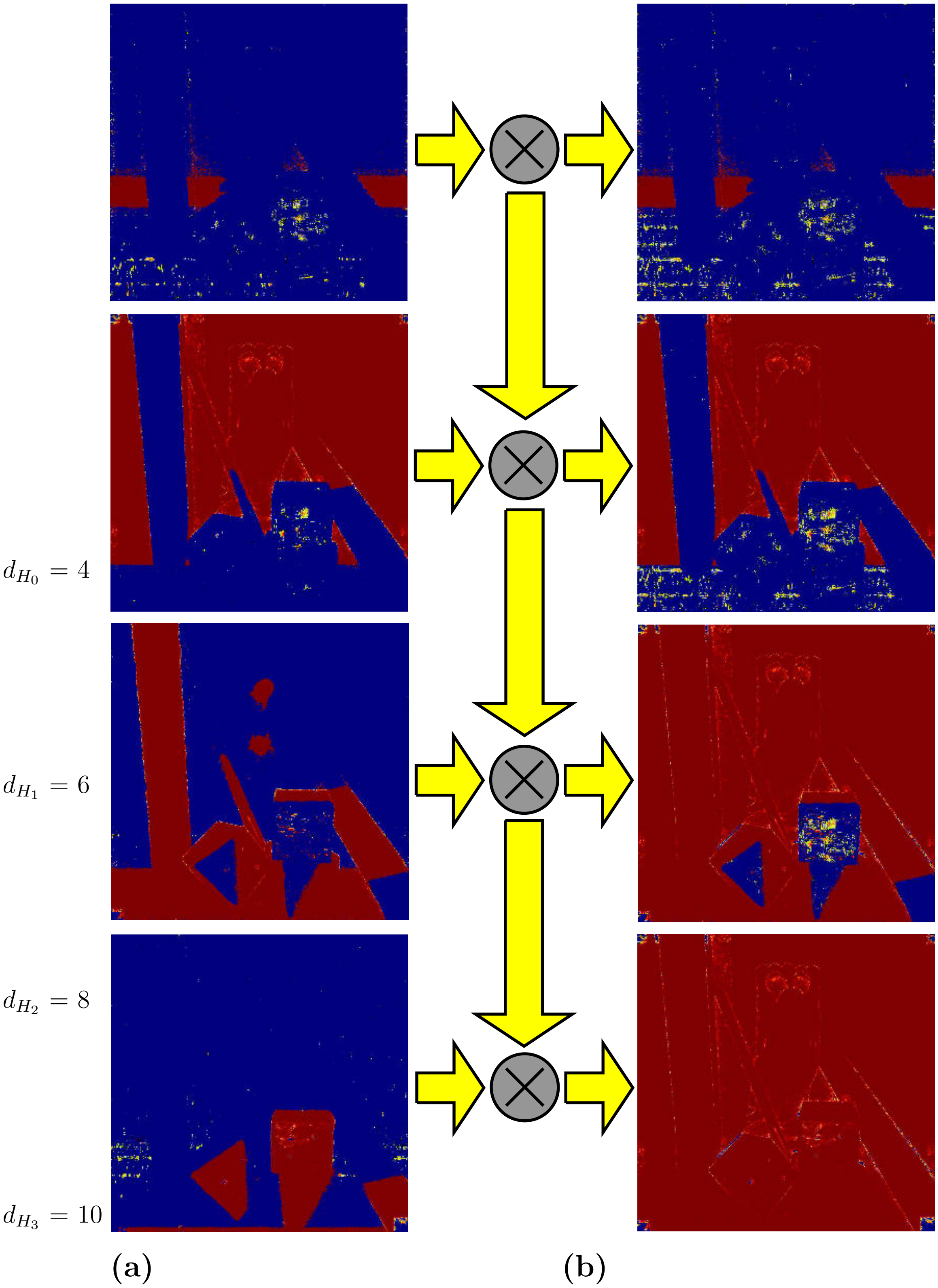


Figure 4.6: *Shows the superimposition of the local coherence maps $c_i(x, y)$ via coherence merge to achieve a final coherence map. (a) displays the local coherence maps related to different global shifts. (b) shows the merged coherence maps. In each iteration step the coherence map fills up.*

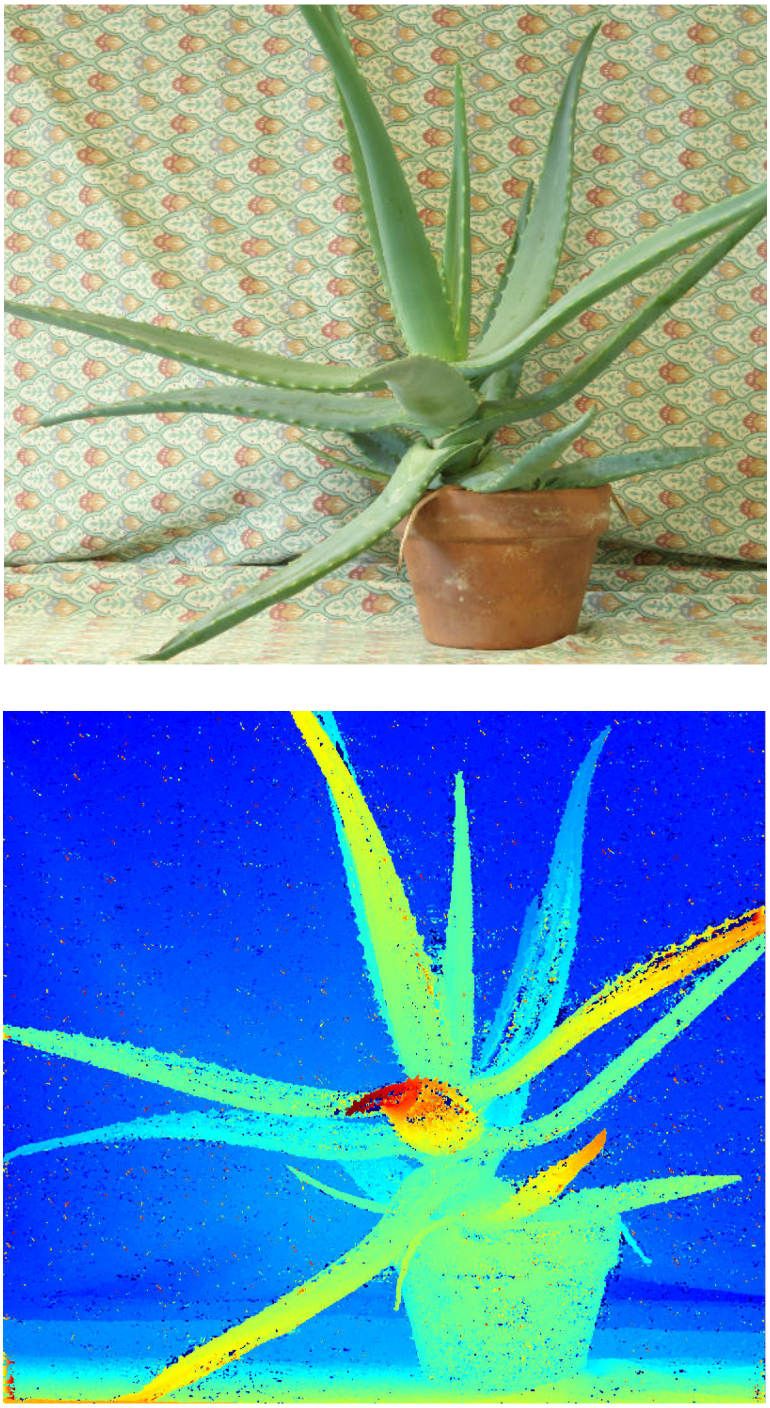


Figure 4.7: *This figure shows the resulting disparity map at the bottom and the related view on top of the Middlebury Aloe dataset [35, 77]. The entire disparity range of the shown scene is 40 px, starting from 51 px down to 11 px. The entire light field consists of 7 images and is processed with the proposed global shifting method.*

5 Coherence Analysis

Coherence is an indicator for the validity of disparity estimations. It is used in global shifting to merge shifted layers. This metric works because coherence drops toward zero when orientation is outside of the measurable pixel range. We now consider a cross-shaped light-field configuration where vertical and horizontal disparity maps can be merged. Coherence can also be used as a merge decision criterion here, as it represents the reliability of orientation estimations – orientations in one direction may provide better estimates than in the other, and coherence can be used to distinguish them.
In this chapter we analyze the use of coherence for this merging and determine the relationship between coherence values, estimated disparity, and ground truth value. We analyze the coherence distribution with respect to the disparity error distribution of the Structure Tensor as shown in figure 5.1 (a). The disparity error distribution illustrates

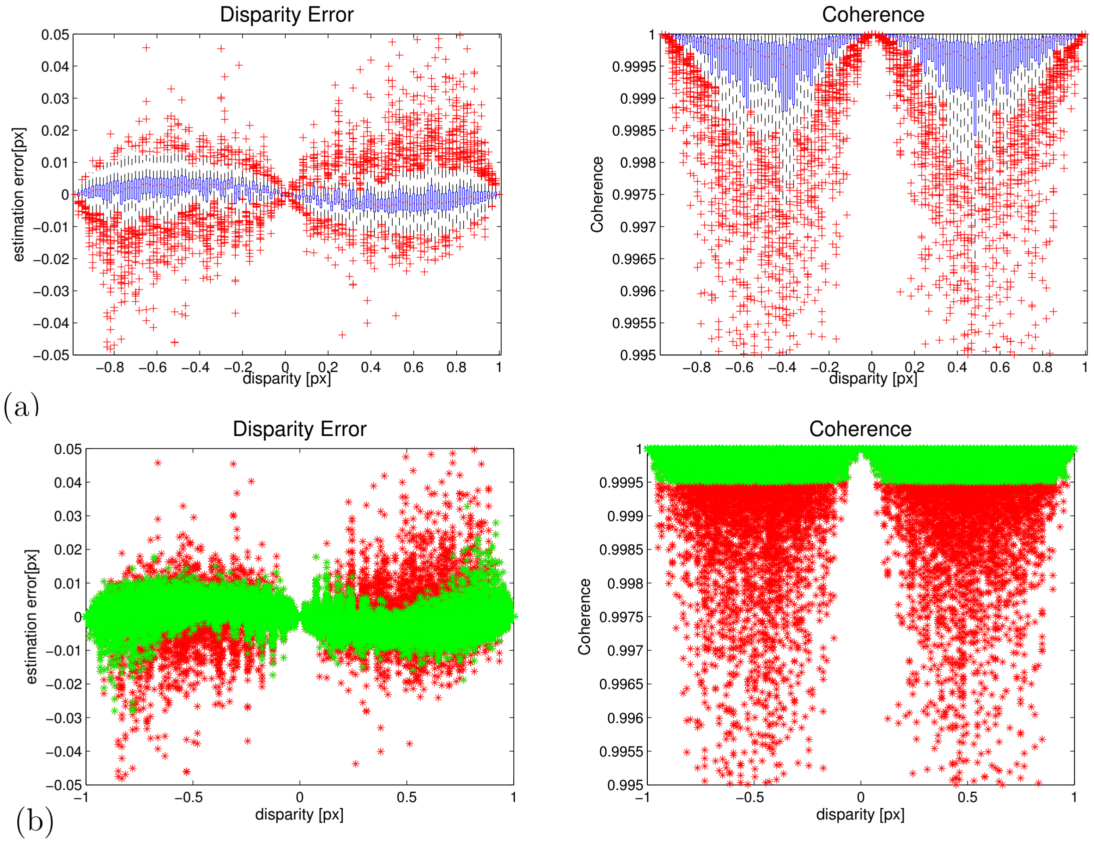


Figure 5.1: *(a)The image on the left shows the precision evaluation of the Structure Tensor using the Scharr derivative filter. The image on the right shows the related coherence values. (b) This figure shows the estimation distribution for all measured orientations before thresholding (red) and after thresholding (green). One can see that the outliers are reduced by simply applying a constant coherence thresholding.*

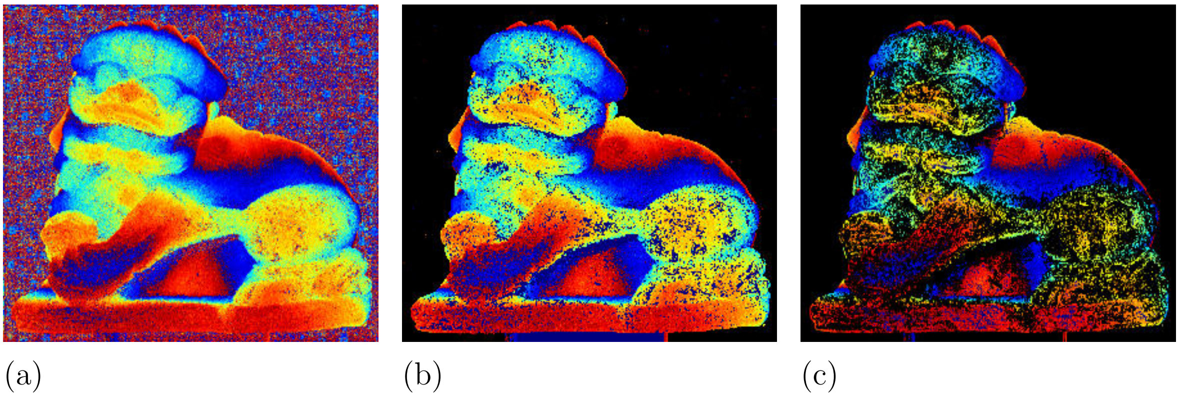


Figure 5.2: *The bare Structure Tensor result without any applied coherence threshold is shown in (a) while in image (b) a coherence threshold of $\eta_{th} = 0.85$ and in (c) of $\eta_{th} = 0.95$ is applied. Filtered disparity values become black.*

the deviation of hundreds of estimates with respect to the ground truth disparity. The related coherence values are shown to the right in 5.1 (a).

As first evaluation we apply a coherence threshold c and observe the influences to the precision of the Structure Tensor. We remove disparity values whose coherence values are below a predefined threshold. The result with respect to the old disparity error distribution is shown in figure 5.1 (b). An applied coherence threshold improves the resulting precision due to the removal of major outliers. We experimented with different distributions metrics on the underlying coherence distributions, but this did not yield any improvements. As one can see in table 5.1, larger threshold values improve the resulting precision but decrease the number of estimates, resulting more sparse disparity maps, as illustrated in figure 5.2. With this evaluation metric it seems difficult, unfortunately, to reliably discern erroneous disparity estimates from correct ones. To get an impression of this we image the computed disparity error for $d = 0.25\,\text{px}$ and $d = 0.5\,\text{px}$ together with the related coherence values, as shown in figure 5.3. It illustrates the correlation between thresholded values (red) and the related disparity estimations. Obviously, not only outliers are removed, but precise estimates as well.

Coherence Threshold	**Sobel**	**Scharr**	**Gaussian derivative**	
			$[3 \times 3]$	$[7 \times 7]$
no threshold	0.1149	0.0425	0.17	0.014
0.98	0.1149	0.0401	0.14	0.0132
0.998	0.1069	0.0306	0.09	0.0120
0.9995	0.0728	0.0168	0.025	0.0077

Table 5.1: *This table shows the precision increase for different applied coherence thresholds. The inner Gaussian filter is selected as $\sigma_{[3\times3]} = 0.4$ and the outer Gaussian filter as $\tau_{[3\times3]} = 0.6$. For the Gaussian derivative, two kernel sizes have been applied. The first $[3 \times 3]$ to compare with the Sobel and Scharr filters directly and the second $[7 \times 7]$ with a larger kernel.*

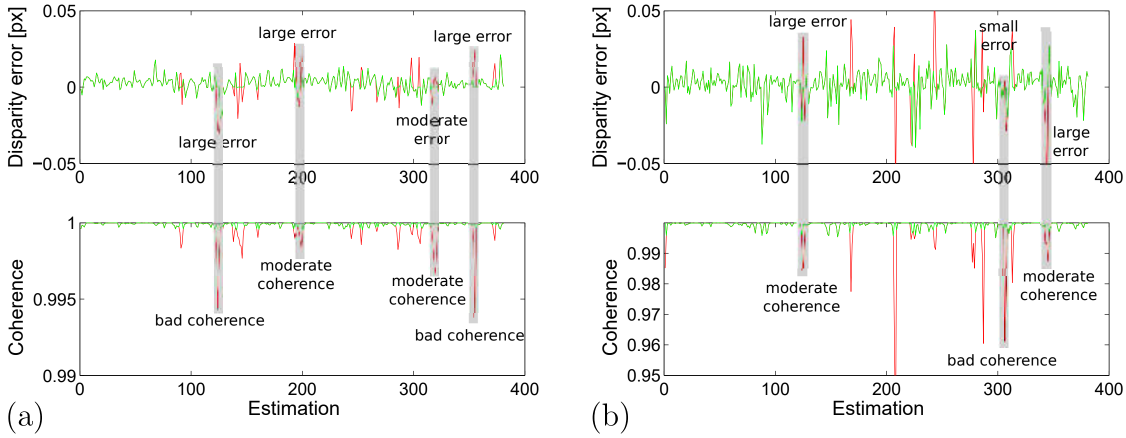


Figure 5.3: *This figure shows the disparity estimation error and the coherence value for (a)* $0.25\,px$ *and (b)* $0.5\,px$ *using the Scharr filter. The applied coherence threshold is* 0.9995*. Filtered pixels are shown red.*

This indicates that the estimation quality is only weakly correlated with the applied coherence measure.

By definition the coherence is defined as the quotient of the eigenvectors λ_1 and λ_2 of the processed Structure Tensor gradients ∂x and ∂s, as shown in figure 5.4. Thus the coherence can be alternatively expressed by

$$c = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \tag{5.1}$$

as described in Jähne [38]. That means coherence defines how well an orientation can be determined in the EPI and not how reliable it is with respect to the true orientation. Thus poor estimations can have high coherence values and good estimations can have

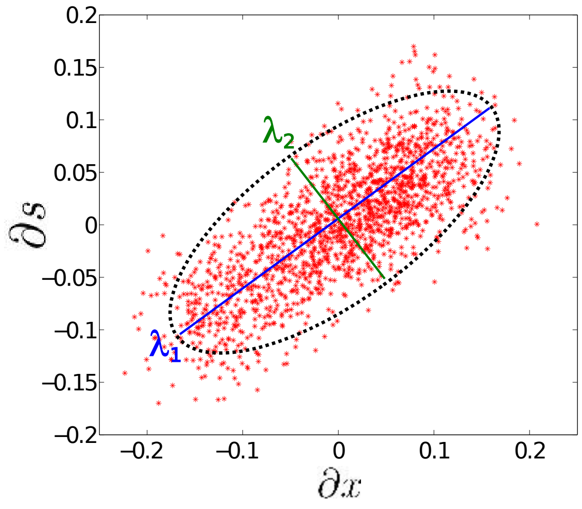


Figure 5.4: *Shows the derivative distribution of* ∂x *and* ∂s *for each point of a predefined orientation. The coherence is then defined by a ratio between the eigenvalues as given in equation 5.1.*

low coherence values.
Now considering a 4D light field such as a cross-shaped light field. In this kind of light field, the vertical and the horizontal directions are processed independently (see figure 5.5 (a)+(b)). For the common view, represented by the central image, it is possible to determine a superimposed disparity map by using the equation 2.10 (see figure 5.5 (c)).
Due to the weak correlation between coherence and disparity, the superimposition of the vertical and horizontal directions does not necessarily improve the resulting disparity map. In horizontal light fields vertical scene edges can be estimated perfectly, while horizontally aligned scene edges cannot be detected at all. In vertical light fields this property is reversed. Thus, at regions where the vertical and horizontal light field have different content, the superimposition obtains good results while in regions both direction has similar content the merging can also worsen the result.
As a final evaluation we compare the superimposed disparity maps achieved by selecting disparity based on coherence versus ground truth based on the minimal distance between estimation and ground truth. For this evaluation we define a cross-shaped light field, with its central image shown in figure 5.6 (c). As expected, the results show that vertical edges are selected from the horizontal light field and horizontal edges are selected from the vertical light field, see figure 5.6 (a). Aside from this, we see that in contrast to the selection with respect to the ground truth value as illustrated in figure 5.6 (b), the patches of the coherence merged result are larger and have a different distribution. In the case of the squared foreground object, which is barely visible, it also shows that an inverted selection with respect to the coherence would lead to better results.

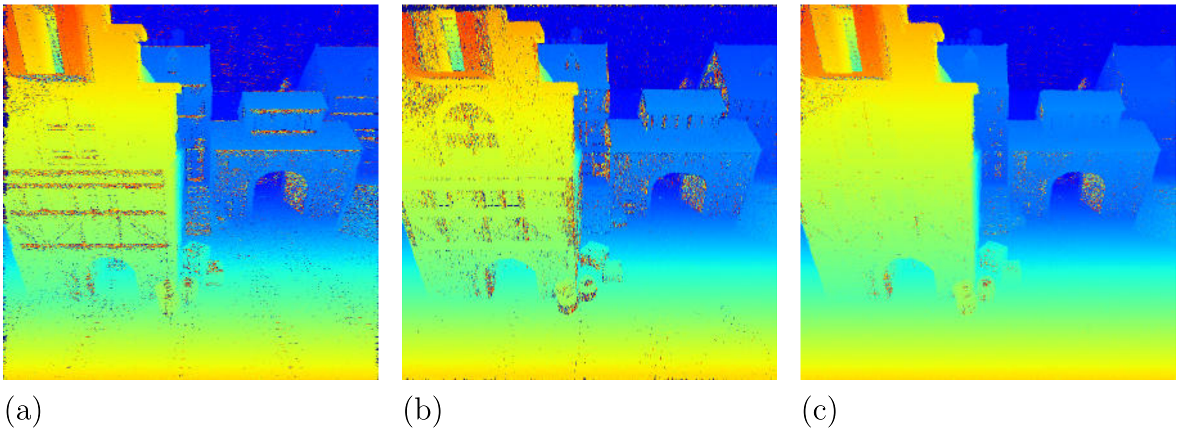

(a) (b) (c)

Figure 5.5: *(a) shows the resulting disparity map of a horizontal light field while (b) shows the resulting disparity map of a vertical light field. (c) shows the superimposed result using the coherence as decision criterion for merging.*

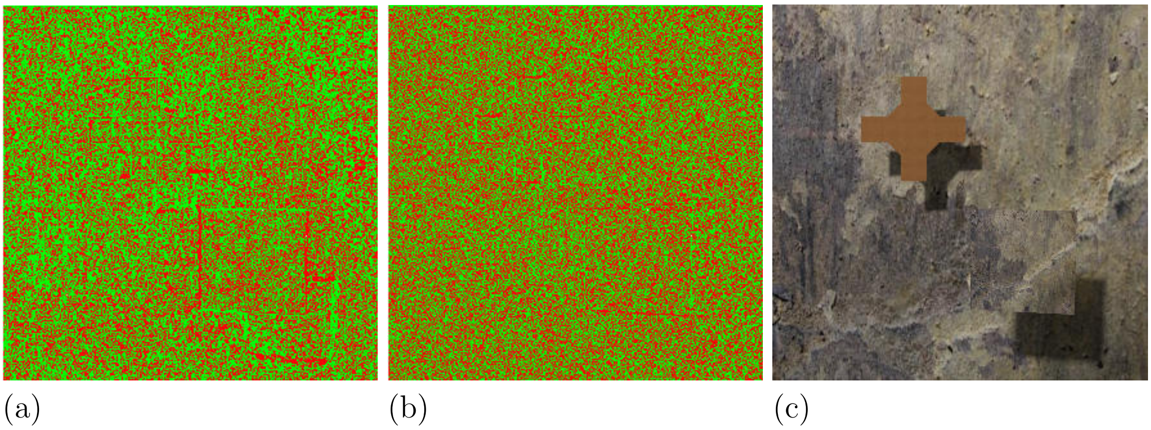


Figure 5.6: *(a) shows the selection decision with respect to the coherence map of the vertical and horizontal direction. In this image red is selected from the vertical direction and green from the horizontal direction. (b) shows the selection decision with respect to the ground truth disparity. In this image red is selected from the vertical direction and green from the horizontal direction. (c) shows the center view of the captured light field.*

5.1 Conclusion

The observed weak correlation persists into the global shifting process, where separate layers must be combined. Thus the coherence is predestined because only valid and invalid measurements need to be separated. A coherence threshold, improves the quality of the estimation visibly, since values not related to a valid orientation get canceled and thus the precision increases. Unfortunately it does not guarantee that poor estimates are also canceled. That's because the coherence only indicates,in how good an orientation can be detected but not in how good it represents the correct orientation. Coherence is our main tool in combining the vertical and horizontal directions. But with the shown weak correlation, it is clear that an improvement in this decision process could lead to better results.

6 Asymmetric Gaussian filter

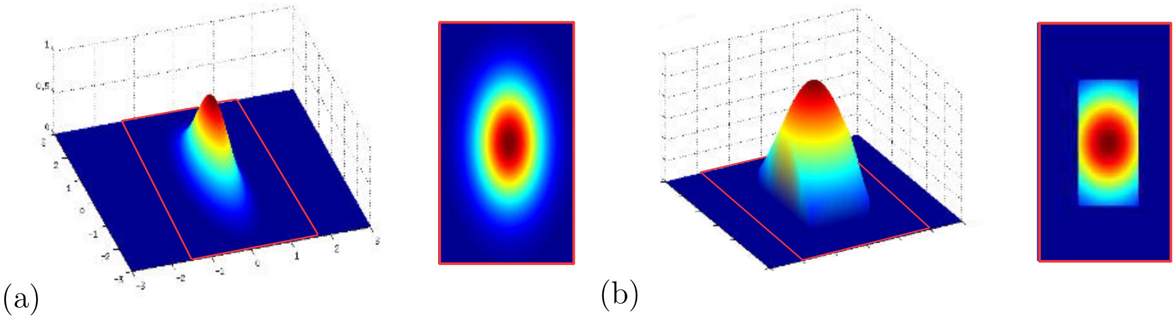


Figure 6.1: *(a) demonstrates the shape of an asymmetric Gaussian kernel. The shape of both directions are determined by the smoothing values $3\sigma_x$ and $3\sigma_s$. (b) shows a second kind of asymmetric Gaussian kernel where an initial symmetric Gaussian kernel is cut into an asymmetric shape.*

In the Structure Tensor orientation estimation two Gaussian filters are needed as shown in equation 2.6. The first Gaussian filter σ termed the inner acts as low pass filter. Since cameras capture spacial frequencies of arbitrary distances it is not avoidable that aliasing appears. That means, the remaining task of the inner Gaussian filter is to reduce occurring image noise. Thus the inner Gaussian filter represents a denoising filter. The second Gaussian filter τ termed outer is also a low pass filter. Since it is applied when the Structure Tensor components are already computed and noise is already removed its purpose is the weighted averaging of neighboring Structure Tensor components, which smooths the resulting disparity map. Thus the outer Gaussian filter represents an averaging filter. The shape of both Gaussian filters is assumed to be symmetric, but what happens when using asymmetric Gaussian kernel to estimate disparities. There are two possible asymmetric Gaussian kernel shapes which can be utilized for the inner or the outer filters. A short illustration about these two possible filter configurations is shown in figure 6.1. The first asymmetric Gaussian filter has a shape defined by a $3\sigma_i$ environment where $i \in \{x, s\}$ means the shape in image x and in camera direction s are independent. The second asymmetric Gaussian filter assumes a symmetric Gaussian kernel having one smoothing value σ while its shape is truncated with respect to the shape in image direction x_g and camera direction s_g.

In this chapter asymmetric filters are evaluated in their advantages and disadvantages for different filter sizes and shapes as well as for their applicability in light-field imaging. To aid in analyzing the influence of asymmetric kernels in disparity estimation, we use a test scene as shown in figure 6.2. While the underlying disparity remains constant in the EPI, color and spacial frequency change randomly. Using several different samples it is possible to attain highly reliable evaluations.

6.1 Inner Gaussian filter evaluation

For the precision evaluation of symmetric as opposed to asymmetric Gaussian filter, 200 arbitrary EPIs are analyzed. The precision result for the symmetric and the two different asymmetric cases is shown in figure 6.3. First, symmetric Gaussian filters having shapes of $[5\times5]$, $[7\times7]$ and $[9\times9]$, with a smoothing value σ steadily increasing, are analyzed as reference.
Second, asymmetric Gaussian filters having a $3\sigma_i$ shape are analyzed. Its shape defined by σ_x and σ_s changes accordingly to the underlying σ. The resulting shape is computed through:

$$\sigma_x = \sigma \tag{6.1}$$

$$\sigma_s = \sigma_{max} - \sigma \mid \sigma_{max} \in \{1, 1.5, 2.5\}. \tag{6.2}$$

where σ and the starting shapes are given as depicted in figure 6.3. Finally, the asymmetric Gaussian filters with a cropped shape of $[5 \times 12]$ and $[12 \times 5]$ are analyzed. As one can see, asymmetric filters defined by σ_x and σ_s lead to lower precisions than a symmetric kernel. Additionally, one can see that when these asymmetric kernels pass a symmetric shape they achieve the similar precision as well as its highest precision. On the other hand truncated asymmetric kernels achieve results similar to symmetric

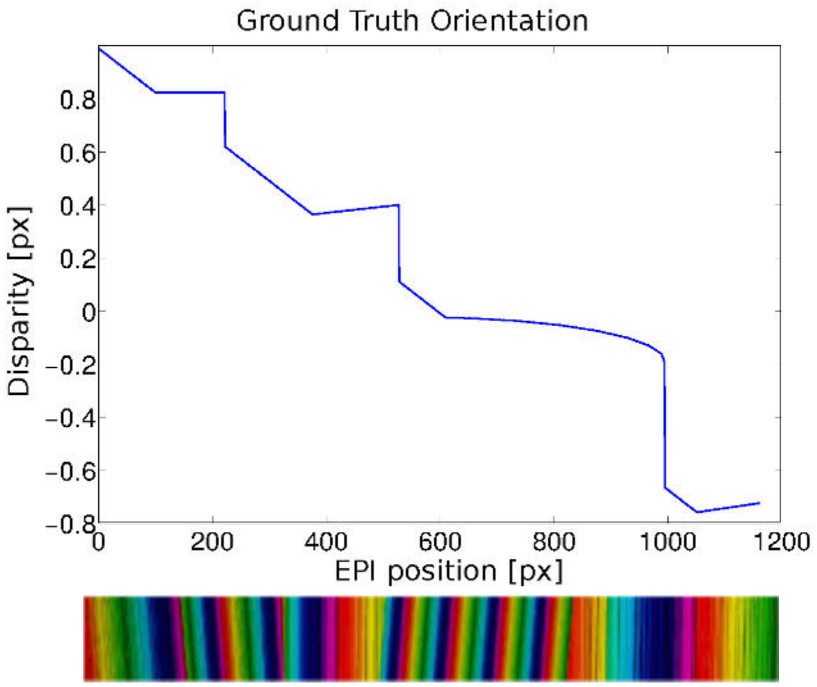


Figure 6.2: *Shown is a scene to a given depth profile, as shown in the top plot. The scene is represented by an EPI as shown below and used to evaluate the asymmetric Gaussian kernel. While the underlying disparity is not changing in the EPIs used for this analysis, the orientations spacial frequency and color can change randomly.*

kernels. For this reason, it can be considered to use them instead of their symmetric counterparts.

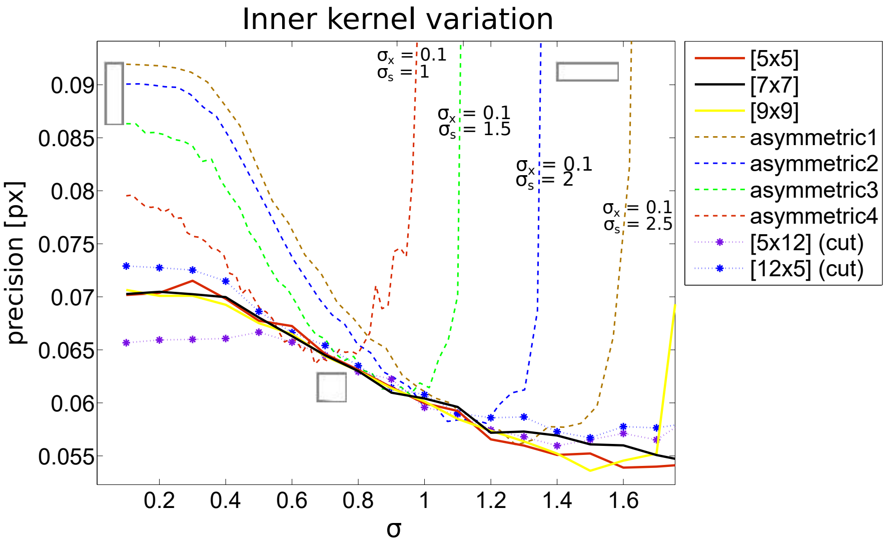


Figure 6.3: *The image shows the precision analysis for different Gaussian filter implementations. A symmetric and two types of asymmetric Gaussian filter are analyzed. The starting shape for asymmetric filter defined by σ_x and σ_s are shown in the plot by rectangular boxes. The analysis shows that the best precision is still achievable with symmetric filters.*

6.2 Outer Gaussian Filter evaluation

The outer Gaussian Filter is evaluated with the same strategy as the inner Gaussian filter. Shown in figure 6.4 are symmetric Gaussian kernels acting as precision references. We overlay the results of the asymmetric Gaussian kernel defined by σ_x and σ_s and the results of the truncated asymmetric Gaussian kernel defined by σ and its asymmetric shape. As one can see in figure 6.4, the influence on precision differs from that of the inner Gaussian filter, the estimation precision improves with increasing filter size. Nevertheless we see behavior for the outer Gaussian filter similar to that of the inner Gaussian filter. While asymmetric Gaussian filters defined by σ_x and σ_s behave even worse by having no clear influence, truncated asymmetric Gaussian kernels behave once again similar to the symmetric ones. Thus also here it can be considered to use truncated asymmetric rather than symmetric Gaussian kernels.

6.3 Object Transitions

In the evaluations presented above we determined that symmetric and truncated asymmetric filter both achieved good results. Now we want to analyze the behavior of object boundaries of these filter implementations. The assumed behavior is that an increasing kernel shape in the image direction results in worse transition between two different

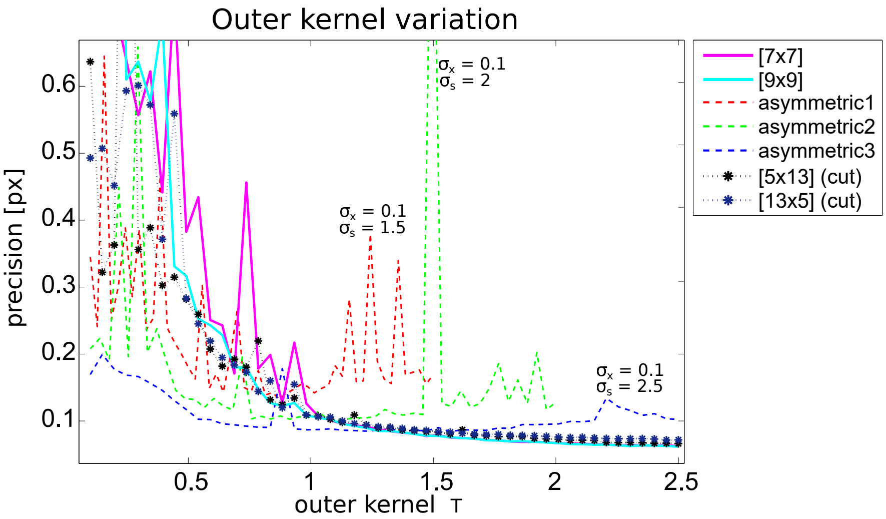


Figure 6.4: *The evaluation of the outer Gaussian filter leads to similar results as for the inner Gaussian filter evaluation. Symmetric kernels and cut asymmetric kernels achieve the best precision while asymmetric kernels defined by asymmetric Gaussian filter defined by σ_x and σ_s lead to worse precisions.*

disparities. That means to improve object boundaries we need to use truncated asymmetric kernels having a shape with $x_g < s_g$. This kind of asymmetry considers more information in the camera direction s_g than in the image direction x_g, which should lead to sharper object transitions. The result of this analysis is shown in figure 6.5. As one can see, the use of a truncated asymmetric Gaussian filter with $x_g < s_g$ leads to better object transitions but also increases the noise level.

This happens because fewer neighboring pixels are used in comparison with the symmetric Gaussian filter. For a similar number of pixels the observed noise level remains the same, which is implicitly proven by the averaging property of the outer Gaussian filter.

6.4 Conclusion

Truncated asymmetric Gaussian kernels achieve improved precision results while enhancing boundaries. Thus the usage of such filters presents a good alternative to symmetric Gaussian filters. The general inner Gaussian filter represents an anti-aliasing filter to reduce the noise level in the EPIs where a cut-off frequency defines the maximal allowed frequency. By definition the cut-off frequency becomes

$$f_c = \sqrt{2\ln(c)} * \sigma_f = \frac{1}{2s} \tag{6.3}$$

where c denotes the power reduction ratio in the power spectra and s the sampling frequency, which becomes the pixel pitch. To determine the matching value σ for the inner Gaussian filter to cut off at f_c, we have to take into account the standard deviation

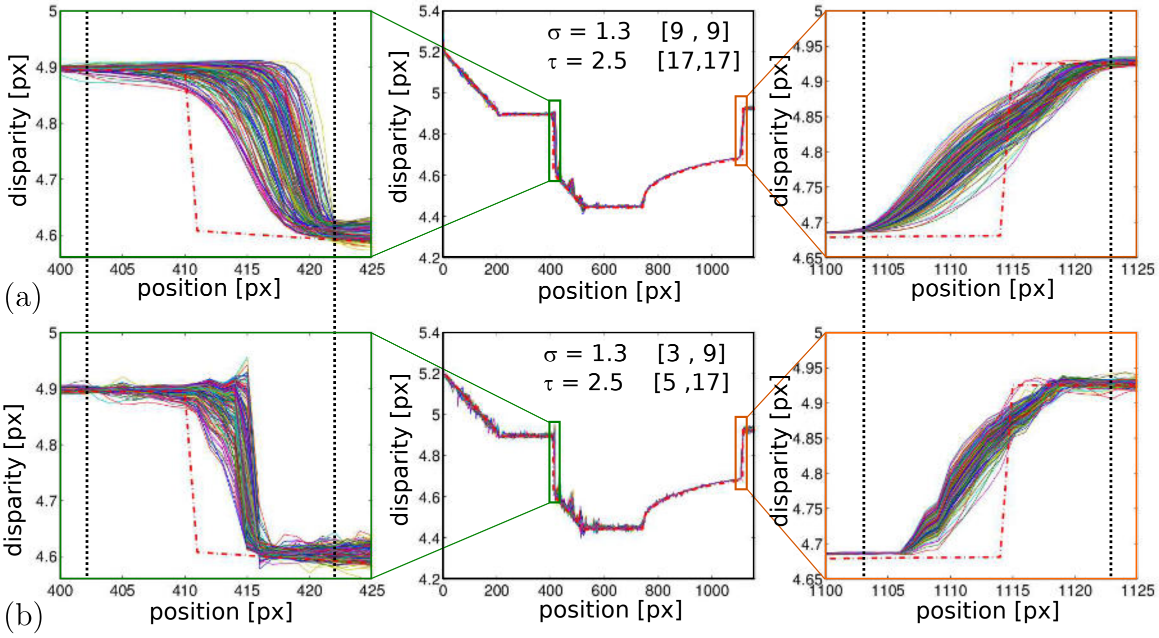


Figure 6.5: *(a) shows the evaluation of symmetric Gaussian filter. The inner Gaussian $\sigma = 1.3$ has a shape of $[9 \times 9]$ and the outer Gaussian filter $\tau = 2.5$ has a shape of $[17 \times 17]$ lead to boundary transitions as shown in the zoomed regions. (b) shows the evaluation of a truncated asymmetric Gaussian filter. The inner Gaussian $\sigma = 1.3$ has a shape of $[3 \times 9]$ while the outer Gaussian filter $\tau = 2.5$ has a shape of $[17 \times 5]$. That means the kernel shape in image direction is decreased which results in an improved boundary transition. Due to the reduced neighboring influence a higher noise ratio appears in contrast to the symmetric result visible in the increased number of spikes.*

σ_f of the Fourier transform of a Gaussian filter and the relation between both. This relation can be expressed by

$$\sigma \cdot \sigma_f = \frac{1}{2\pi}. \tag{6.4}$$

Inserting equations 6.3 into this expression leads to

$$\sigma = \frac{\sqrt{2\ln(c)}s}{\pi}. \tag{6.5}$$

In the image domain this becomes

$$\sigma_{px} = \frac{\sigma}{s} = \frac{\sqrt{2\ln(c)}}{\pi}. \tag{6.6}$$

The resulting standard deviation σ_{px} for a Gaussian filter having its cut-off frequency at the full-width half-maximum (FWHM) position is given for $c = 2$. Thus the standard deviation σ_{px} becomes 0.375 which implies a small filter shape.

For outer Gaussian filters a truncated asymmetric filter results in improved object transitions with similar precision. The usage of truncated asymmetric kernel for the outer Gaussian filter is recommended while its averaging value τ can be selected with respect to the smoothness desired in the result.

7 Orientation Analysis as Eigenvalue Problem Representation

The Structure Tensor analysis can also be viewed as Singular Value Decomposition (SVD) – also known as principal component analysis (PCA) – to compute the underlying orientations in an epipolar-plane image.

In this chapter we introduce PCA/SVD and Canonical Correlation Analysis (CCA) as alternative methods to estimate orientation. Using the SVD method within a defined evaluation window E_i in the EPI the right-singular vectors which relate to the underlying column space describe the orientation, as shown in figure 7.1.

Beside that we introduce the transition from evaluation window based PCA or CCA approaches to a pixel-based orientation estimation. We will see that pixel-based approaches lead to the Structure Tensor equations as introduced in chapter 2. In the precision evaluation of different approaches, we also evaluate advanced Structure Tensors which are able to distinguish between two orientations, such as transparent overlays or occluding orientations as introduced by T. Aach *et al.* [1].

Finally we introduce the second-order Structure Tensor and derive an improved Structure Tensor for single orientation estimation which yields a more robust estimate and adds the possibility of analyzing heterogeneous light-field structures which result from arrays of cameras having different modalities, e.g. varying exposure time from camera to camera.

7.1 Principal component analysis (PCA)

The PCA is mostly used in statistics to distinguish principal components v_t of data, where t defines the number of possible principal components. In orientation analysis we use principal component analysis to determine the underlying orientation inside an evaluation window E in the epipolar-plane image S. Within each evaluation window, having N pixels, we need to compute the derivatives $\frac{\partial D_i}{\partial x}$ and $\frac{\partial D_i}{\partial y}$ with $i \in N$. This derivative matrix H can be defined as

$$H = \left(E * \frac{\partial(\sigma * S)}{\partial x}, E * \frac{\partial(\sigma * S)}{\partial y} \right) = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{pmatrix} \quad \text{with} \quad h_i = (D_{x,i}, D_{y,i}) \tag{7.1}$$

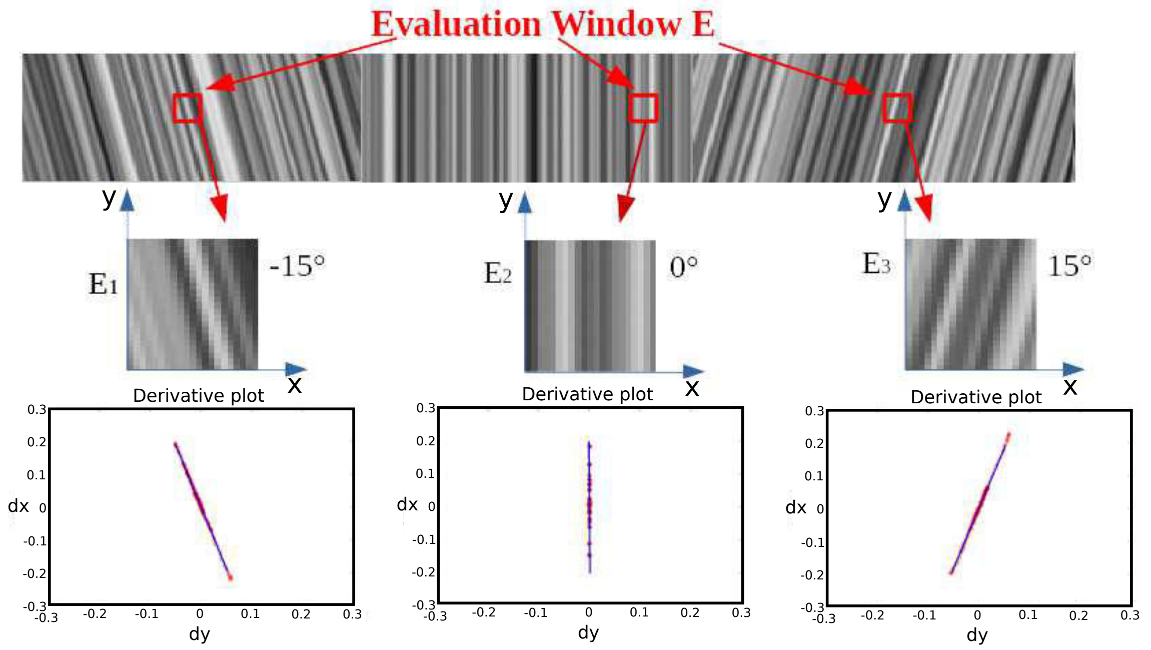


Figure 7.1: *Illustrates the principle of the orientation estimation using the principal component analysis. In the underlying EPI an evaluation window is convoluted with the EPI. Inside each evaluation window the orientation is estimated using the singular value decomposition.*

where σ is Gaussian smoothing to reduce noise and $D_{x,i}$, $D_{y,i}$ are the derivative values in the evaluation window. Direction derivatives in E are represented as column vectors. A visualization of these data is plotted with its principal components in figure 7.2. As one would suspect, the first principal component v_1 represents the normal direction of the underlying orientation and points in the direction with maximum variance σ_1 of the data.

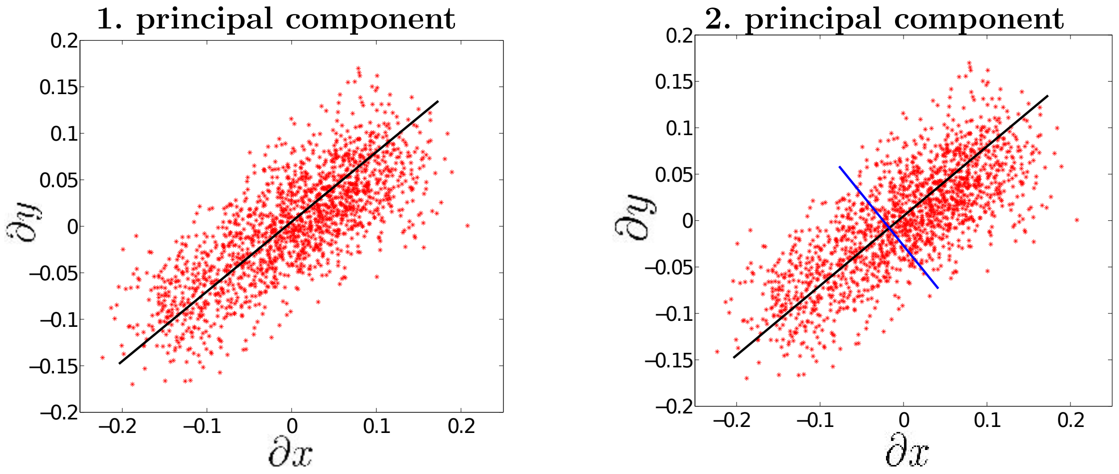


Figure 7.2: *(a) shows the first estimated principal component (black line). (b) shows the related second principal component (blue line), perpendicular to the first one.*

The first principal component is computed with the equation

$$v_1 = \arg \max_{||v_1||=1} \left(\sum_i (h_i v_1)^2 \right). \tag{7.2}$$

An equivalent matrix notation representation of this equation is given by

$$v_1 = \arg \max_{||v_1||=1} \left(||Hv_1||^2 \right) = \arg \max_{||v_{(1)}||=1} \left(v_1^T H^T H v_1 \right) \tag{7.3}$$

with the definition, that v_i has to be a unit vector, the equation also satisfies

$$v_1 = \arg\max \left(\frac{v_1^T H^T H v_1}{{v_1}^T v_1} \right). \tag{7.4}$$

For a symmetric matrix such as $H^T H$ the maximum value possible occurs when v is the eigenvector, and the quotient becomes the largest eigenvalue. The first principle component, represented with the first eigenvector is drawn in figure 7.2. To compute the next principle component we have to apply an orthonormal projection into the data space by removing the previous principal component using Gram-Schmidt

$$\hat{H} = H - \sum_{s=1}^{k-1} H v_s v_s^T. \tag{7.5}$$

SVD in PCA

The characteristic that the maximum argument is reached means that the solution of the equation is the largest eigenvalue (singular value) which is also the maximal value attainable. To apply the SVD we have to define the factorization of the matrix H as

$$H = U \Sigma V^* \tag{7.6}$$

where V contains the sought orientation vectors, because our initial matrix H contains all derivatives in its columns direction. This means each column is one space of the matrix H and the singular vectors sought are represented by the right singular vectors V, which are related to the column space. A detailed explanation about the factorization is given in the following section.

7.2 Introduction SVD

The singular value decomposition is used to compute the singular vector matrix $U_{[m \times m]}, V_{[n \times n]}$ and the singular value matrix $\Sigma_{[m \times n]}$ of a matrix $H_{[m \times n]}$ while m and n define the dimensions of the matrix. Every real or complex matrix possesses at least one singular-value decomposition which solves the equations

$$H\,v_i = \sigma_i u_i \qquad \text{or} \qquad H^* u_i = \sigma_i v_i. \tag{7.7}$$

where $i \in 1, ..., \min(m, n)$.
The matrix $H_{[m \times n]}$ itself is a composition of its left singular vectors $U_{[m \times m]}$, its rectangular diagonal matrix $\Sigma_{[m \times n]}$, containing all eigenvalues, and its right-singular vectors $V_{[n \times n]}$. This leads to the following representation

$$H = U\,\Sigma\,V^*. \tag{7.8}$$

with

$$U^* U = I \qquad \text{and} \qquad V^* V = I \tag{7.9}$$

where the columns of U are orthonormal eigenvectors of $H\,H^*$ and the columns of V are orthonormal eigenvectors of $H^* H$. The matrix Σ contains the square roots of the eigenvalues for $H\,H^*$ and $H^*\,H$. This can directly be used to solve the given eigenvalue problem from equation 7.7.
The two introduced matrices $H\,H^*$ and $H^* H$ formulate the core of the singular value decomposition and are of the following shape

$$H H^* = U\,\Sigma\,V^* V\,\Sigma^* U^* = U(\Sigma \Sigma^*) U^* \tag{7.10}$$

and

$$H^* H = V\,\Sigma^* U^* U\,\Sigma\,V^* = V(\Sigma^* \Sigma) V^*. \tag{7.11}$$

where V and U becomes the eigenvectors and Σ a diagonal matrix with the squares of the eigenvalues. To determine the eigenvalues and eigenvectors we have to solve the equations

$$\det(H\,H^* - sI) = 0 \qquad \text{and} \qquad \det(H^*\,H - sI) = 0 \tag{7.12}$$

to determine the related vectors V and U as well as the eigenvectors $\hat{\Sigma} = \sigma_1, .., \sigma_n$ which are the same for both matrices. The computed eigenvalues $\hat{\Sigma}$ are the squares of the sought singular values. The entries in the singular value matrix Σ are then the square roots of the computed eigenvalues.
In the following computations we are always interested in the right-singular vectors $V_{[n \times n]}$ and the related eigenvalues because they describe the orientation in the EPI, as explained earlier.

Reduced SVDs

In applications it is very unusual to use the full SVD, because of the increased memory and computation time. Thus, depending on the problem, reduced versions of the SVD are preferred. The first version is the thin SVD,

$$H = U_n \, \Sigma_n \, V^* \tag{7.13}$$

where only the first n components if $n < m$ of U corresponding to the row vectors of V^* are computed. Then U is $[n \times n]$, Σ is $[n \times n]$ and V remain the same size. The second reduced form is the compact SVD,

$$H = U_r \, \Sigma_r \, V_r^* \tag{7.14}$$

where only the r components with non zero singular values are computed. Thus U_r is $[m \times r]$, Σ_r is $[r \times r]$ and V_r is $[r \times n]$. The third and the most reduced form is the truncated SVD

$$\tilde{H} = U_t \, \Sigma_t \, V_t^* \tag{7.15}$$

where only the t largest singular values are computed. The matrix U_t becomes $[m \times t]$, Σ_r becomes $[t \times t]$ and V_t is $[t \times n]$. In the case of the orientation analysis $t = 1$, because we are just interested in the right singular vectors V_1 with the largest eigenvalue. Thus we achieve a reduction in SVD computation cost through focusing on the part needed.

7.3 Canonical-correlation analysis (CCA)

As introduced in section 7.1 we utilize the derivative matrix

$$H = \left(E * \frac{\partial(\sigma * S)}{\partial x}, E * \frac{\partial(\sigma * S)}{\partial y} \right) = (H_x, H_y) \tag{7.16}$$

to compute orientation with singular value decomposition. This is the first and simplest method to compute a single orientation contained in an evaluation window. Another method is the usage of covariance matrices

$$C(H_x, H_y) = \begin{pmatrix} var(H_x, H_x) & cov(H_x, H_y) \\ cov(H_y, H_x) & var(H_y, H_y) \end{pmatrix} \tag{7.17}$$

with

$$var(H_x, H_x) = \frac{\sum_{i=1}^{n} (H_{x,i} - H_{x,\mu})(H_{x,i} - H_{x,\mu})}{n} \tag{7.18}$$

and

$$cov(H_x, H_y) = \frac{\sum_{i=1}^{n} (H_{x,i} - H_{x,\mu})(H_{y,i} - H_{y,\mu})}{n}. \tag{7.19}$$

In the variance equation, the denominator is $n-1$ if the mean value of the underlying distribution is not known. Looking at figure 7.2, a zero mean assumption $H_\mu = (H_{x,\mu}, H_{y,\mu}) = (0,0)$ can be made. This means the denominator of the variance is n instead of $n-1$. This leads to the following vector notation

$$C(H_x, H_y) = \frac{1}{n}\begin{pmatrix} H_x^* H_x & H_x^* H_y \\ H_y^* H_x & H_y^* H_y \end{pmatrix} = \frac{1}{n} H^* H \tag{7.20}$$

which has beside a constant factor the same shape as the definition in equation 7.11 to compute the right singular values. This means, using the covariance matrix C, the problem to compute the orientation is reduced from an SVD to an eigenvalue problem.

$$det\left(C(H_x, H_y) - sI\right) = 0 \tag{7.21}$$

Next we want to reduce the evaluation window to a size of one pixel and apply a Gaussian filter to the resulting matrix. Then orientation evaluation becomes point wise and can be described with the following equation

$$J(D_x, D_y) = \tau * \begin{pmatrix} D_x^2 & D_x D_y \\ D_y D_x & D_y^2 \end{pmatrix} \tag{7.22}$$

with

$$D_x = \frac{\partial(\sigma * S)}{\partial x} \quad \text{and} \quad D_y = \frac{\partial(\sigma * S)}{\partial y} \tag{7.23}$$

which is the Structure Tensor representation. The closed form for estimating orientation is then given by the equation

$$d_{st} = \tan\left(\frac{1}{2}\arctan\left(2\frac{\hat{D}_x \hat{D}_y}{\hat{D}_x^2 - \hat{D}_y^2}\right)\right) \tag{7.24}$$

with

$$\hat{D}_x = D_x * \tau \quad \text{and} \quad \hat{D}_y = D_y * \tau. \tag{7.25}$$

7.3.1 Single Orientation estimation with PCA and CCA

Using the singular value decomposition and the canonical correlation analysis offers several different ways to compute orientation. We introduce these in the next subsection. In fact the SVD can be reduced to the same eigenvalue problem as described by the CCA, which represents a more practical and faster implementation. In addition, the reduction of the evaluation window and an application of a Gaussian filter leads to a pixel wise computation of the underlying orientation. The relation between the introduced methods is visualized in figure 7.3 In the next sections we introduce different methods to compute orientation in an EPI, using the SVD and the CCA. Both methods are nearly complementary in their precision but differ in their computational effort.

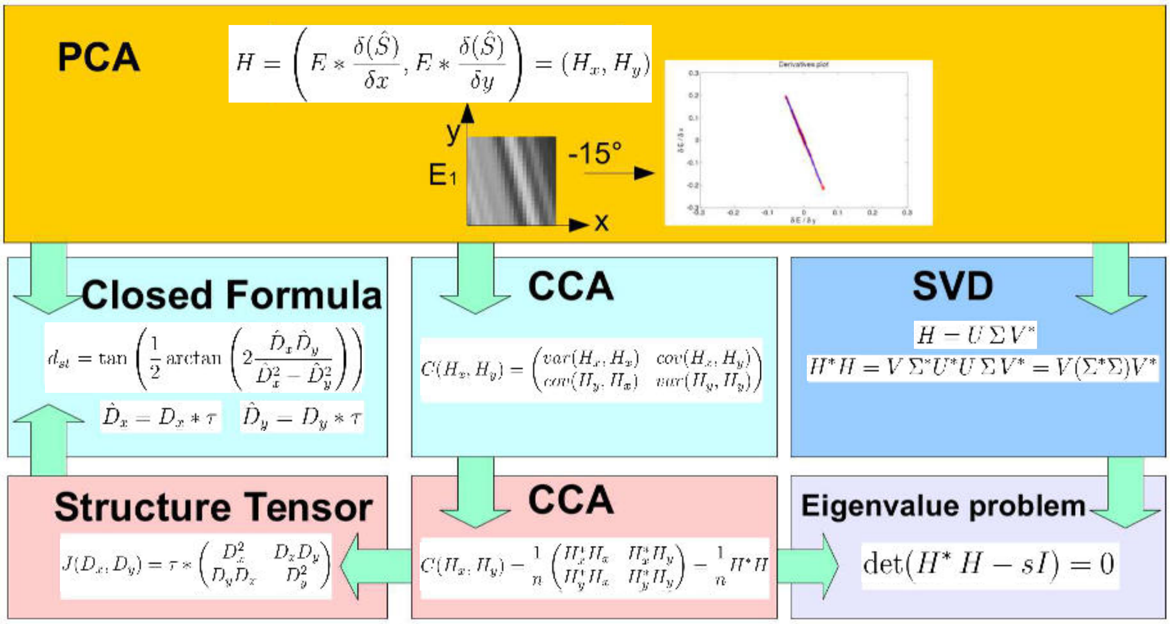

PCA
$H = \left(E * \frac{\delta(\hat{S})}{\delta x}, E * \frac{\delta(\hat{S})}{\delta y}\right) = (H_x, H_y)$
y
E1
-15°
x
Derivatives plot
Closed Formula
$d_{st} = \tan\left(\frac{1}{2}\arctan\left(2\frac{\hat{D}_x\hat{D}_y}{\hat{D}_x^2 - \hat{D}_y^2}\right)\right)$
$\hat{D}_x = D_x * \tau$
$\hat{D}_y = D_y * \tau$
CCA
$C(H_x, H_y) = \begin{pmatrix} var(H_x, H_x) & cov(H_x, H_y) \\ cov(H_y, H_x) & var(H_y, H_y) \end{pmatrix}$
SVD
$H = U \Sigma V^*$
$H^* H = V \Sigma^* U^* U \Sigma V^* = V(\Sigma^* \Sigma) V^*$
Structure Tensor
$J(D_x, D_y) = \tau * \begin{pmatrix} D_x^2 & D_x D_y \\ D_y D_x & D_y^2 \end{pmatrix}$
CCA
$C(H_x, H_y) = \frac{1}{n}\begin{pmatrix} H_x^* H_x & H_x^* H_y \\ H_y^* H_x & H_y^* H_y \end{pmatrix} = \frac{1}{n} H^* H$
Eigenvalue problem
$\det(H^* H - sI) = 0$

Figure 7.3: *Overview about correlation between the principal component analysis (PCA), the single value decomposition (SVD), the canonical correlation analysis (CCA) and the Structure Tensor.*

PCA with D_x and D_y

The resulting precisions $\sigma_{d_{PCA}}$ are computed with the PCA approach

$$H = \left(E * \frac{\partial(\sigma * S)}{\partial x}, E * \frac{\partial(\sigma * S)}{\partial y}\right) = (H_x, H_y) \tag{7.26}$$

by using two different box filter sizes.
The first box filter is $E_{[3\times3]}$ and the second is $E_{[7\times7]}$. Subscripts denote the box size. The resulting orientation d_{PCA} of this method is attained by computing the ratio between the two components of eigenvector V_1, and is described by the formula

$$d_{PCA} = \frac{V_{1,2}}{V_{1,1}} \qquad \text{with} \qquad V_1 = \begin{pmatrix} V_{1,1} \\ V_{1,2} \end{pmatrix} \tag{7.27}$$

The resulting precisions $\sigma_{d_{PCA}}$ are compared to the precision $\sigma_{d_{st}}$ attained with the closed form orientation analysis

$$d_{st} = \tan\left(\frac{1}{2}\arctan\left(\frac{2\hat{D}_x\hat{D}_y}{\hat{D}_x^2 - \hat{D}_y^2}\right)\right). \tag{7.28}$$

The precision results are shown in table 7.1.

$\sigma_{[3\times3]} = 0.4$	d_{st}	PCA $E_{[3\times3]}$	PCA $E_{[7\times7]}$
	$\sigma_{d_{st}}$	$\sigma_{d_{PCA}}$	$\sigma_{d_{PCA}}$
Sobel	0.073	0.062	0.044
Scharr	0.016	0.012	0.0061
Gaussian $[3\times 3]$	0.081	0.069	0.053
Gaussian $[7\times 7]$	0.0037	0.0026	0.00059

Table 7.1: *The table shows the precision $\sigma_{d_{st}}$ and $\sigma_{d_{PCA}}$ for different derivative filter and different box filter sizes. While the inner Gaussian filter stays constant, the outer Gaussian smoothing for d_{st} is set to $\tau_{[5\times5]} \;=\; 0.6$.*

PCA with D_x^2, D_xD_y and D_y^2

This method uses a modified version of the derivative vector from section 7.3.1. The new derivative vector evaluates orientation by the equation

$$H = \left(E*\left(\frac{\partial(\sigma*S)}{\partial x}\right)^2, E*\left(\frac{\partial(\sigma*S)}{\partial x}\frac{\partial(\sigma*S)}{\partial y}\right), E*\left(\frac{\partial(\sigma*S)}{\partial y}\right)^2\right) \quad (7.29)$$

$$= (H_{x,x}, H_{x,y}, H_{y,y}). \quad (7.30)$$

To analyze precision, the result of the Structure Tensor orientation estimation is computed first with equation 7.28 as reference value. Next, the precision for a box filter $E_{[3\times3]}$ and $E_{[7\times7]}$ is computed. After solving the eigenvalue problem the resulting eigenvector matrix V is a 3×3 matrix having the following shape

$$V = \begin{pmatrix} V_{1,1} & V_{2,1} & V_{3,1} \\ V_{1,2} & V_{2,2} & V_{3,2} \\ V_{1,3} & V_{2.3} & V_{3,3} \end{pmatrix} = (V_1, V_2, V_3). \quad (7.31)$$

The resulting eigenvector V_3 related to the smallest eigenvalue can be used to estimate occluded orientations which is introduced by Mülich *et al.* [62]. The first two dimensions of the eigenvector V_1, with respect to the largest eigenvalue, describe the single orientation in the EPI and can be computed using the formula

$$d = \frac{V_{1,1}}{V_{2,1}}. \quad (7.32)$$

The resulting precision of this method is shown in table 7.2.

$\sigma_{(3\times3)} = 0.4$	d_{st}	PCA $E_{[3\times3]}$	PCA $E_{[7\times7]}$
	$\sigma_{d_{st}}$	$\sigma_{d_{PCA}}$	$\sigma_{d_{PCA}}$
Sobel	0.07	0.067	0.058
Scharr	0.016	0.013	0.0052
Gaussian $[3 \times 3]$	0.077	0.075	0.062
Gaussian $[7 \times 7]$	0.0037	0.003	0.0008

Table 7.2: *The table shows the precision $\sigma_{d_{st}}$ and $\sigma_{d_{PCA}}$ for different derivative filter and different box filter sizes. Chosen is a constant inner Gaussian smoothing. The outer Gaussian smoothing used for d_{st} is set to $\tau_{[5\times5]} = 0.6$.*

CCA with D_x and D_y

The next method uses the canonical correlation analysis to estimate orientation. This evaluation uses the derivative vector H from equation 7.16 and computes the covariance matrix for all values inside the evaluation window $E_{[3\times3]}$

$$C(H_x, H_y) = \frac{1}{n}\begin{pmatrix} H_x^* H_x & H_x^* H_y \\ H_y^* H_x & H_y^* H_y \end{pmatrix} = \frac{1}{n} H^* H. \tag{7.33}$$

Next we reduce the size of the evaluation window $E_{[1\times1]}$ to one pixel. This reduction modifies the CCA in a way that, with an additional Gaussian Filter τ, it becomes the Structure Tensor

$$J(D_x, D_y) = \tau * \begin{pmatrix} D_x^2 & D_x D_y \\ D_y D_x & D_y^2 \end{pmatrix} \tag{7.34}$$

with

$$D_x = \frac{\partial(\sigma * S)}{\partial x} \quad \text{and} \quad D_y = \frac{\partial(\sigma * S)}{\partial y}. \tag{7.35}$$

The resulting orientation is also given by the eigenvector V_1 with respect to the largest eigenvalue. Thus the disparity can be computed using equation 7.27. The results of the precision evaluation are shown in the next table 7.3.

CCA with D_x^2, $D_x D_y$ and D_y^2

Now we want to compute the single orientation using the CAA applied to the derivative vector H introduced in equation 7.30. The resulting covariance matrix $C(H_x, H_y)$ used for the orientation estimation with a given evaluation window E has the following shape

$$C(H_x, H_y) = \frac{1}{n}\begin{pmatrix} (H_{x,x}^* H_{x,x}) & (H_{x,x}^* H_{x,y}) & (H_{x,x}^* H_{y,y}) \\ (H_{x,x}^* H_{x,y}) & (H_{x,y}^* H_{x,y}) & (H_{x,y}^* H_{y,y}) \\ (H_{x,x}^* H_{y,y}) & (H_{x,y}^* H_{y,y}) & (H_{y,y}^* H_{y,y}) \end{pmatrix} = \frac{1}{n} H^* H. \tag{7.36}$$

$\sigma_{[3\times3]} = 0.4$	d_{st}	PCA $E_{[3\times3]}$	PCA τ
	$\sigma_{d_{st}}$	$\sigma_{d_{PCA}}$	$\sigma_{d_{PCA}}$
Sobel	0.069	0.093	0.069
Scharr	0.013	0.027	0.017
Gaussian $[3\times3]$	0.074	0.112	0.071
Gaussian $[7\times7]$	0.0034	0.0065	0.0036

Table 7.3: *The table shows the precision $\sigma_{d_{st}}$ and $\sigma_{d_{PCA}}$ for different derivative and box filter sizes. The inner Gaussian smoothing is $\sigma_{[3\times3]} = 0.4$. While the outer Gaussian smoothing, used for d_{st} and PCA τ, is set to $\tau_{[5\times5]} = 0.6$.*

For the second evaluation we reduce the evaluation window $E_{[1\times1]}$ again to one pixel. Then the covariance matrix also becomes a Structure Tensor. This Structure Tensor can be used to estimate double orientations in the occlusion case, as discussed in the PCA subsection. The Structure Tensor is given by the equation

$$J(D_x, D_y) = \tau * \begin{pmatrix} D_x^4 & D_x^3 D_y & D_x^2 D_y^2 \\ D_x^3 D_y & D_x^2 D_y^2 & D_x D_y^3 \\ D_x^2 D_y^2 & D_x D_y^3 & D_y^4 \end{pmatrix} \tag{7.37}$$

with

$$D_x = \frac{\partial(\sigma * S)}{\partial x} \quad \text{and} \quad D_y = \frac{\partial(\sigma * S)}{\partial y} \tag{7.38}$$

The resulting eigenvectors are three-dimensional and the eigenvector V_1, related to the largest eigenvalue, describes the underlying orientation in its first two components, as introduced in equation 7.32. The results of the precision analysis are shown in table 7.4.

$\sigma_{[3\times3]} = 0.4$	d_{st}	PCA $E_{[3\times3]}$	PCA τ
	$\sigma_{d_{st}}$	$\sigma_{d_{PCA}}$	$\sigma_{d_{PCA}}$
Sobel	0.07	0.07	0.067
Scharr	0.016	0.021	0.013
Gaussian $[3\times3]$	0.076	0.079	0.075
Gaussian $[7\times7]$	0.0037	0.005	0.003

Table 7.4: *The table shows the precision $\sigma_{d_{st}}$ and $\sigma_{d_{PCA}}$ for different derivative and box filter sizes. The inner Gaussian smoothing remains constant for all measurements while the outer Gaussian smoothing used for d_{st} and the PCA τ is set to $\tau_{[5\times5]} = 0.6$.*

CCA with D_{xx}^2, D_{xy} and D_{yy}

As a final method, we wish to take advantage of the second order derivatives. They can be used either to separate two transparent overlying orientations, as introduced in

Wanner *et al.* [91], or to estimate a single orientation. Thus we introduce the derivative vector

$$\begin{aligned} H &= \left(E * \frac{\partial^2 \hat{S}}{\partial x^2}, E * \frac{\partial^2 \hat{S}}{\partial x \partial y}, E * \frac{\partial^2 \hat{S}}{\partial y^2} \right) && (7.39) \\ &= \left(H_{xx}, H_{xy}, H_{yy} \right). && (7.40) \end{aligned}$$

Next, we directly apply the canonical correlation analysis to the defined derivative vector. The resulting covariance matrix becomes

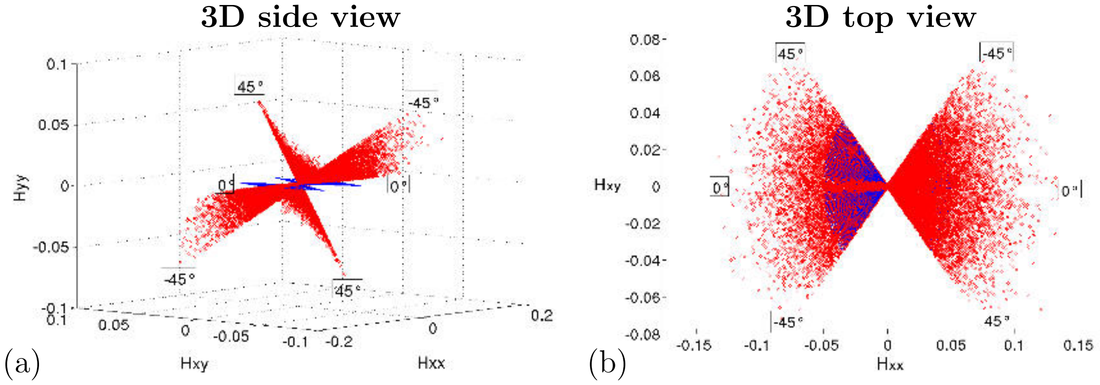
$$C(H_{xx}, H_{xy}, H_{yy}) = \frac{1}{n} \begin{pmatrix} (H^*_{xx} H_{xx}) & (H^*_{xx} H_{xy}) & (H^*_{xx} H_{yy}) \\ (H^*_{xx} H_{xy}) & (H^*_{xy} H_{xy}) & (H^*_{xy} H_{yy}) \\ (H^*_{xx} H_{yy}) & (H^*_{xy} H_{yy}) & (H^*_{yy} H_{yy}) \end{pmatrix} = \frac{1}{n} H^* H. \tag{7.41}$$

Reducing this matrix to a pixel-wise evaluation and an additional Gaussian filter τ leads to the following Structure Tensor

$$J(D_{xx}, D_{xy}, D_{yy}) = \tau * \begin{pmatrix} D_{xx} D_{xx} & D_{xx} D_{xy} & D_{xx} D_{yy} \\ D_{xx} D_{xy} & D_{xy} D_{xy} & D_{xy} D_{yy} \\ D_{xx} D_{yy} & D_{xy} D_{yy} & D_{yy} D_{yy} \end{pmatrix} \tag{7.42}$$

with

$$D_{xx} = \frac{\partial^2 (\sigma * S)}{\partial x^2} \quad , \quad D_{yy} = \frac{\partial^2 (\sigma * S)}{\partial y^2} \quad \text{and} \quad D_{xy} = \frac{\partial^2 (\sigma * S)}{\partial x \partial y}. \tag{7.43}$$


3D side view

3D top view

Figure 7.4: *(a) shows the 3D visualization of the second order Structure Tensor. All derivative values of H are located on a cone surface (red dots). The blue lines represent the orientation estimation solution. (b) shows the top view of the 3D graph. It shows that the 2D projection of the data matches with the single orientation.*

To analyze the behavior of this Structure Tensor for each orientation we use an evaluation EPI as shown in figure 7.5. The visualization of the geometrical orientation of the H vector components in a 3D space is seen in figure 7.4. This visualization contains the derivative values, represented as red dots for each pixel in the small windows, and shows the estimated orientation of the Structure Tensor for the central pixel as blue lines. The top view of the 3D space reveals that a 2D projection of the 3D data already contains the single orientation. Thus, the single orientation is computable independently from the third dimension of the H vector. That leads to a new derivative vector

$$H \;=\; \left(E * \frac{\partial^2 \hat{S}}{\partial x^2}, E * \frac{\partial^2 \hat{S}}{\partial x \partial y}\right) = (H_{xx}, H_{xy})\,. \tag{7.44}$$

With the dimension reduced H vector the covariance matrix becomes

$$C(H_{xx}, H_{xy}) = \frac{1}{n}\begin{pmatrix}(H^*_{xx}H_{xx}) & (H^*_{xx}H_{xy})\\ (H^*_{xx}H_{xy}) & (H^*_{xy}H_{xy})\end{pmatrix} = \frac{1}{n}H^*H. \tag{7.45}$$

Decreasing the evaluation window to one pixel and the application of Gaussian smoothing τ leads to a Structure Tensor representation defined by

$$J(D_{xx}, D_{xy}) = \tau * \begin{pmatrix}D_{xx}D_{xx} & D_{xx}D_{xy}\\ D_{xx}D_{xy} & D_{xy}D_{xy}\end{pmatrix}. \tag{7.46}$$

The shape of this Structure Tensor is similar to the first introduced Structure Tensor from equation 7.28. The only difference is an additional derivative filtering in the x direction. Taking this into account, we select a derivative filter with $(2R-1)$ elements.

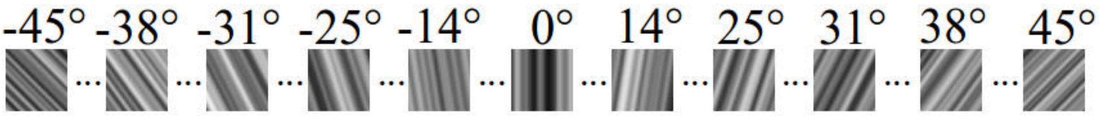


Figure 7.5: *EPIs used to compute the derivative of the second order Structure Tensor and represent it in a 3D space as seen in figure 7.4.*

The advantage of an odd symmetry filter with $(2R-1)$ elements is shown by its transfer function as describe by Jähne [38]. Thus, the transfer function for the smallest possible derivative filter becomes

$$\mathcal{D} \;=\; \frac{1}{2}[-1\,0\,1] \;\circ\!\!-\!\!\!-\!\!\bullet\; \cos(\pi\hat{k}) \qquad 0 \leq \hat{k} \leq 1 \tag{7.47}$$

where $\hat{k}$ denotes the normalized wave number. As one can see, this filter attenuates both low frequencies and high frequencies. With this understanding, the usage of an inner Gaussian filter becomes obsolete since its main purpose was anti-aliasing, noise removal and value averaging, all of which, aside from the averaging, is now done by the additional derivative filter. The averaging is transferred to the outer Gaussian filter, which has negligible effect on its value, and its shape doesn't change by the defined offset. This makes the entire processing not only faster, due to the removal of the inner

	d_{st}	d'_{st}	d_{impr}	d'_{impr}
	Δd_{st}	$\Delta d'_{st}$	Δd_{impr}	$\Delta d'_{impr}$
Sobel	0.071	0.065	0.07	0.064
Scharr	0.016	0.007	0.012	0.0067
Gaussian $[3 \times 3]$	0.075	0.071	0.074	0.069
Gaussian $[7 \times 7]$	0.0037	0.0005	0.001	0.00049

Table 7.5: *The table shows the precision Δd_{st} and Δd_{impr} for different derivative filter and Gaussian smoothing values. For d_{st} and d_{impr} we use an inner Gaussian smoothing of $\sigma_{[3\times3]} = 0.4$ and an outer Gaussian smoothing $\tau_{[5\times5]} = 0.6$. For d'_{st} and d'_{impr} we use an inner Gaussian smoothing with $\sigma_{[3\times3]} = 0.7$ and and outer Gaussian smoothing with $\tau_{[5\times5]} = 1.6$.*

Gaussian filter, but also applicable to heterogeneous light fields. Thus the new derived Structure Tensor becomes

$$J = \tau * \begin{pmatrix} \left(\frac{\partial \hat{S}}{\partial x}\right)^2 & \frac{\partial \hat{S}}{\partial x} \cdot \frac{\partial \hat{S}}{\partial s} \\ \frac{\partial \hat{S}}{\partial s} \cdot \frac{\partial \hat{S}}{\partial x} & \left(\frac{\partial \hat{S}}{\partial s}\right)^2 \end{pmatrix} =: \begin{pmatrix} \hat{J}_{xx} & \hat{J}_{xs} \\ \hat{J}_{xs} & \hat{J}_{ss} \end{pmatrix} \tag{7.48}$$

with the abbreviation

$$\hat{S} := \frac{\partial S}{\partial x}. \tag{7.49}$$

8 Heterogeneous Light Fields

In contrast to traditional binocular or multi-view stereo approaches, the redundant sampling of light-field imaging (i.e. more than two views for triangulation) allows one to obtain dense and high quality depth maps with significant increase in accuracy and reliability. It also extends capabilities beyond those of traditional analysis methods. For example, previously, a constant intensity has been assumed for estimating disparity from orientations in most approaches to analysis of epipolar-plane images (EPIs). Here, we introduce an adapted structure-tensor approach which improves depth estimation. This extension also includes a model of non-constant intensity on EPI manifolds. We derive an approach to estimate high quality depth maps in luminance gradient light fields, as well as in color-filtered light fields. Color-filtered light fields pose particular challenges due to the fact that structures can change significantly in appearance or completely vanish with wavelength. We demonstrate solutions to this challenge and obtain a dense sRGB image reconstruction in addition to dense depth maps. This and the next chapter were submitted for consideration to the IEEE CVPR 2016 conference.

8.1 Introduction

The basis of light-field [27] imaging is the plenoptic function as introduced in [2]. It represents a multi-dimensional function describing all the information available of light reflected from a scene. This comprises the direction and spectral radiance of the light. To capture light fields, the plenoptic function is simplified in its dimensionality to a four dimensional subspace, at times termed the lumigraph. This light field representation was first introduced in computer graphics by both Gortler *et al.* [30] and Levoy *et al.* [49]. The lumigraph describes the ray path parameterized by two parallel planes. Along the ray path, radiance remains constant. More generally, every ray leaving a particular surface point appears the same – that is, the appearance of the surface point is independent of the perspective from which it is viewed. This is referred to as exhibiting Lambertian behavior. Due to this constraint, most methods to compute disparities such as [91, 43, 14, 20, 18] relate only to light fields having this property. Even for binocular or multi-view stereo approaches, as proposed in [50, 41, 22, 70] correspondence between points is modeled by having the same appearance from any view.

Thus, current cameras such as Raytrix [66] and Lytro [26] focus on reconstruction from input images having similar color information. A violation, in case of images captured with different color filters or illuminations, needs sophisticated pre-processing algorithms to adapt the data for the depth estimation, as in the work of Yong *et al.* [34]. In that approach, the input image data is mapped to a log-chromaticity color space

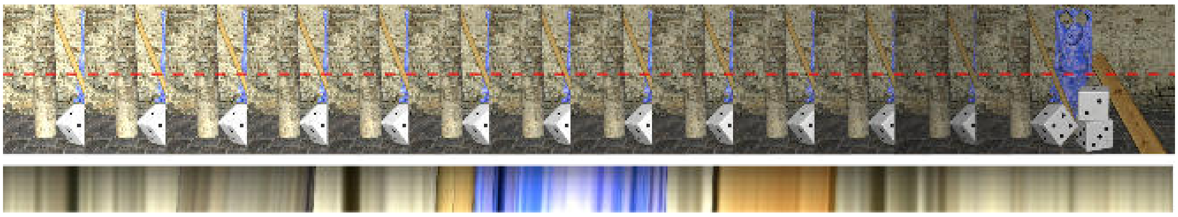


Figure 8.1: *This EPI has a linear illumination gradient in vertical direction. The first row of the shown EPI related to the first captured image of the heterogeneous light field while the last row is related to the last captured image. As one can see the illumination increases continuously. The red line shows the position where the shown EPI is extracted.*

to obtain an illumination-independent color representation for finding corresponding points in the input data. Thus neither multi-view stereo nor current light-field imaging address the direct computation of depth maps from heterogeneous input data. Such heterogeneous, or hyper-spectral, images may be generated using a single camera with revolving color filters before the objective, as described in Tominaga [85]. Unfortunately with this setup, it is not possible to make depth estimates on the underlying scene. Here, we present heterogeneous light fields which have properties that change between captured images – such as with the presence of illumination gradients or the application of colored filters. We demonstrate that a modified structure tensor is able to process heterogeneous light fields. We analyze the limits of both illumination gradients and randomly illuminated light fields. Furthermore, we show that even for color filtered light fields the structure tensor approach computes highly reliable depth information locally which can be merged to a dense depth map. With this dense depth map it is possible to compute a hyper-spectral image with respect to a reference view out of the used color filtered light field. To visualize the obtained hyper-spectral image we introduce a method to approximate the sRGB color space from the hyper-spectral information and display the final RGB reconstruction.

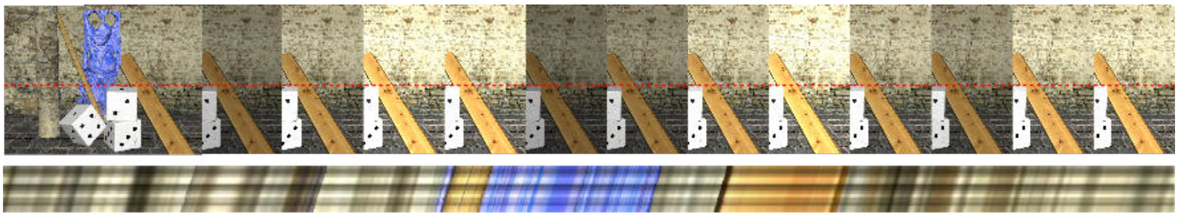


Figure 8.2: *This EPI has a random distributed illumination in vertical direction. The first row of the shown EPI is related to the first captured image of the heterogeneous light field while the last row is related to the last captured image. The red line shows the position where the shown EPI is extracted.*

8.2 Illumination gradient light field

In this section we analyze the precision attained with the newly defined Structure Tensor from equation 7.48 for different derivative filters in contrast to the traditional Structure Tensor. For this, we compute the precision as described in Diebold *et al.* [20] and discussed in the last chapter. For evaluation purpose several synthetic EPIs where generated, covering a discretized angular space between $[-45°, 45°]$ as shown in figure 7.5. The resulting overall precision for all evaluated orientations $i \in N$ becomes

$$\sigma_d = \sqrt{\frac{1}{N}\sum_i^N (\mu_i)^2 + \frac{4}{N}\sum_i^N (\sigma_i)^2} \tag{8.1}$$

where σ_i defines the standard deviation of the evaluated estimations and μ_i denotes the mean values. We additionally consider systematic errors that arise. The results of the precision analysis are shown in table 8.1. As one can see, precision increases, but the question remains of how it changes under luminance-gradient light fields.

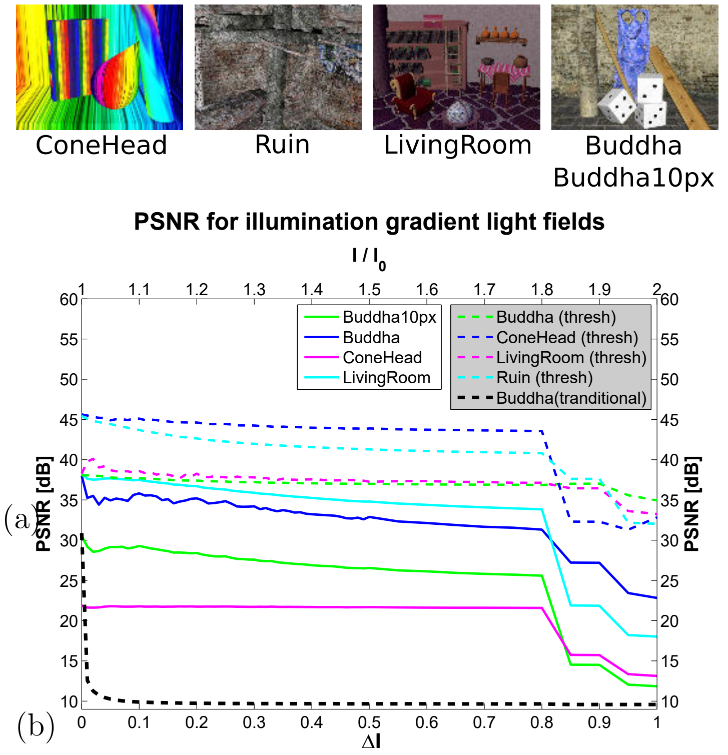


Figure 8.3: *Shows the PSNR of the shown scenes (a) for different applied illumination gradients ΔI. Figure (b) shows the PSNR applied on the entire disparity map as well as only to coherence thresholded $\theta = 0.8$ values. For the Buddha scene the traditional implementation of the structure tensor is shown as black dashed line. As one can see, the PSNR remains almost constant until a gradient limit is reached, here around $\Delta I = 0.85$. In contrast, the PSNR of the traditional structure tensor drops down instantly.*

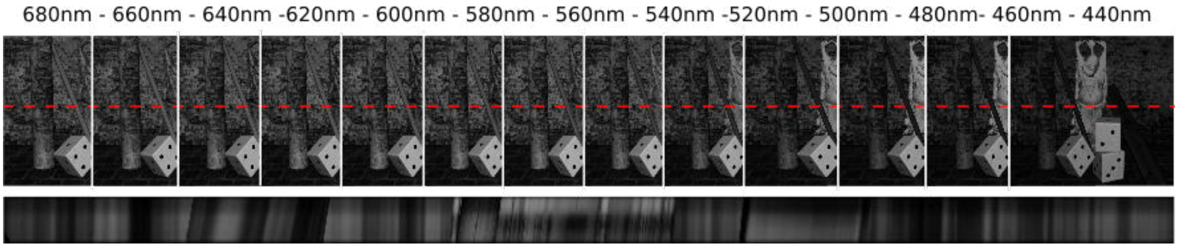

680nm - 660nm - 640nm -620nm - 600nm - 580nm - 560nm - 540nm -520nm - 500nm - 480nm- 460nm - 440nm

Figure 8.4: *Shows an example of a linear heterogeneous light field. Each image is captured with a different color filter having a full width half mean of* 10 *nm. The red line shows the position, where the shown EPI is extracted. In color filtered light fields the intensity changes in orientation direction with respect to the underlying color content.*

In luminance-gradient light fields, the illumination changes from image to image, which is termed illumination gradient ΔI in the following. To analyze these kinds of heterogeneous structures, we apply an illumination gradient ΔI to synthetically rendered light fields as shown in figure 8.3. An example of a luminance-gradient light field and the appearance of a resulting EPI are shown in figure 8.1. To compare the reliability of the Structure Tensor for different illumination gradients we compute for each evaluation scene the PSNR with respect to the ground truth disparity map. The results for different ΔI are shown in figure 8.3. As one can see, the new Structure Tensor keeps an almost constant PSNR until the illumination gradient reaches a limiting value. From there, the PSNR immediately decreases. The entire evaluation is made with 16bit images since it is essential to avoid saturation. In the event of saturated regions, the PSNR will decrease due to missing orientation information, and not because of the applied gradient. Thus, 16 bit images allow isolating the influence of the illumination gradient on the estimation result. Aside from the PSNR of the improved Structure Tensor, the PSNR of the traditional Structure Tensor of the Buddha scene is shown in figure 8.3. As one can directly see, the traditional Structure Tensor is not able to process a heterogeneous light field. Furthermore, we analyze heterogeneous light fields having random illumination distributions as shown in figure 8.2. Here, we randomly

Derivative Filter	σ_d for J_1	σ_d for J_2	σ_d for J_3
Sobel	0.0588	0.0322	0.0315
Scharr	0.0299	0.0082	0.0085
Gaussian 3x3	0.0698	0.0321	0.0326

Table 8.1: *The table shows the resulting precision of the traditional structure tensor J_1 proposed by Wanner [92] in comparison to the new structure tensor with an applied inner Gaussian filter J_2, and without an additional inner Gaussian Filter J_3. The evaluation is made for different possible derivative filters of the same shape. As one can see, the new structure tensor outperforms in the Scharr filter implementation the traditionally structure tensor by far.*

shuffle the image related multiplier which was used to achieve the illumination gradient in the EPI. The results of this evaluation are shown in figure 8.5, illustrating the applicability to acquired light fields. Small illumination variations invariably occur in acquired light fields. Illumination differences appear due to flickering of the light source or because of varying camera properties across the light field array, i.e. exposure time or sensitivity. Thus, the designed Structure Tensor not only improves the estimation result in homogeneous light fields but also makes it possible to process heterogeneous light fields.

8.3 Color-filtered light fields

In this section we introduce color-filtered light fields as shown in figure 8.4. Color-filtered light fields are captured with color filters of different wavelengths so that each image of a light field contains differing color information. The used band-pass filters have a full-width half-mean of 10 nm and are uniformly distributed in the color spectrum between 400 nm and 700 nm. This means the color-filtered light field contains the full spectral information of the underlying scene distributed over 31 images. After generating synthetic color-filtered light fields, we apply the introduced structure tensor J_3 to test scenes such as the rainbow textured scene shown in figure 8.6 (a). Due to

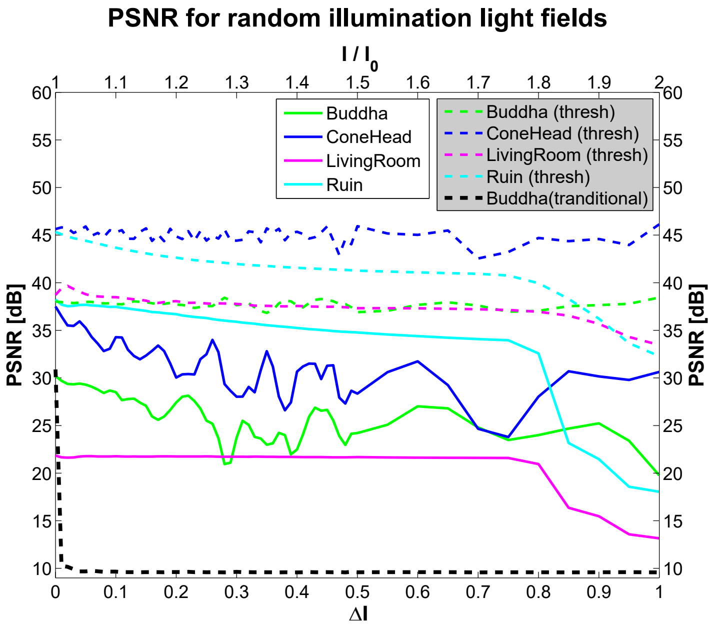


Figure 8.5: *Illustrates the PSNR for different applied randomly shuffled illuminations defined by an underlying illumination gradient ΔI. The figure shows the PSNR calculated for the entire disparity map as well as only for coherence thresholded $\theta = 0.8$ values. For the Buddha scene the traditional implementation of the Structure Tensor is shown as black dashed line. As one can see keeps the PSNR quiet constant while the PSNR of the traditional Structure Tensor drops down instantly.*

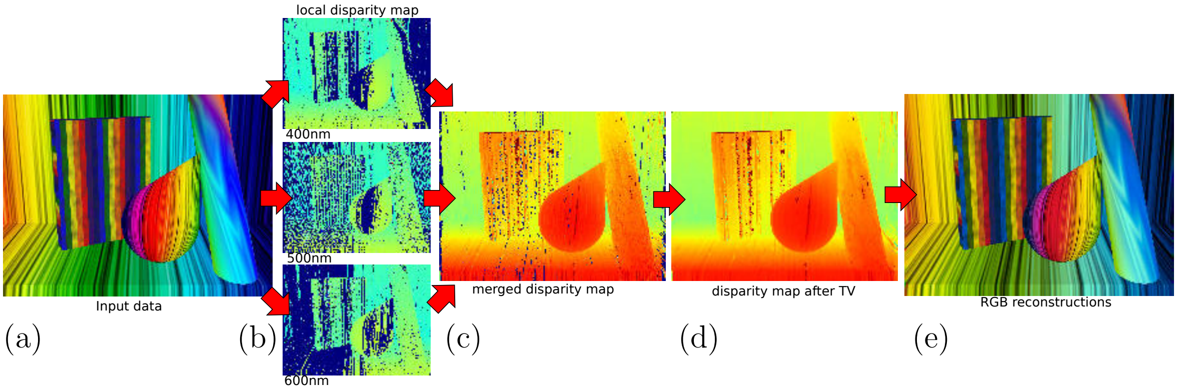


Figure 8.6: *(a) shows the reference image of the light field which become transformed into a color filtered light field. (b) are the local disparity estimations. (c) shows the merged disparity map out of the local disparity maps. (d) shows the final disparity map after total variation we applied. (e) shows the final disparity estimation.*

its breadth of color, this scene illustrates that the structure tensor is able to locally estimate the underlying orientation with respect to the visible wavelength. The local estimation results for 500 nm, 600 nm and 700 nm are depicted in figure 8.6 (b).

8.3.1 Color merge and total variation

It is necessary to merge the local disparity estimations into a reference view $r \in \Pi$ to obtain a dense disparity map. Thus the measured disparity needs to be transferred in the reference view. To select single disparity values, we use examine coherence order and replace estimations in the reference view with smaller coherence value. For the entire merging, we select each row $s \in \Pi \backslash r$ in the EPI and transfer the local disparity estimation of each pixel x to the addressed position y_s in the reference view r which is given by the equation

$$y_s = |x + s \cdot d_s(x)| \tag{8.2}$$

where the absolute value ensures only even pixel values are addressed in the reference view. That is important to consider since two or more local disparity estimates $d_s(x)$ at different positions x can address the same pixel in the reference view. This can happen when one object occludes another. The disparity merge can finally be described by

$$d_r(y_s) = d_s(x) \;\mid\; c_s(x) > c_r(y_s),\, d_s(x) > d_r(y_s) \tag{8.3}$$

where d_r and c_r are initialized with the local result of the reference view. Rounding of the applied pixel position introduces error in the resulting disparity map. To minimize this error we determine the actual shift position $\hat{x}$ in row s. The new position becomes

$$\hat{x} = y_s - s \cdot d_s(x). \tag{8.4}$$

When the disparity value at the new position $d_s(\hat{x})$ is within an epsilon environment ϵ with respect to the initial disparity value $d_s(x)$, it replaces the initial disparity value at location x. A reverse disparity value calculation checks whether the rounding has caused an object boundary to be crossed and, if so, rejects (in this case the selected disparity d_s). The final merged disparity map for a reference view of the rainbow textured scene is shown in figure 8.6 (c). Unfortunately, it still contains some small patches with undefined disparity values. For the further processing, it is necessary to have a dense disparity map. Thus we apply a second-order total variation, as introduced in the following subsection 8.3.2, on the merged disparity map. The proposed second order total variation approach minimizes the functional

$$F_{TV_2}(u) := \frac{1}{2}\|u - f\|_2^2 + \alpha TV(u) + \beta TV^2(u) \tag{8.5}$$

where $\alpha, \beta > 0$ denote regularization parameters. The result after the applied second order total variation is shown in figure 8.6 (d). The final hyper-spectral image can now be determined by addressing all color values along the orientations whose direction is given by the achieved disparity map.

	PSNR [dB]
Buddha	24.94
ConeHead	31.39
LivingRoom	25.80
Ruin	33.31

Table 8.2: *Shows the PSNR of the disparity estimation result, with respect to the ground truth for synthetically generated color-filtered light fields.*

8.3.2 Total Variation - A combined first and second Order TV

Processing light fields with the Structure Tensor approach results in sparse disparity maps, as shown in figure 8.8 (b). Due to its local processing, these disparity maps exhibit not only artifacts and high noise levels but also regions D without disparity information. To achieve fully populated disparity maps we introduce a combined first and second order total variation algorithm which applies denoising and inpainting to the computed disparity map $f : \Omega \rightarrow \mathbb{R}, \Omega \subset \mathbb{R}^2$.

For a suitable total variation in light-field imaging, we represent the regularization result by the function $u : \Omega \rightarrow \mathbb{R}$, where Ω defines a domain with Lipschitz boundary, as described by K. Bredies [16]. The total variation then becomes

$$TV^l(u) := \int_\Omega \frac{1}{2}|\nabla^l u|\, du \tag{8.6}$$

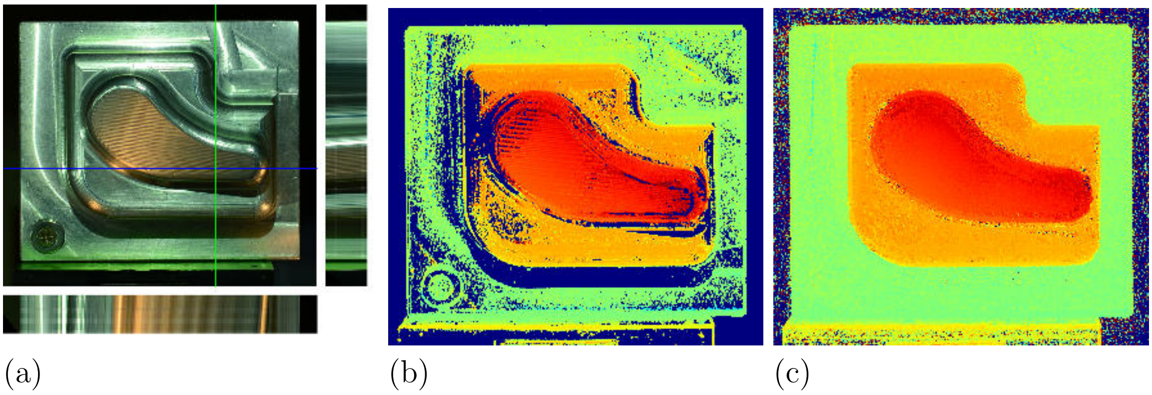


Figure 8.7: *(a) shows the center view of a captured metal test part. The two inserted lines address the row and column, where the EPIs is cut out. The related EPIs are shown on the right side and at the bottom. (b) represents the estimated disparity map using the traditional Structure Tensor as introduced in equation 2.6. (c) shows the improved estimation after applying our proposed new Structure Tensor 7.48.*

where $l = 1, 2$ dependent on the total variation order. With this, the minimization problem is given by

$$\min_{u} \int_{\Omega \setminus D} \frac{1}{2} \|u - f\|_2^2 \, dx + \Phi(u) \quad | \; \forall x \in \Omega \tag{8.7}$$

where $\Phi(u)$ is known as regularization which becomes a combined adaptive second-order total variation as introduced by F.Lenzen *et al.* [47, 48] and K. Papafitsoros [64]. The proposed combined second order total variation approach minimizes the functional

$$F_{TV2}(u) := (\lambda)\frac{1}{2}\|u - f\|_2^2 + (1 - \lambda)\alpha TV(u) + \beta TV^2(u) \tag{8.8}$$

where $\alpha, \beta > 0$ denote regularization parameters and $\lambda : \Omega \rightarrow [0, 1]$ a trade off between the controlling of the first order and the data fitting. This trade off λ becomes the coherence value c of the related disparity value, and represents the novelty of this approach. For a coherence value of zero $(c = 0)$, the data fitting is switched off and a pure first and second order inpainting is applied to the data. In contrast, when the coherence value becomes one $(c = 1)$ the data fitting is switched on, the first order is switched off, and only the second order total variation denoises the result.

To use the TV on input data that is not continuously differentiable, we introduce a dual variable $\xi \in \mathbb{R}^2$ which encodes the discontinuities. Here, the norm of the gradient of u becomes the scalar product defined by

$$|\nabla^l u| = \sup_{\xi \in \Psi} < \xi, \nabla^l u >, \tag{8.9}$$

where Ψ defines a disk, scaled by $\tau_l \in \mathbb{R}$ at each point $x \in \Omega$:

$$\Psi = \left\{ \xi \in C^\infty(\Omega, \mathbb{R}^{2l}), \forall x \in \Omega : \|\xi(x)\|_2 \leq \tau_l \right\} \tag{8.10}$$

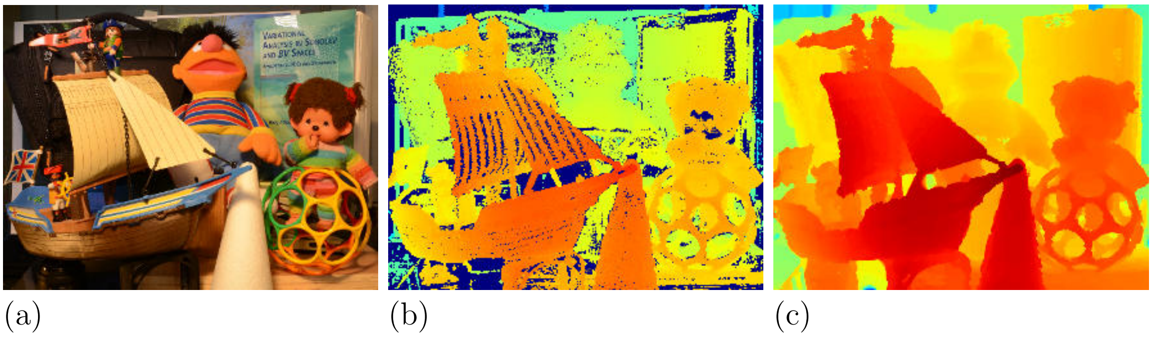


Figure 8.8: *(a) shows the center view input image of a captured light field. (b) shows the computed disparity map. Invalid disparity values are colored in blue. (c) shows the result after the second order total variation approach was applied. The related point cloud is shown in figure 8.10.*

and C^∞ defines functions that are differentiable over all degrees. Equation 8.9 obviously reaches its maximum when the dual variables point in the gradient direction. Thus it is possible to express the underlying total variation for the first and second order as

$$TV^l(u) := \sup_{\xi \in \Psi} \left\{ \int_\Omega < \xi, \nabla^l u > dx \right\} = \sup_{\xi \in \Psi} \left\{ \int_\Omega u \operatorname{div}^l \xi \, dx \right\} \tag{8.11}$$

8.3.3 Energy minimization using a primal dual algorithm

The proposed variational approach is based on the primal dual algorithm first introduced by T. Pock *et al.* [68, 69] which applies a steepest ascent onto the primal variable and a steepest descent on the dual variable. This approach gets extended to the second order as shown by F. Lenzen *et al.* [48]. The proposed algorithm becomes

$$\begin{align} \xi_l^{i+1} &= \Pi_{\Psi_l}\left(\xi_l^i + \alpha \nabla^l u_+^i\right), \tag{8.12}\\ u^{i+1} &= u^i + \sigma\,\lambda\,\left(u^i - f\right) \tag{8.13}\\ u^{i+1} &= u_{i+1} + \left(1-\lambda\right)\,\rho\,\operatorname{div}\xi_1 + \beta\,\operatorname{div}^2\xi_2 \tag{8.14}\\ u_+^{i+1} &= 2u^{i+1} - u^i, \tag{8.15}\end{align}$$

where an extrapolation step of u is added to determine ξ_l^{i+1}, which needs to be back-projected onto the scaled unit disk. The back-projection can be expressed by the formula

$$(\Pi_{\Psi_l}\xi_l)(x) = \frac{\xi_l(x)}{\max\{\tau_l, \|\xi_l(x)\|\}}. \tag{8.16}$$

The comparison between a first-order only regularization and an added second order regularization is shown in figure 8.9. To use only the first order total variation, the regularization variable β needs to be set to zero, which will deactivate the second order term. The advantage of the enabled second order total variation can be seen in the comparison of figure 8.9 and in the point cloud result 8.10.

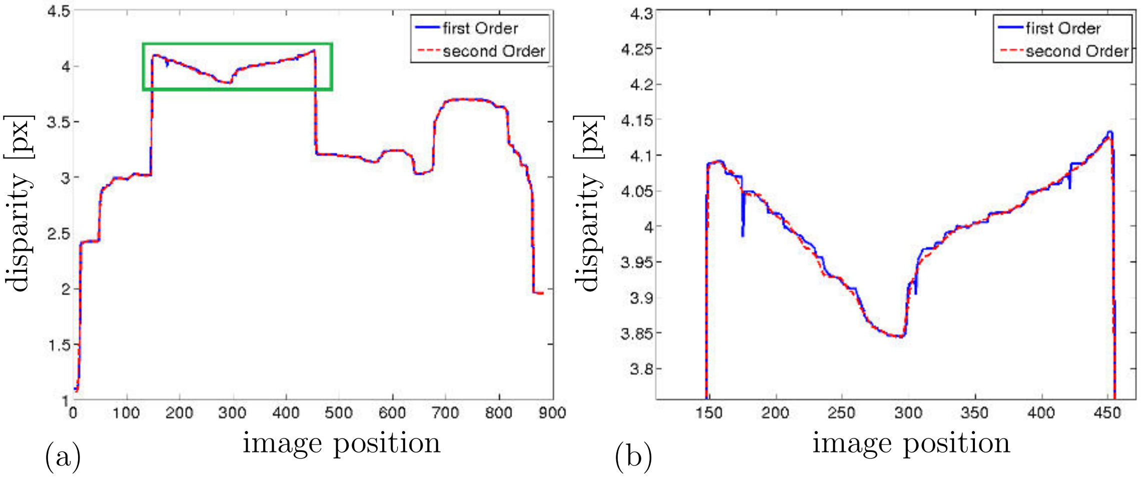


Figure 8.9: *(a) shows a scan line close to the image center of the light field as shown in figure 8.8(a). (b) shows the content of the left green box which shows the surface of the sail.*

8.3.4 Adaptive edge regularization

For this approach we consider α, β as dynamic variables, which vary depending on the image location. We define $\alpha(x), \beta(x) \; : \; \Omega \; \rightarrow \; \mathbb{R}_{+}, \; \forall \; x \; \in \; \Omega$ with respect to the edge map $E(x)$ of the input data f. The new $\alpha(x)$ and $\beta(x)$ become

$$\begin{aligned} \alpha(x) &= E(x) \cdot \big(\alpha(x) - \alpha_{\text{edge}}\big) + \big(1 - E(x)\big) \cdot \alpha_{\text{edge}} && (8.17) \\ \beta(x) &= E(x) \cdot \big(\beta(x) - \beta_{\text{edge}}\big) + \big(1 - E(x)\big) \cdot \beta_{\text{edge}} && (8.18) \end{aligned}$$

with $\alpha_{edge} \ll \alpha$ and $\beta_{edge} \ll \beta$ as reduced edge mobility, which avoids an over smoothing and a loss of contrast at edges. Unfortunately, this adaptivity of the edge mobility is only possible if the center view possess full edge information. In some heterogeneous light fields it is not possible to use this edge mobility adaptation. Thus α, β remain as constant factors.

8.3.5 Occlusion Detection

The reconstruction of the correct color information at object boundaries is not possible for linear color-filtered light field setups, as illustrated in figure 8.12. That implies, for achieving correct color information even for object boundaries, symmetric color filter setups are unavoidable. Nevertheless it is important to detect occluded areas, to apply an occlusion handling respectively.

Thus, the read-out direction along orientations become mirrored with respect to the central reference view, before hitting the occluded areas, see figure 8.11.

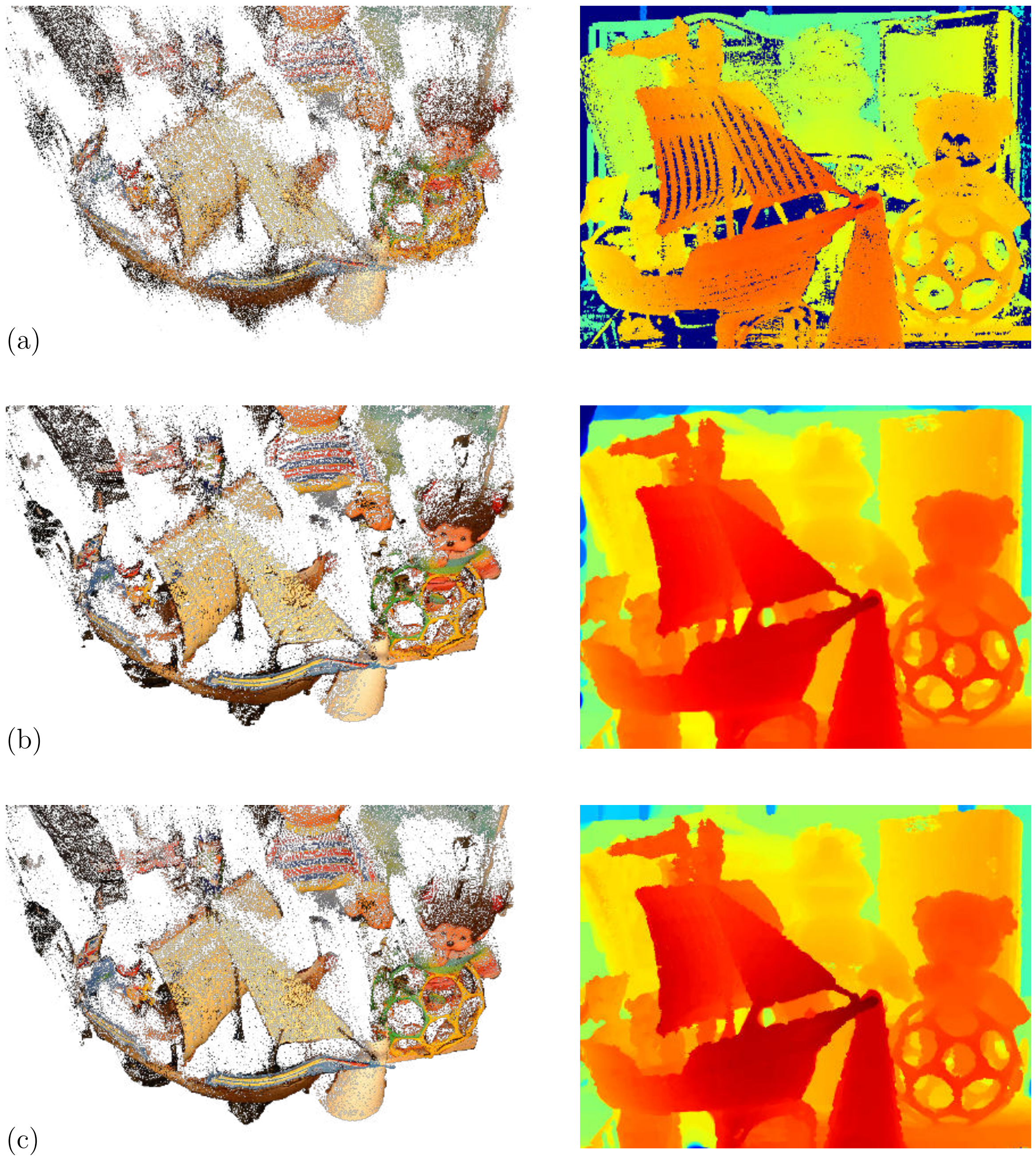


Figure 8.10: *Images on the left side show the 3D reconstruction based on the related disparity maps on the right. The result shown in (a) belongs to the Structure Tensor resulting disparity map. (b) shows the result after an applied first order total variation and (c) the result after the enabled second order total variation.*

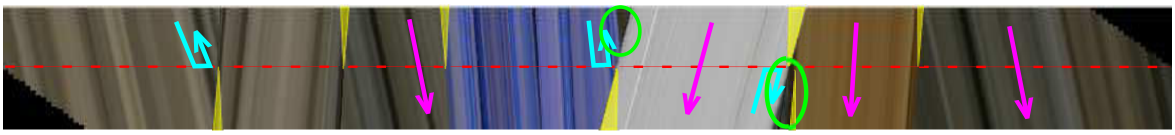


Figure 8.11: *Shown is the inversion of the color read-out direction for orientations hitting an occluded area – illustrated by blue arrows. Occluded areas are marked in yellow color and double occlusion areas are green encircled.*

To obtain the needed occlusion maps, the disparity maps related to the central three images of the light field are used, as shown in figure 8.13 (a), to detect occlusion areas by checking disparity consistency along the estimated orientation. While foreground objects keep a constant disparity value along the entire orientation, background content becomes occluded and thus a disparity change implies an occlusion. Thus it also becomes possible to determine the direction of the occlusion (right or left side of the object). The resulting occlusion map, as shown in figure 8.13 (b), addresses occluded regions as well as its direction. By using symmetric light-field setups and the described occlusion handling, it is possible to enhance the color of object boundaries significantly, as shown in figure 8.14.

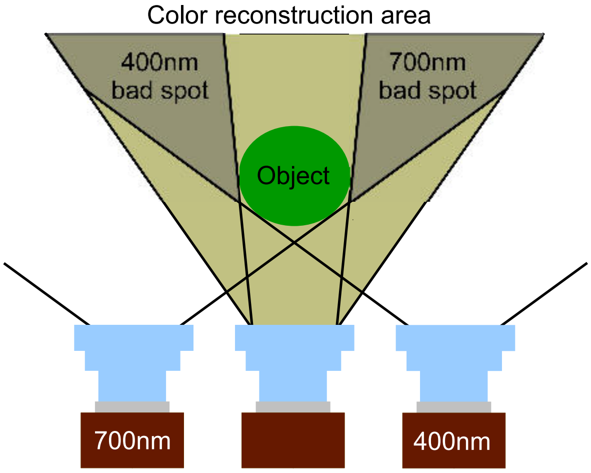


Figure 8.12: *This image shows the occlusion problem appearing in heterogeneous light-field setups. While foreground objects cover partially background seen by other cameras, a lack of spectral information appears in such areas as illustrated. Thus bandpass filter related bad spots are the consequence.*

Unfortunately also double occlusion is possible. Double occlusion appears, when background is occluded alternately by two foreground objects while moving the camera to acquire the light field. At such areas, it is not possible to determine the correct spectral information for the background. Thus a symmetric color-filtered light field in vertical direction can help, to resolve double occluded areas, as long as double occlusion not

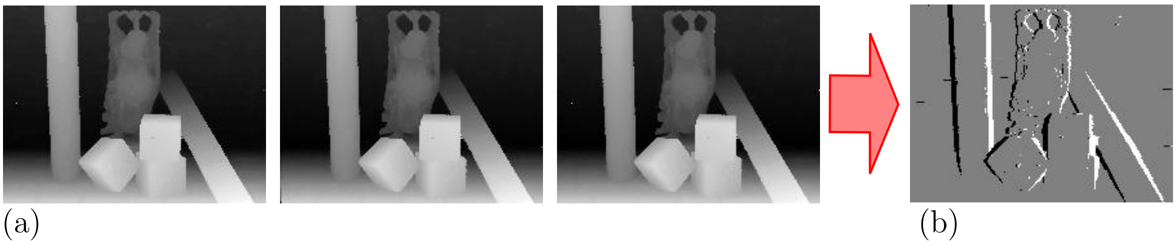


Figure 8.13: *(a) shows the three central estimated disparity maps, which are used to determine an occlusion map, which is shown in (b).*

also appears at the same position in vertical direction. Double occlusion correction is not analyzed in this chapter, because the main focus of this work are three-dimensional heterogeneous light fields.

8.3.6 sRGB reconstruction

For an RGB color reconstruction from spectral images we need to find a method to determine the (R, G, B) values from the pixel values S_i of the captured spectral images $i \in N$. For this, we use the approach of Tominaga [85]. He proposes that the CIE color space (X, Y, Z) can be determined by the pixel values S_i multiplied by a weighting function M

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \cdot \begin{pmatrix} S_1^{'} \\ \vdots \\ S_N^{'} \end{pmatrix}. \tag{8.19}$$

The weighting function to approximate the CIE color space can be determined by approximating the CIE color matching functions. Thus the camera quantum efficiency $QE_{\text{cam}}(\lambda)$ and the spectral sensitivity functions $BP_i(\lambda)$ of the image i related band pass filter as well as the spectral color distribution $LS(\lambda)$ of the light source need to be known. Then the CIE color matching functions $(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda))$ can be estimated by the formula

$$\begin{pmatrix} \bar{x}(\lambda) \\ \bar{y}(\lambda) \\ \bar{z}(\lambda) \end{pmatrix} = M \cdot \begin{pmatrix} QE_1^{'}(\lambda) \\ \vdots \\ QE_N^{'}(\lambda) \end{pmatrix} + e(\lambda) \tag{8.20}$$

with

$$QE_i^{'}(\lambda) = BP_i(\lambda) \cdot QE_{\text{cam}}(\lambda) \cdot LS(\lambda) \tag{8.21}$$

where e denotes noise. For the best possible approximation of the CIE color matching function we introduce the functional F_M which minimizes the area difference between the color matching function and the obtained approximation of M for the entire frequency domain Λ with $\lambda \in \Lambda$. The proposed functional becomes

$$F_M = \min_{\vec{k}} \int_{\Lambda} \begin{pmatrix} \bar{x}(\lambda) \\ \bar{y}(\lambda) \\ \bar{z}(\lambda) \end{pmatrix} - M_{3\text{x}N} \cdot QE_{\text{cam}}^{'}(\lambda)\, d\lambda. \tag{8.22}$$

After determining the weighting function M, we can transfer the spectral information to the CIE color space using equation 8.19. Next we convert the CIE color space as proposed by Tominaga [85] to sRGB color space [80]

$$\begin{pmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{pmatrix} = T \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{8.23}$$

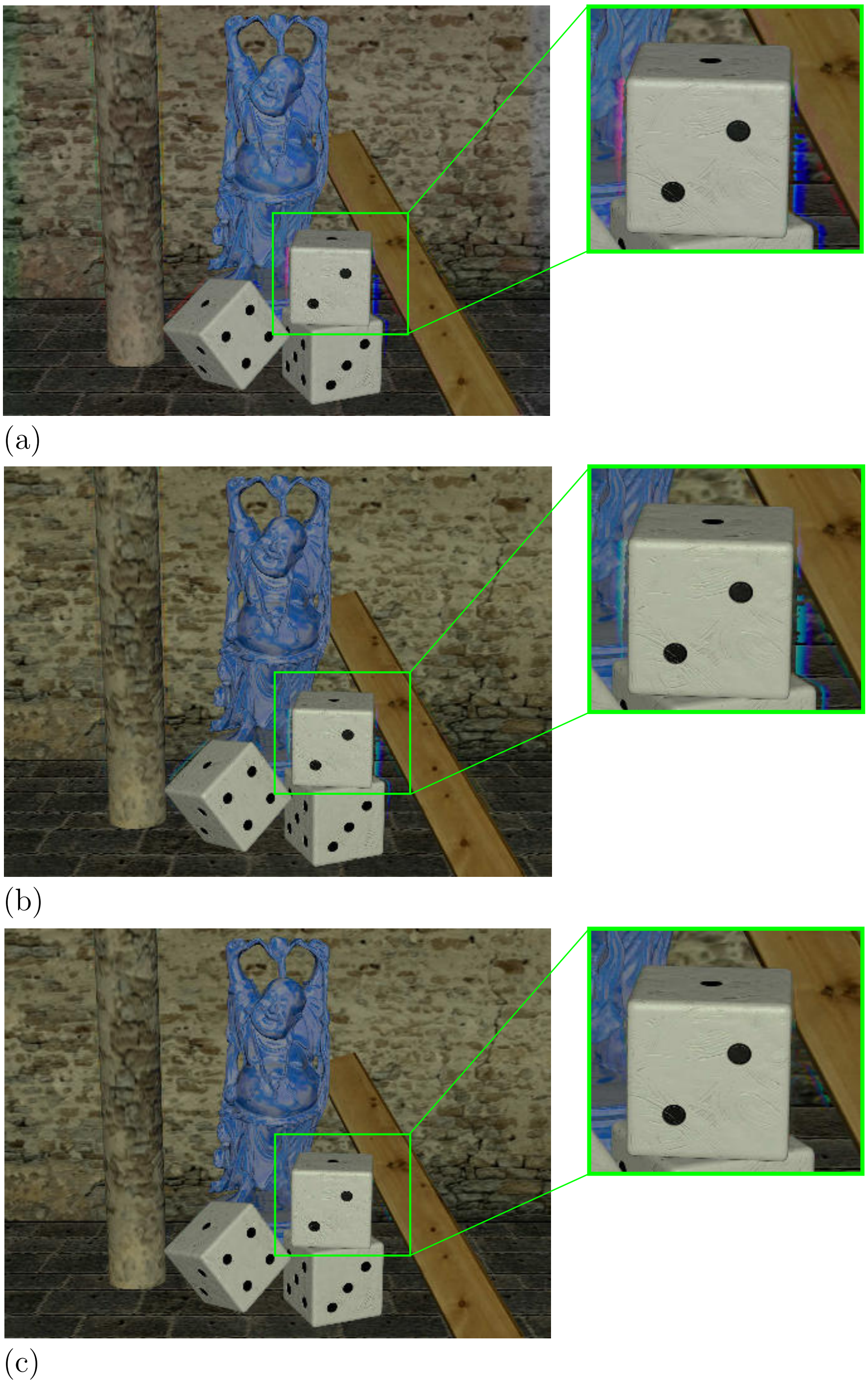


Figure 8.14: *(a) shows the RGB-reconstruction for a linear-color-filtered light field setup sampling the spectrum from* 400 *nm till* 700 *nm in* 10 *nm steps using color filters with a full width at half maximum of* 10 *nm. As on can see, blue content is missing on the right and red content is missing on the left. Thus the remaining color becomes more intense. (b) shows a RGB-reconstruction of a symmetric color-filtered light field setup, sampling the spectrum from* 400 *nm-*700 *nm-*400 *nm in* 20 *nm steps using the same filter. (c) shows the RGB-reconstruction after the application of the proposed occlusion handling with some remaining double occlusion artifacts.*

with

$$T = \begin{pmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} . \tag{8.24}$$

For the final visualization we also apply a gamma correction of $\gamma = 0.68$ to the linear RGB-space which transforms the linear values into sRGB.

8.4 Results

As result for homogeneous light fields, we show the comparison of the new structure tensors with respect to the traditional structure tensor implementation proposed in Wanner *et al.* [92]. Their implementation leads to the result shown in figure 8.7(a), while our proposed method shows a full coverage of the captured metal evaluation part as demonstrated in figure 8.7(b).

Next, we want to show the applicability of the proposed structure tensor to color-filtered light fields and the RGB reconstruction for one synthetic example and for four real captured light fields. To acquire heterogeneous light fields properly it is important to use linear-ordered color filters. They provide a more stable estimation for intensity gradients as shown for linear illumination gradient light fields in contrast to random distributed filters configurations. Due to that a constant quality in the orientation estimation of color-filtered light fields is guaranteed, as long as the gradient is not reaching the critical maximum. Furthermore, considering a color distribution of objects placed in the scene, as shown in figure 8.16 (c), it becomes important to select the filter respectively, to ensure color dependent orientations to appear. To estimate orientation for colors, only seen by one filter, is not possible. Thus we can derive two constraints

- For broad color spectra of target object, the used band pass filter can have a narrow bandwidth to obtain analyzable orientation.
- For narrow color spectra of target objects, the used band pass filter needs to be chosen that an orientation are visible in at least 5 neighboring images to guarantee a valid estimation.

As shown in figure 8.4, the EPI contains local orientation information while black transitions illustrate vanishing color content. Figure 8.15(a) shows the center view image of the initial synthetic homogeneous light field. This light field contains 31 images which was converted to a color-filtered light field as shown in figure 8.4. Each image is filtered with a band-pass filter having a full width at half maximum of 10 nm. The filters are uniformly distributed in 10 nm steps between 400 nm and 700 nm. The resulting RGB images of the synthetic light field is seen in figure 8.15(c).

The real color-filtered light fields are captured with a PCO-edge 5.5 camera, mounted on a high-precise translation stage. For the heterogeneous data we have a symmetric light-field configuration starting from 400 nm to 700 nm and back to 400 nm while each filter has a full width at half maximum of 10 nm. The filters employed are 400 nm, 450 nm, 500 nm, 515 nm, 532 nm, 550 nm, 560 nm, 589 nm, 600 nm, 650 nm, 700 nm and back down to 400 nm. The processing results of the acquired color-filtered light fields are shown in figure 8.16(c). Input data and additional images are provided in the additional material.

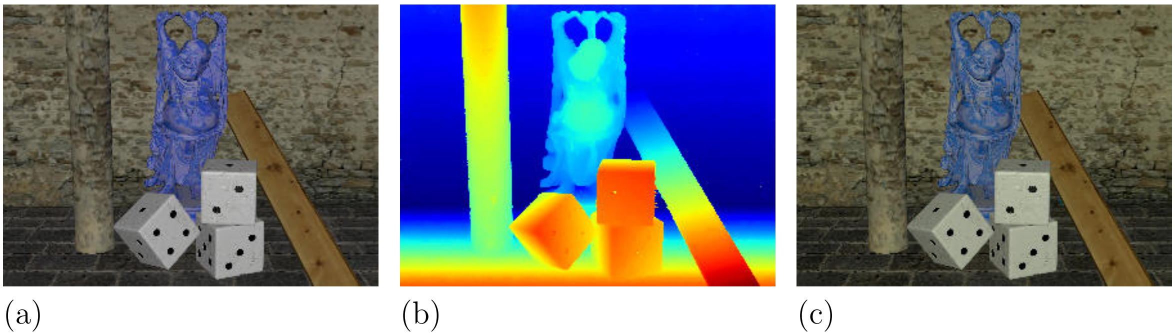


Figure 8.15: *(a) shows the original RGB center view of a cross shaped light field which consists of* 11 *images. (b) shows the final disparity result of the color filtered light field after the applied second order total variation. (c) shows the RGB reconstruction of the computed hyper-spectral image.*

8.5 Conclusion

The modified structure tensor approach introduced in the last chapter significantly reduces the error in the estimate of orientation in EPIs compared to the traditional structure tensor. We also demonstrate that the modified structure tensor has the advantage of high reliability in processing heterogeneous light fields. This makes it possible to better analyze acquired light fields since varying camera properties and flickering light sources inevitably cause illumination variations. It is nearly impossible to capture real homogeneous light fields having constant illumination along the orientation. But with the new designed structure tensor these small variations do not disturb the estimation; in fact they enhance the result. Additionally, metallic surfaces may be analyzed with greater density and accuracy, as shown in figure 8.7. Furthermore, we applied the structure tensor to color-filtered light fields. Here we have seen that the orientation computation is only possible in regions where the orientation remains visible in the EPI.

Thus we introduced a method to merge local estimations in a reference view to achieve a denser disparity map for that view. After applying a total variation approach to obtain a dense disparity map, we use the disparity information to address the hyper-spectral information along the orientation line. To prove this concept, we reconstructed the sRGB color space from the captured hyper-spectral information for both acquired and synthetic light fields. For this purpose we approximated the CIE color matching

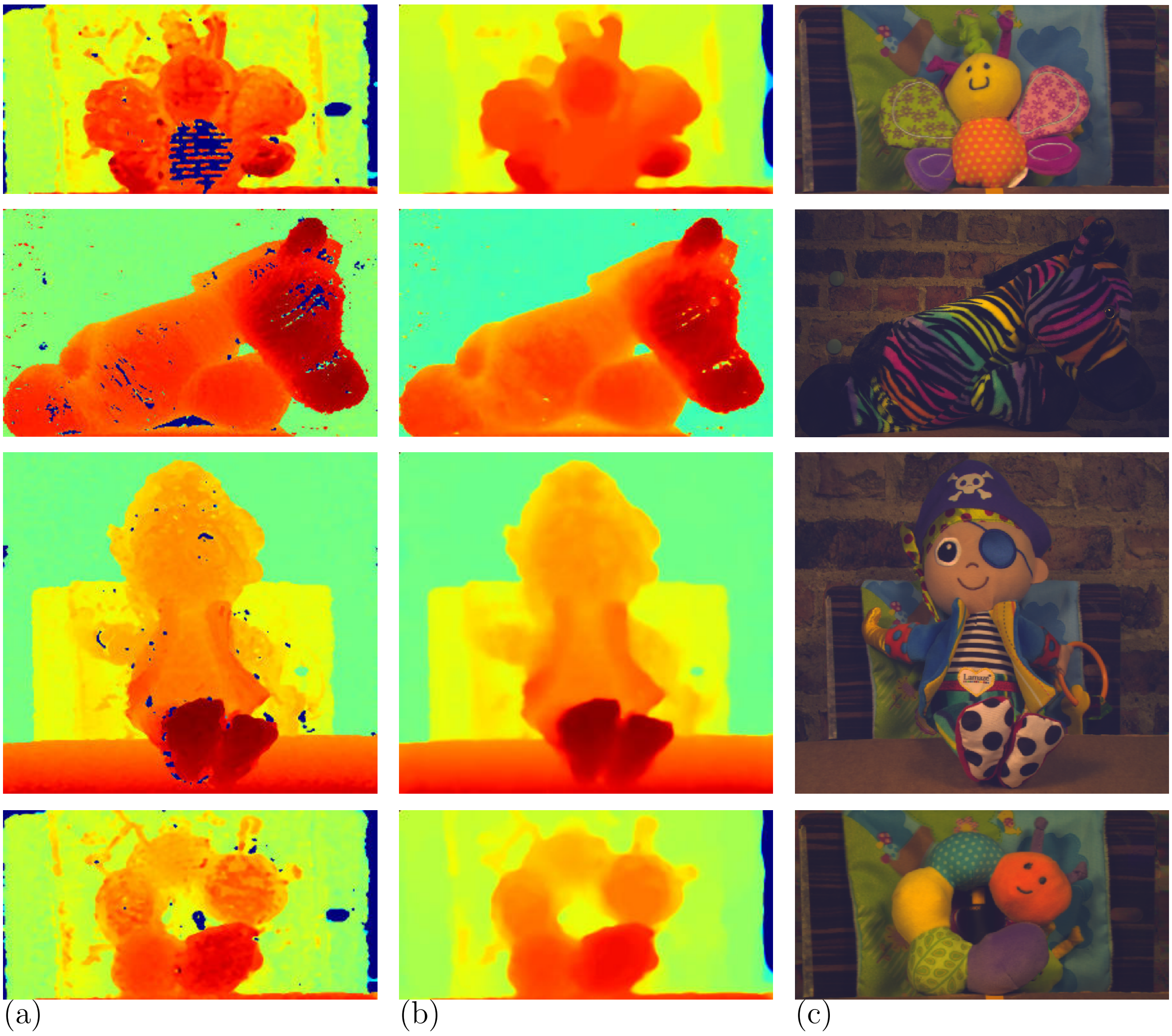


Figure 8.16: *(a) shows the merged disparity maps. (b) shows the result after the applied second order total variation. (c) shows the RGB reconstruction of the real captured light fields.*

function with the used band-pass filters and successfully approximated the CIE-color space. Finally the CIE-color space is mapped onto the sRGB color space. The resulting sRGB images are shown in figure 8.16 where one can see that, due to the height reliability of the estimated disparity, the color reconstruction matches the object boundaries perfectly. Color-filtered light fields hold a more densely sampled spectra as it is possible for a single camera. Thus an advanced analysis in the color domain is possible. Possible applications are in food surveillances, e.g to distinguish different cheeses by comparing their spectra, or to determine if food is still eatable or not. Another application is the study about defects in skin, where bacteria or chemical changes influencing the color spectra. It also opens the possibility to use the structure tensor for other types of heterogeneous light fields like for light fields with applied polarization filters to obtain BRDF information.

9 Spherical Light Fields

The following chapter is based on a cooperation with Bernd Krolla of the Augmented Vision department at DFKI Kaiserslautern and partially published in the Proceedings of the British Machine Vision Conference [44]. A full-view spherical camera exploits its extended field of view to map the complete environment onto a 2D image plane. Thus, with a single shot, it delivers more information about the surroundings than can be gathered with a normal perspective or plenoptic camera, as commonly used in light-field imaging. However, in contrast with a light-field camera, a spherical camera does not capture directional information about the incident light and, thus, a single shot from a spherical camera is not sufficient to reconstruct 3D scene geometry. In this chapter we introduce a method combining spherical imaging with the light-field approach. To obtain 3D information with a spherical camera, we capture several independent spherical images by applying a constant vertical offset between the camera positions and combine the images in a *Spherical Light Field* (SLF). We can then compute disparity maps by Structure Tensor orientation analysis on epipolar-plane images which, in this context, are 2D cuts through the spherical light field with constant azimuth angle. This method competes with the acquisition range of laser scanners and allows for a fast and extensive recording of a given scene.

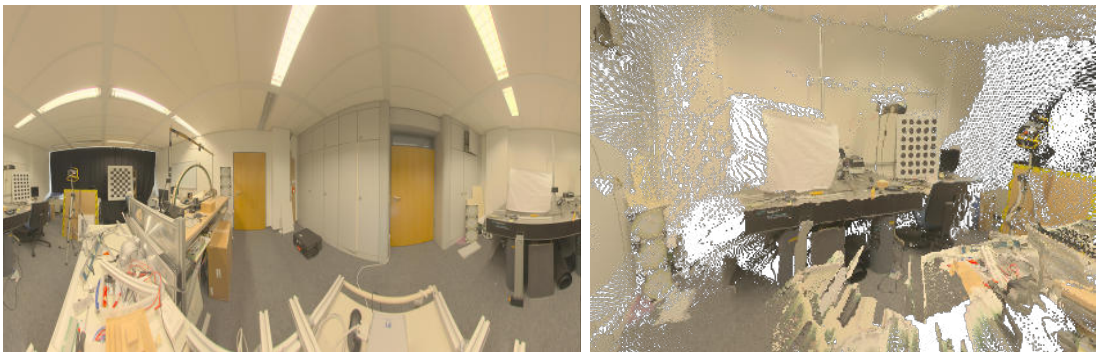


Figure 9.1: *The left image shows the captured* $360\circ$ *spherical center view image of the HCI optics laboratory. The right image shows a part of the 3D reconstruction.*

9.1 Introduction

Since projects such as Microsoft *Street Side* [61] or Google *Street View* [4] have provided numerous spherical images to online users, spherical imaging has experienced increasing attention in the recent past. To acquire such spherical images, a wide variety of hardware devices is available, delivering results of varying quality and accuracy. The devices separate into professional solutions [93, 79, 53, 71] and consumer

oriented camera devices such as [83, 76]. Torii *et al.*[86] provide a fundamental and elegant definition of spherical cameras, subsuming central dioptric and catadioptric cameras under the assumption of known camera parameters into this camera model Spherical cameras are able to handle interesting application scenarios not realizable with standard perspective cameras. Pagani *et al.* [5] researched *Structure from Motion* approaches using full spherical cameras, whereas the work of Aly and Bouguet [3] is more focused on the calibration of unordered sets of spherical images. Gutierrez *et al.* [31] showed that visual SLAM can be performed without loss of image features caused by camera rotation when using spherical instead of perspective cameras. Furthermore, as application oriented approaches, 3D reconstruction using multi-spherical stereo has been employed to reconstruct the 3D environment of a static scene [32].
The combination of omnidirectional images with *High Dynamic Range* (HDR) imaging as introduced in [59, 75] expands image processing possibilities such as noise reduction, shadow handling or avoidance of under- and over-exposed image regions. In addition to spherical image acquisition, light-field imaging has gained more and more attention. In this chapter we introduce the acquisition of spherical light fields and the determination of the 3D geometry by using the Structure Tensor orientation estimation.

Related work.
The interface between light-field imaging and omnidirectional camera systems has been addressed by recent research whereas, to the best of our knowledge, full spherical images have not been considered. Birklbauer and Bimber created panorama light fields by stitching multiple perspective light fields taken by a rotating light field camera [12]. Due to the devices employed [58, 74] the vertical field of view (FOV) remained limited. Taguchi *et al.*[82] used an array of spherical mirrors to model catadioptric cameras for wide-angle light-field rendering. While providing dense depth estimation and refocusing capabilities for the captured scenes, the setup entailed decreasing tangential resolution close to the mirror borders, limiting the FOV to 150° × 150°. Unger *et al.* [88] employed a capture configuration similar to Taguchi, as well as a fisheye-camera translated on a plane to capture hemispherical HDR images of a scene. Aiming at the rendering of artificial objects in the captured environment, the total acquisition time took up to 12 hours for a single scene. This restricts the application scenario to indoor static environments, since constant illumination conditions during the acquisition are crucial for the subsequent light-field processing. An alternative approach to obtain full spherical depth and disparity maps for a surrounding scene is to use laser scanners, which measure the depth in a bounded range around the device. Even though these devices achieve highly accurate reconstructions, they are in general high-priced devices and the resulting scans are commonly provided without texture information of the scene.

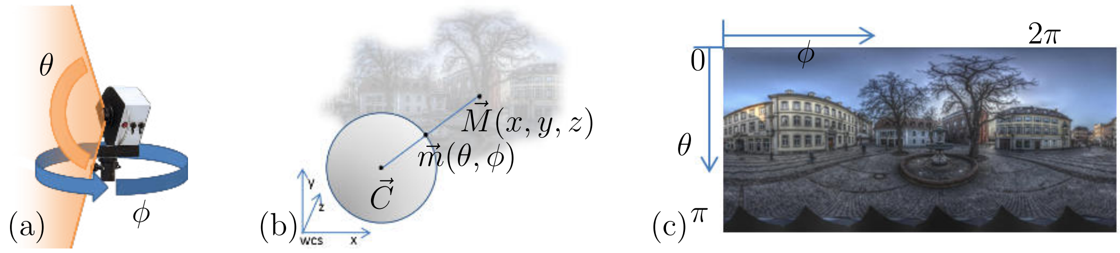


Figure 9.2: *(a) Spherical image acquisition using a rotating tripod mounted camera equipped with a fish eye lens. (b) The image results from the back projection of 3D points* $M(x, y, z)$ *to their corresponding image points* $m(\theta, \phi)$ *assuming* C *to be the cameras center of projection. (c) In the current work, the resulting image is a High Dynamic Range (HDR) image with a resolution of* 14000×7000 *pixel and is parametrized using spherical coordinates* $\phi[0, 2\pi)$ *and* $\theta[0, \pi]$.

Contributions

In this chapter, we combine spherical imaging with light-field analysis and introduce the concept of Spherical Light Field (SLF) recording. To capture an SLF, we obtain several spherical images from different elevations. This results in a 3D data structure parametrized by the two angular directions and the height of the capturing device. We show that by computing 2D cuts through this structure with fixed azimuth angle, we obtain the analogue of an epipolar-plane image, where we can efficiently perform depth reconstruction via orientation analysis. Furthermore, this makes it possible to directly adapt light field analysis techniques which rely on epipolar-plane image analysis to the scenario of omnidirectional scene acquisition. Compared to conventional light-field cameras, we acquire significantly more information about the surrounding scene. In particular, we also capture the scene in high dynamic range. In the context of this work, we can thus benefit especially from improved texture representation as well as improved illumination estimation to increase the performance of subsequent analysis of the SLF. We demonstrate that SLF offers the possibility of very short acquisition times using small sets of 9-13 high-resolution spherical images for disparity estimation.

9.1.1 Spherical image acquisition

Our proposed approach for SLF acquisition relies on the utilization of spherical cameras as shown in figure 9.2 (a). A convenient description of this camera type is provided by Torii *et al.* [86], who consider a spherical camera to consist of a camera center C with a surrounding unit sphere acting as projection surface. This definition implies that no intrinsic parameters such as focal length or distortion values known from perspective imaging need to be considered. According to the collinearity constraint, any 3D point M of the camera's environment is mapped via the camera center C to its corresponding image point m, see figure 9.2 (b). Any position within the resulting spherical image is

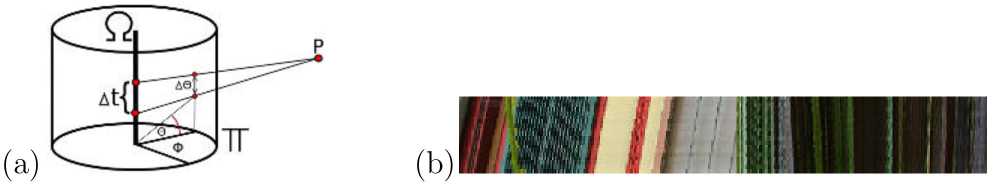


Figure 9.3: (a) Parametrisation of the Spherical Light Field. (b) Example of an Epipolar Plane Image (EPI) assembled from 15 images.

uniquely defined by the image coordinates $\phi \in [0, 2\pi)$ and $\theta \in [0, \pi)$. By applying the Mercator projection [10], the spherical image is mapped conformally onto an image on a cylindrical surface Π, see figure 9.2 (c). Note that this kind of data representation implies a significant distortion of image content close to the image poles ($\theta \to 0$ and $\theta \to \pi$). However, it assures that any content of the scene is shifted along the latitude-axis of the image, with respect to vertical displacement of the camera position. Therefore, this representation is suitable for epipolar-plane image (EPI) reconstruction, as outlined in the following section.

9.1.2 Spherical Light Fields

To describe an SLF, we define a new parametrization for the camera domain and the surrounding spherical 2D mapped image, see figure 9.3 (a). We take the cylindrical surface Π and denote the center line with Ω. The cylindrical surface Π is parametrized by image coordinates $(\phi, \theta) \in \Pi$. The line Ω contains the focal points $t \in \Omega$ of all possible camera positions in the vertical direction. A Spherical Light Field can now be described by a function

$$L : \Omega \times \Pi \to \mathbb{R} \qquad (t, \phi, \theta) \mapsto L(t, \phi, \theta), \tag{9.1}$$

where $L(t, \phi, \theta)$ defines the intensity of the incident light ray in the image plane (ϕ, θ) passing through the focal point t. To estimate the disparity, we address a 2D slice Σ_{ϕ^*} of the SLF by setting ϕ to a fixed value ϕ^*. The restriction of the light field to such a slice is called an epipolar-plane image (EPI), and formally defined as

$$S_{\phi^*} : \Sigma_{\phi^*} \to \mathbb{R} \tag{9.2}$$

$$(\theta, t) \mapsto S_{\phi^*}(\theta, t) := L(t, \phi^*, \theta). \tag{9.3}$$

An example is shown in figure 9.3 (b). Assuming a Lambertian scene, the EPI yields information about the disparity of a scene point in the form of oriented lines. Each line corresponds to the projection of a scene point, and its slope is directly related to parallax, so is in a one-to-one correspondence with the distance of this point from the camera center. To compute the disparity on the EPI, we can thus perform an orientation analysis on the given EPI S_{ϕ^*}, using the structure tensor

$$J = \tau * \begin{pmatrix} (S_\theta)^2 & S_\theta \, S_t \\ S_t \, S_\theta & (S_t)^2 \end{pmatrix} =: \begin{pmatrix} J_{\theta\theta} & J_{\theta t} \\ J_{\theta t} & J_{tt} \end{pmatrix} \tag{9.4}$$

with the abbreviations

$$S_t := \sigma * \frac{\partial S}{\partial t}, \quad S_\theta := \sigma * \frac{\partial S}{\partial \theta}. \tag{9.5}$$

The orientation angle and thus the disparity map d for the EPI S_{ϕ^*} can be computed directly from the components of the structure tensor via

$$d = \tan\left(\frac{1}{2}\arctan\left(\frac{J_{tt} - J_{\theta\theta}}{J_{\theta t}}\right)\right). \tag{9.6}$$

As a reliability measure of the estimated disparity, one can employ the coherence κ defined by

$$\kappa = \sqrt{\frac{\left(J_{tt} - J_{\theta\theta}\right)^2 + 4J_{\theta t}}{\left(J_{tt} + J_{\theta\theta}\right)^2}}\ . \tag{9.7}$$

The full set of disparity and coherence maps is computed by iterating over all EPIs from the SLF and storing the computed disparity and coherence values at the corresponding azimuthal slice.

9.2 Results

To acquire spherical images in real environments, we use the omnidirectional dioptric *Civetta* camera manufactured by Weiss AG [93], see figure 9.2 (a). This camera is equipped with a fish eye lens and provides omnidirectional 360° × 180° HDR images by stitching multiple perspective images together. Since the camera software handles distortion and overlaps of the input images, the resulting spherical HDR images comply with the spherical camera model introduced previously. By applying the Mercator projection, they are mapped to a plane and stored as EXR-files [51] with a resolution of 14000 × 7000 pixels, see figure 9.2 (c). The file size of up to 320MB results from a combination of high resolution and a 24bit HDR color representation. For the capturing process, we need to consider that the camera requires a static scene to provide accurate results. Since the HDR characteristic of the images is obtained by capturing multiple images with varying exposure time from the same position, moving objects cause artifacts in the resulting image. To acquire the actual SLF, camera positions of increasing height were engaged by varying the tripod's elevation by a fixed amount on the order of several millimeters. In addition to the desired pixel offset along the latitude coordinate θ, minor offsets along the longitude coordinate ϕ also occurred due to manual adjustment of the tripod height. Thus, to assure optimal data quality for a reliable EPI generation, a realignment of the images was performed as a first post-processing step after image capture. To perform the image realignment along the ϕ coordinate, standard computer vision methods were applied by extracting and matching SIFT-features [54] from the different images. To improve the robustness of the realignment, feature extraction was limited to a strip along the image equator (+/- 60° latitude) by masking distorted image regions close to the image poles. After rejecting match outliers, the average offset between the images could be retrieved up to subpixel precision, and this was used to align the captured set of spherical images. For 3D reconstruction we applied the structure tensor as introduced in chapter 2 followed by second order total variation as introduced in chapter 8.3.2. The result in figure 9.4 and figure 9.5 shows the point cloud of a courtyard and the HCI optics lab in larger scale.

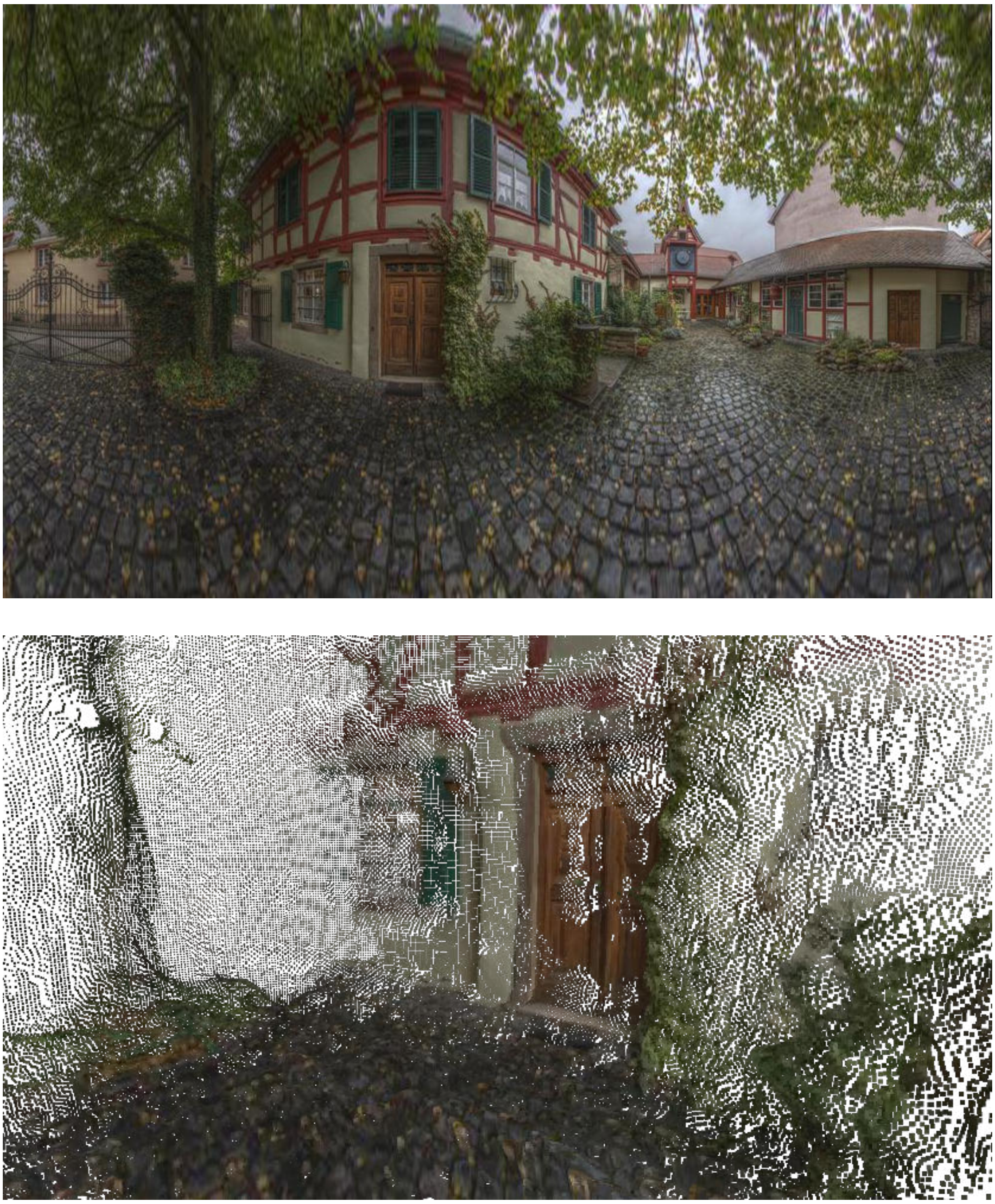


Figure 9.4: *The top image shows the captured* 360° *spherical center view image of a court yard. The bottom image shows a part of the 3D reconstruction.*

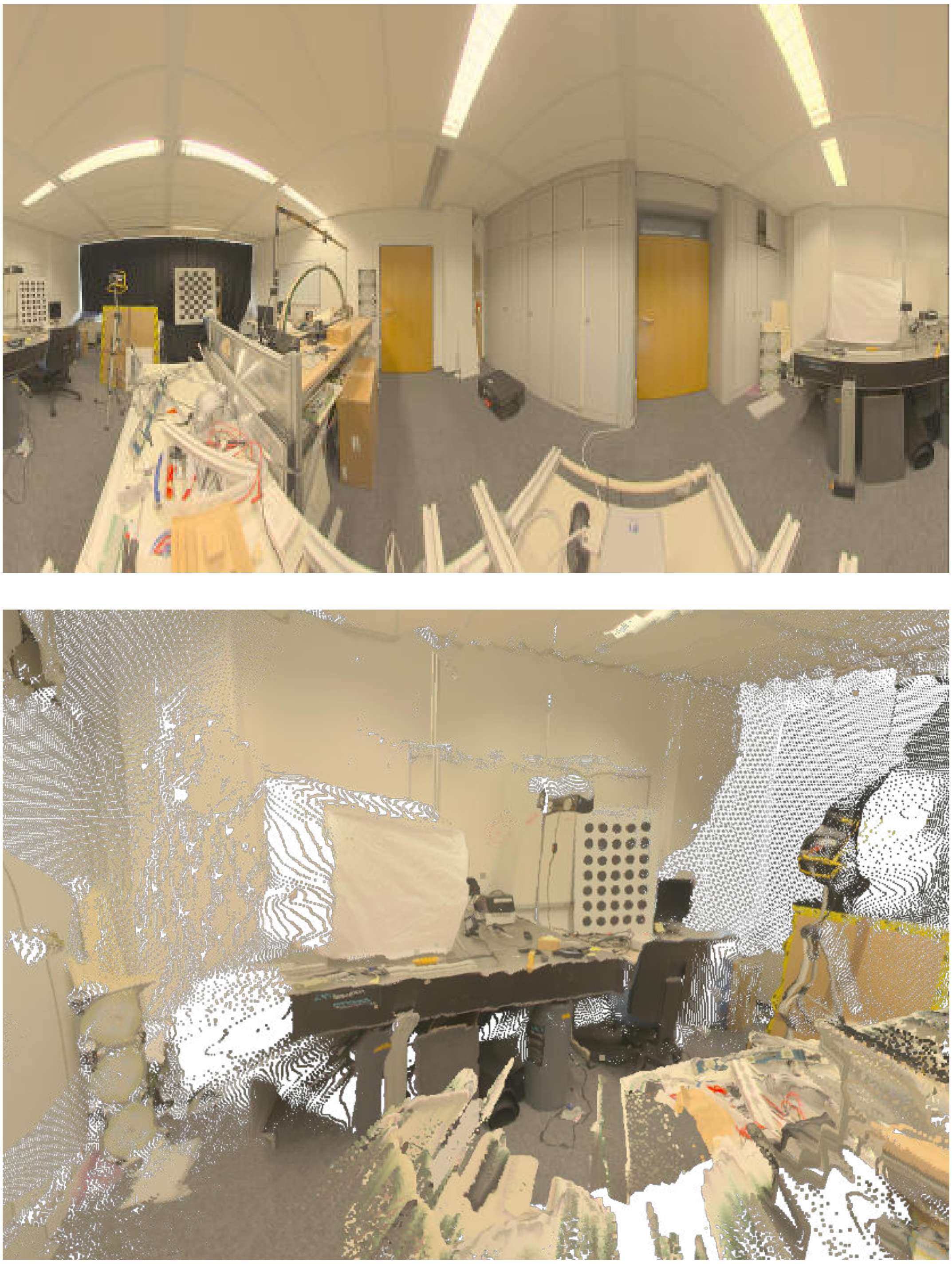


Figure 9.5: *The top image shows the captured* 360° *spherical center view image of the HCI optics laboratory. The bottom image shows a part of the 3D reconstruction.*

9.3 Conclusion

We capture spherical light fields in real-world environments using full spherical cameras. The mapping of the resulting spherical images to a conformal representation on a 2D plane allows to easily construct epipolar-plane images, on which it is possible to apply orientation analysis for fast and accurate disparity estimation. The resulting full view spherical disparity maps can then be employed for a 3D scene reconstruction of the cameras surroundings. Furthermore, combining spherical and HDR imaging approaches for the capturing of real scenes can greatly simplify the task of disparity estimation due to e.g. improved contrast. The concept of spherical light fields presents a promising avenue to expand the applications for light-field processing, in particular towards those which require a detailed and complete map of the surroundings.

10 Conclusion

In this thesis we introduced various types of Structure Tensor implementations and analyzed their achievable precision. This starts from the normal 2D Structure Tensor applicable on each EPI independently followed by the 2.5D and 3D Structure Tensor, up to the 4D Structure Tensor applicable only over the entire light field. As we have shown, the 2.5D Structure Tensor obtains the best precision results.

Additionally, we evaluated the impact of using an asymmetric Gaussian filter in the Structure Tensor computation. Here we determined that truncated asymmetric Gaussian filter obtains similar results to symmetric Gaussian filter but leads to even better definition at transitions between objects. Aside from this, we analyzed the usability of the coherence measure in global shifting and integration of the vertical and horizontal direction estimates in cross-shaped light-field configurations. Unfortunately, we found that while it operates perfectly for the global shifting, coherence is a poor indicator for choosing among vertical and horizontal light field estimates.

Further, to achieve high quality light fields, we found that having the correct configuration is as important as having the correct equipment. Thus we focused on the light-field setup itself, and how to determine setup configurations to optimally resolve the needed depth range in obtaining high quality depth maps.

In the second half of the thesis, singular value decomposition and canonical correlation analyses were introduced and used to estimate orientation in EPIs. It was shown that these approaches can be converted from one to the other, as well as being transferred to the Structure Tensor representation. With this conversion tool it was possible to derive an improved metric for single orientation estimation from the second-order Structure Tensor, which is also used to separate reflective and transparent orientation layers [91]. The derived improved Structure Tensor achieves more than higher precisions, it is also able to process heterogeneous light fields such as the introduced illumination-gradient light fields and the color-filtered light fields. For the color-filtered light field we additionally showed how to achieve high reliable dense disparity maps which can be used to determine hyper-spectral images and RGB-reconstructions of real captured scenes. Further investigations are made in cooperation with B. Krolla of the DFKI in Kaiserslautern about spherical light fields and its correct mapping to achieve full 360° disparity maps and from this full $3D$ reconstructions.

In the appendix we introduce temporal light fields which exploit the temporal direction to apply a foreground object removal.

11 Outlook

The improved structure tensor introduced in this thesis does not only achieve better results in analyzing homogeneous light fields, it also extends the area of application to heterogeneous light fields. Unfortunately, the boundary value problem remains, even for the improved structure tensor. That means boundaries between two neighboring objects are bad estimated, as shown in the analysis of asymmetric Gaussian kernel. There we achieved an improvement but no optimal solution. This problem could be solved by combining the structure tensor with the zero crossing method. Zero crossings determine transitions between objects better and could enhance the introduced total variation, that correct transitions become more defined.

Aside this, new application methods are possible using heterogeneous light fields, such as food inspection or material classification. Considering other heterogeneous light fields like polarized light fields also the analysis of BRDF information becomes feasible. In the end, due to the fast processing of the structure tensor and the possibility to parallelize the EPI processing, also movie analysis becomes an important field of research in near future.

Appendices

A Temporal Light Fields

The previously introduced light fields are all captured in horizontal or vertical spatial directions to achieve epipolar-plane images (EPIs) which encode scene depth information. In this chapter we introduce temporal light fields by acquiring temporal image sequences using a single camera at a fixed position. The acquisition of the sequences can be triggered in equidistant or random times, depending on the intended use. In its properties, a temporal light field is closely related to an optical flow approach. Using equidistant trigger times makes temporal light fields suitable to determine object velocities which are encoded as slopes in the horizontal and vertical EPIs.

Additionally, temporal light fields are suitable for applying foreground/background segmentation, and thus can support foreground object removal.

Considering moving foreground objects against a static background makes it possible to segment foreground and background related regions by analyzing slopes in the vertical and horizontal EPIs. Thus a reconstruction of the full static background without interfering foreground objects is possible. In the following we want to detail how this foreground object removal works and to demonstrate its applicability to some exemplar data sets. For the analysis, only horizontal EPIs are considered since this suffices for a demonstration of foreground removal.

A.1 Foreground object removal

For the proposed foreground object removal algorithm, images can be captured at randomly times. Each sequence can consist of arbitrary mount of images while a minimal required number of 10 images is mandatory for this method. Otherwise the needed filter operations are not properly applicable. A subset of such a sequence is shown

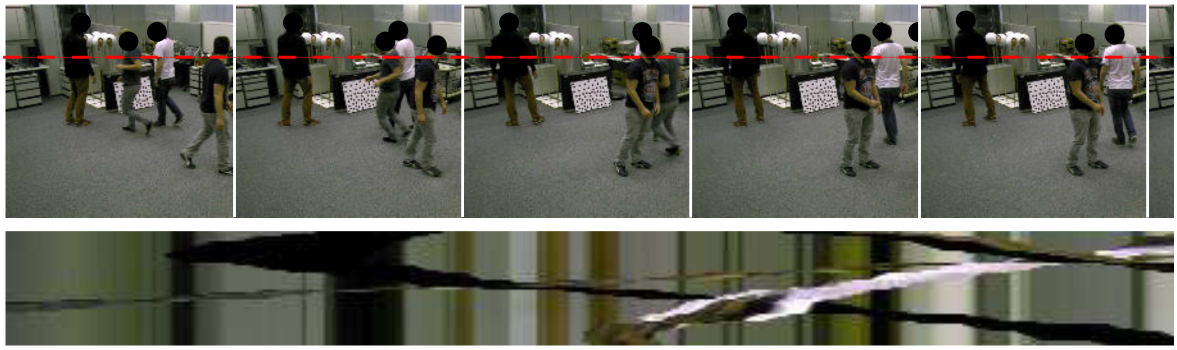


__Figure A.1:__ A subset of the captured temporal light field. The full light field contains 47 *images while only* 9 *are displayed. At the red horizontal line an EPI is sliced out and plotted below. The larger version of the input subset is shown in figure A.3.*

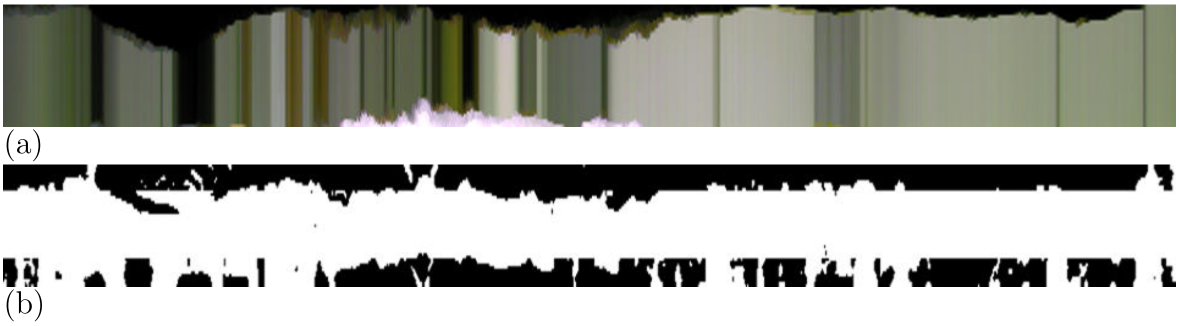


Figure A.2: *(a) This image shows the gray scaled value related color sorted EPI. (b) This image shows the mask which addresses background related content. In this image, white addresses background-related content and black foreground-related content.*

in figure A.1 which contains in total 47 images. The displayed EPI shows that everything related to the background has vertical orientation, while everything else has other slopes. Thus, to extract the background, it is only necessary to determine regions having vertical orientations.

With the assumption that interfering objects can appear randomly at each position in the EPI makes it likely that no vertical orientation can be found in some EPI regions. To avoid this we sort the colors in each EPI row with respect to its gray scaled converted value. That guarantees that pixel with similar color content are close to each other as shown in figure A.2 (a).

An additional property which can be seen in the sorted EPI is that interfering objects are either moved to the upper, to the lower or to both boundaries of the image. Thus the correct background-related content needs to be centrally located in temporal EPI direction. The analysis of this central EPI region is evaluated in two ways. Firstly, to address foreground-related image content we apply the Structure Tensor. By exploiting coherence c and disparity d estimates we can address background-related content by checking for vertical orientations with a high reliability. Thus we use a coherence threshold ξ and a disparity threshold χ, to address background-related content which has $c > \xi$ and $|d| < \chi$. Anything else is classified as foreground content. Unfortunately, texture-less regions will also be defined as foreground. To avoid that, we secondly focus exclusively on the vertical gradient η. Regions with vertical gradients below a threshold θ relate to the background, and everything above this threshold θ relate to the foreground. By merging both constraints we obtain binary masks, which address foreground and background regions as seen in figure A.2 (b). The shown masks relate to the sorted EPI as shown in figure A.2. After addressing all background-related pixels we can postulate that more than 50% of the addressed pixels relate to the background. Thus it is possible to determine for each EPI column a median value, and average all pixels inside an ϵ environment around this median value. This final averaging reduces image noise and also represents the extracted static background as demonstrated in figure A.3.

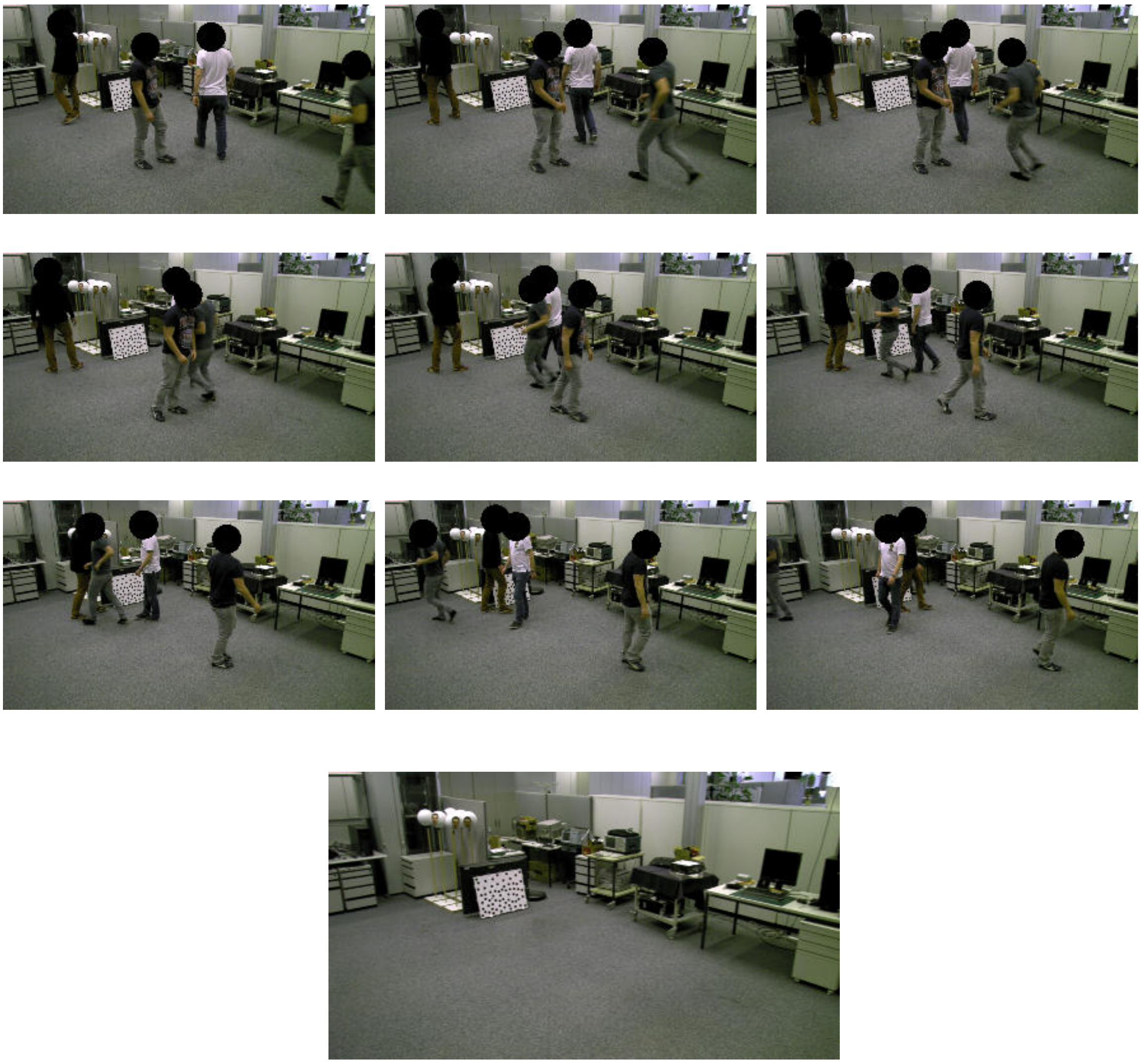


Figure A.3: *Shown is a subset of* 9 *images. The entire temporal light field consists of* 47 *images in total. The bottom image represents the final background reconstruction where all the foreground content is removed.*

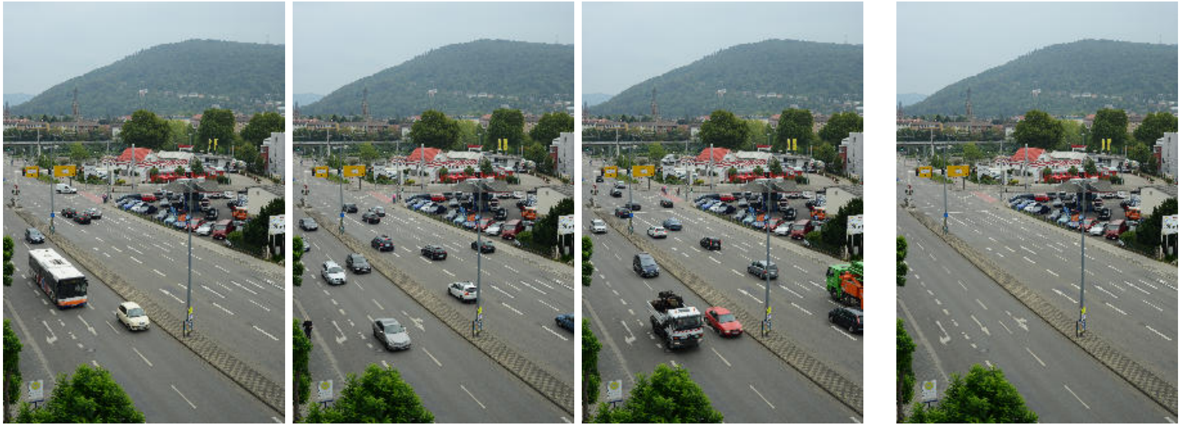


Figure A.4: *This temporal light field consists of 42 images in total. Shown is a subset of 3 images. The right-most image represents the background reconstruction.*

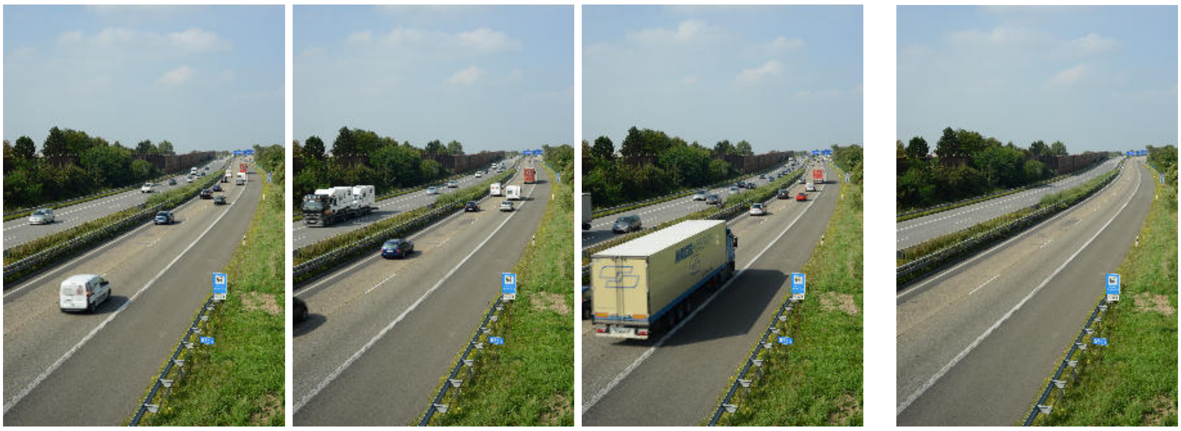


Figure A.5: *The temporal light field consists of 53 images in total. Here only 3 images are selected and shown. The right-most image shows the final background reconstruction using all images.*

B Orientation estimation

Disparity evaluation for the 2D Structure Tensor (Scharr filter)

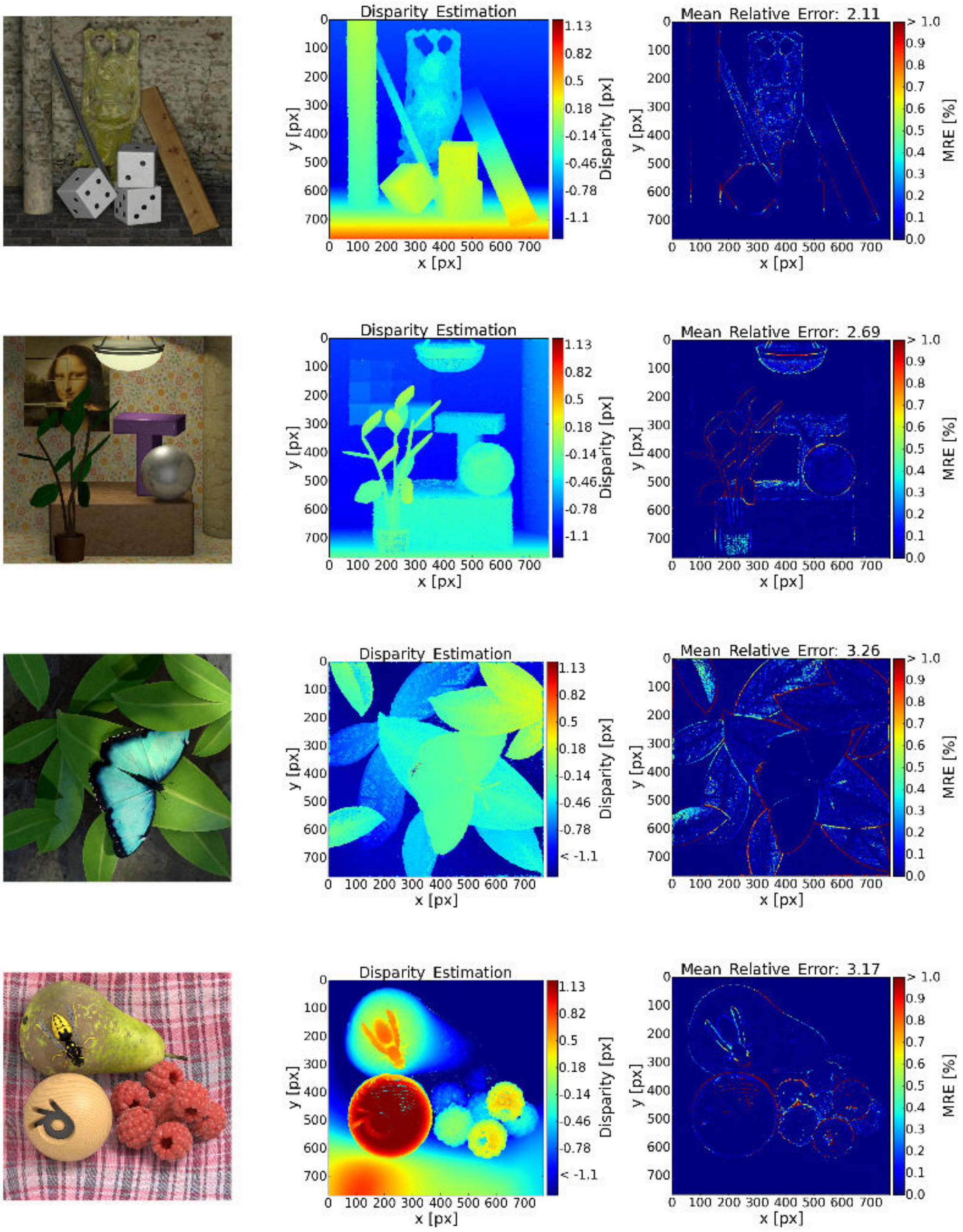


Figure B.1: *The first column shows the center view image. The second column shows the related disparity map. The last row visualizes the mean relative error distribution in the image. The chosen inner Gaussian smoothing is* $\sigma_{[5\times5]} = 0.5$ *and the outer Gaussian smoothing is* $\tau_{[9\times9]} = 1.3$

Disparity evaluation for the 2.5D Structure Tensor (Scharr filter)

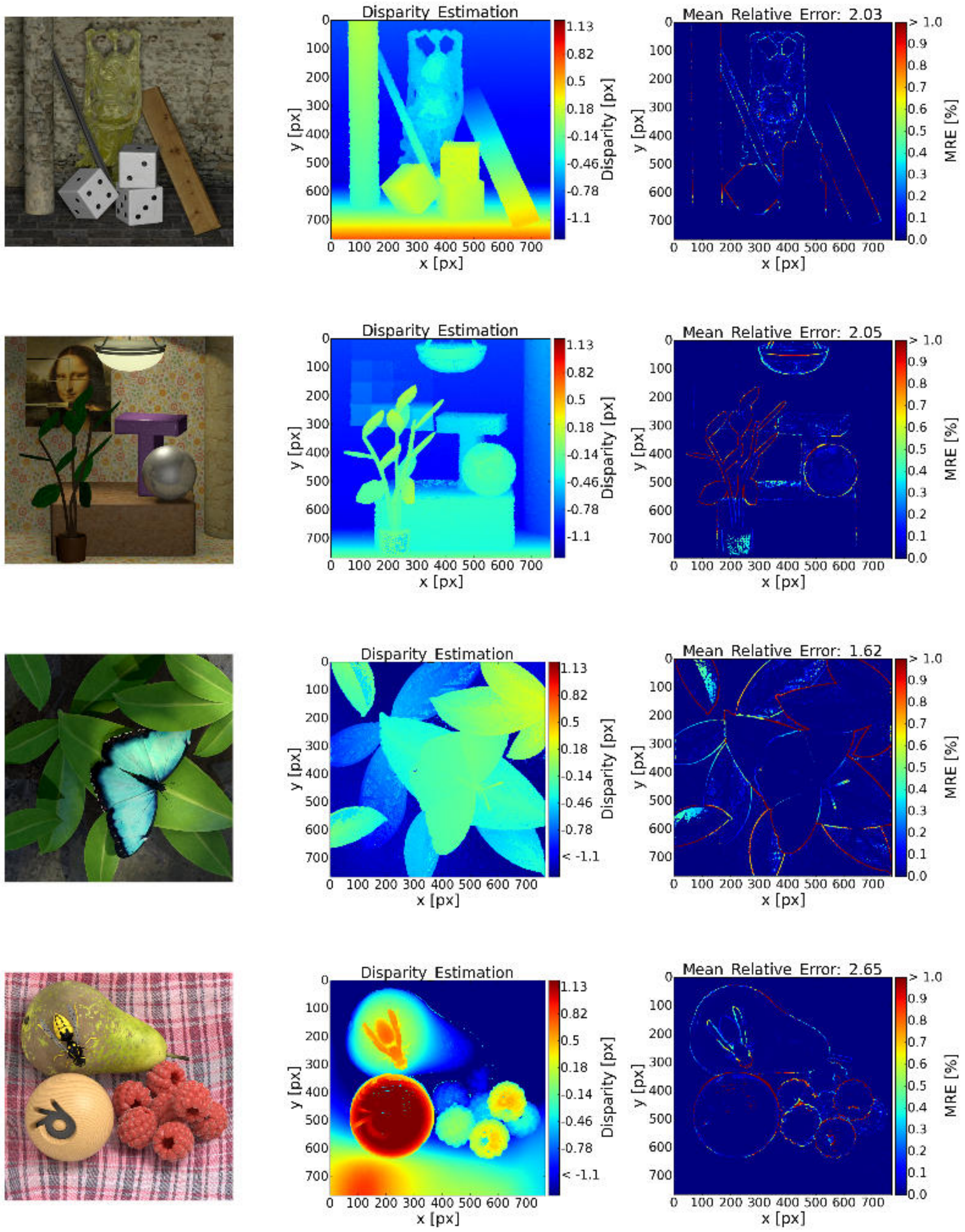


Figure B.2: *The first column shows the center view image. The second column shows the related disparity map. The last row visualizes the mean relative error distribution in the image. The chosen inner Gaussian smoothing is* $\sigma_{[5\times5]} = 0.5$ *and the outer Gaussian smoothing is* $\tau_{[9\times9]} = 1.3$

Disparity evaluation for the 3D Structure Tensor (Scharr filter)

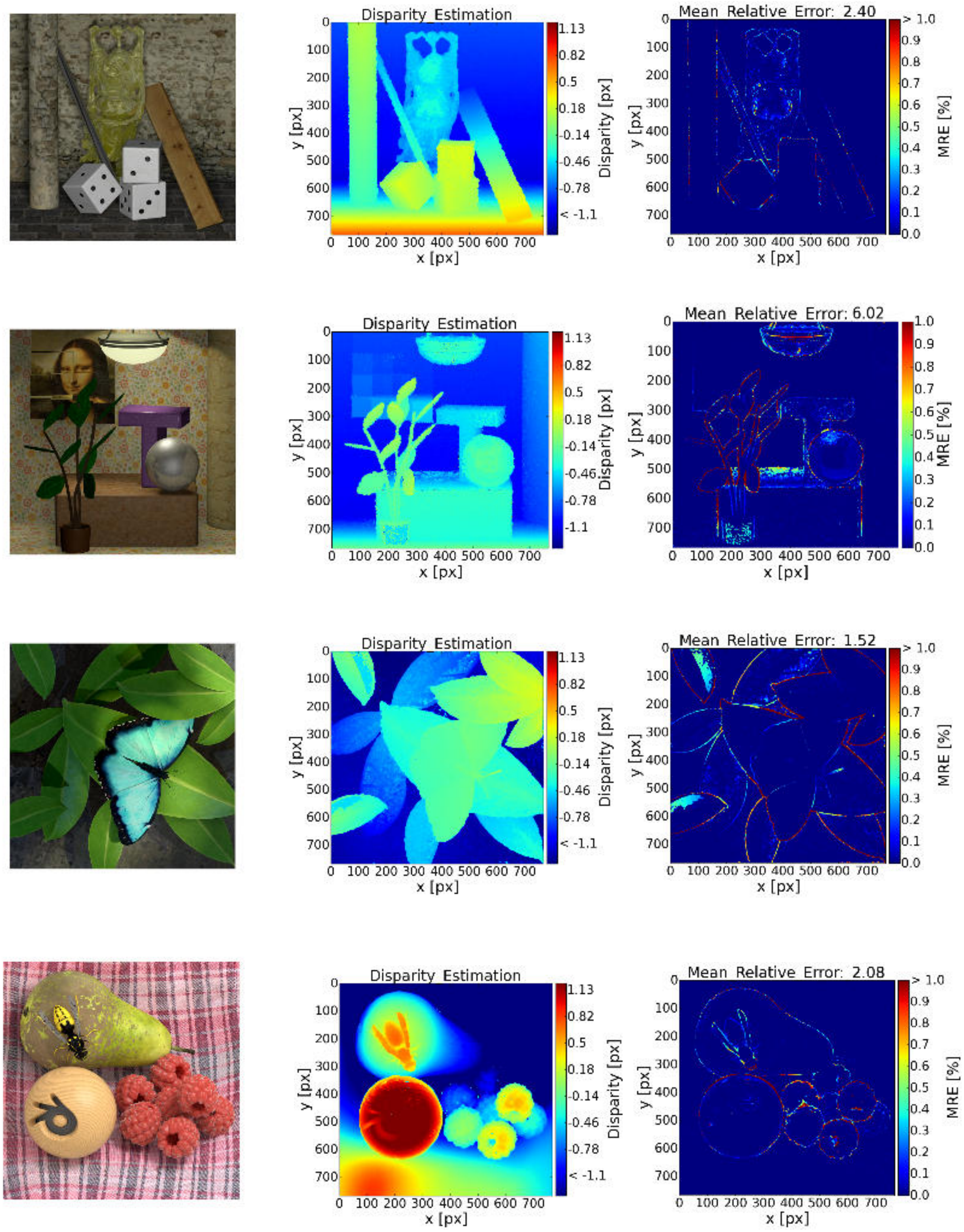


Figure B.3: *The first column shows the center view image. The second column shows the related disparity map. The last row visualizes the mean relative error distribution in the image. The chosen inner Gaussian smoothing is* $\sigma_{[5\times5]} = 0.5$ *and the outer Gaussian smoothing is* $\tau_{[9\times9]} = 1.3$

Disparity evaluation for the 4D Structure Tensor (Scharr filter)

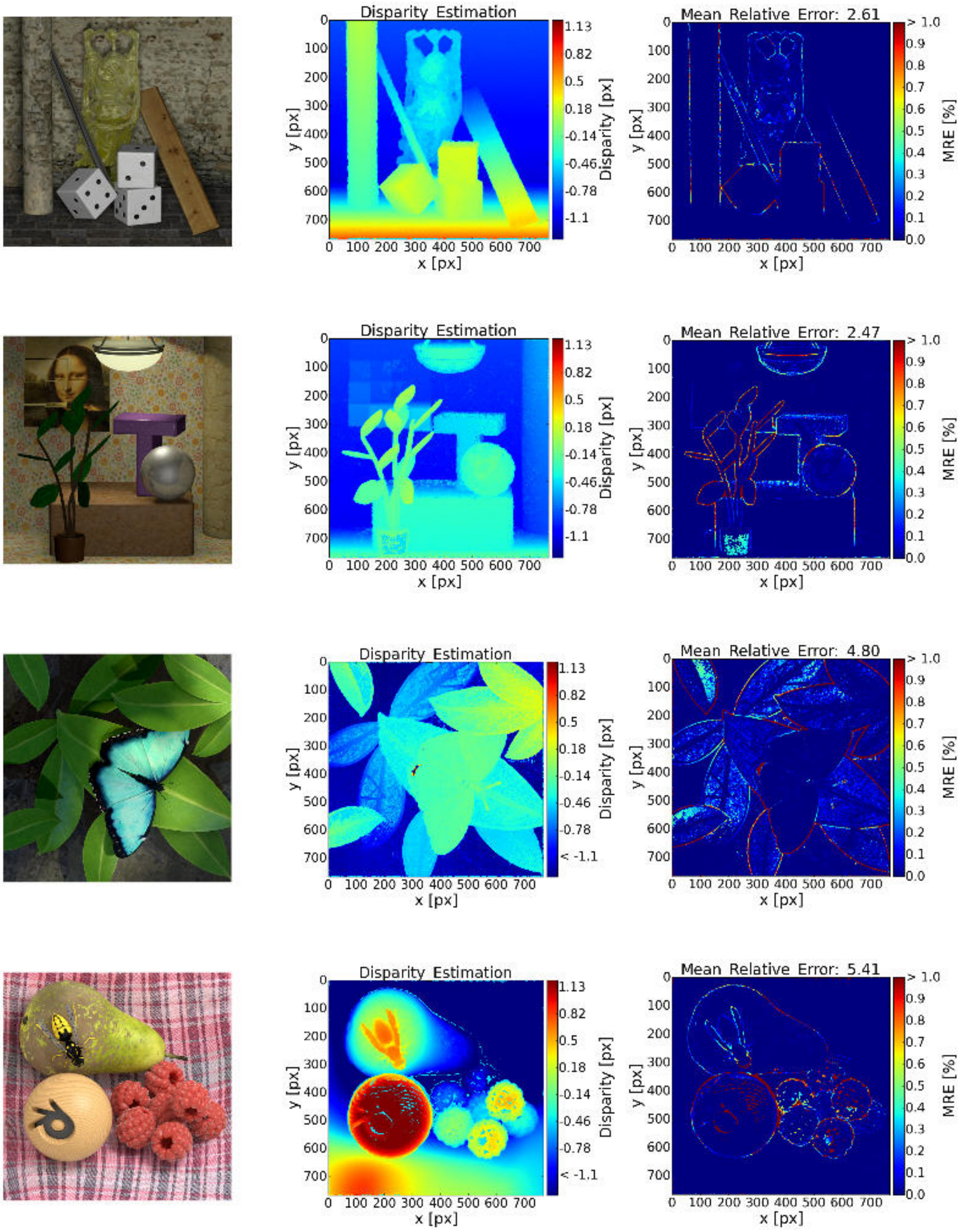
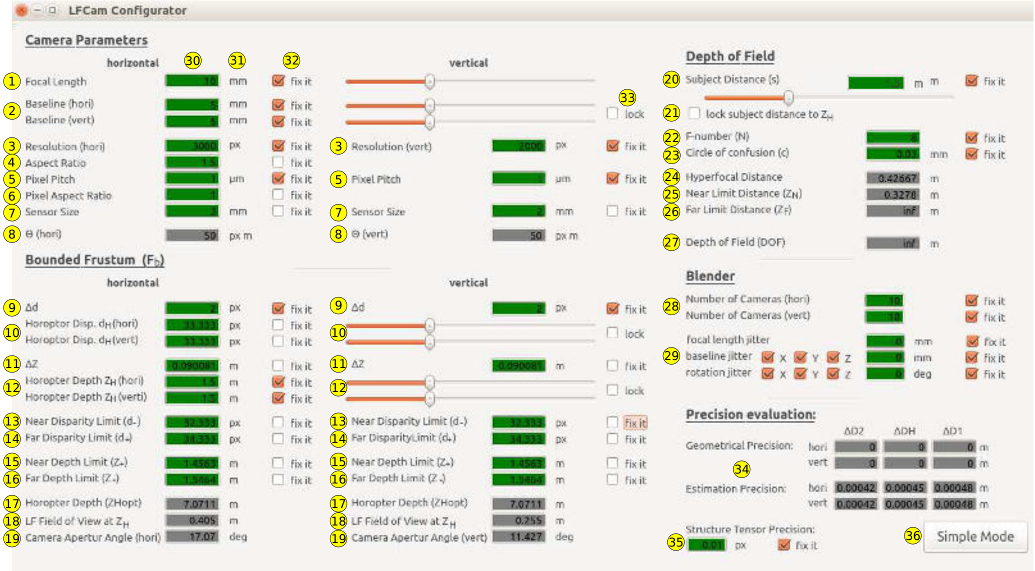


Figure B.4: *The first column shows the center view image. The second column shows the related disparity map. The last row visualizes the mean relative error distribution in the image. The chosen inner Gaussian smoothing is* $\sigma_{[5\times5]} = 0.5$ *and the outer Gaussian smoothing is* $\tau_{[9\times9]} = 1.3$

C Light-Field Camera Configurator



LFCam Configurator

Camera Parameters

horizontal 30 31 32

1 Focal Length 10 mm ☑ fix it
2 Baseline (hori) 5 mm ☑ fix it
Baseline (vert) 5 mm ☑ fix it
3 Resolution (hori) 3000 px ☑ fix it
4 Aspect Ratio 1.5 ☐ fix it
5 Pixel Pitch 1 µm ☑ fix it
6 Pixel Aspect Ratio 1 ☐ fix it
7 Sensor Size 3 mm ☐ fix it
8 Θ (hori) 50 px m

vertical

33 ☐ lock
3 Resolution (vert) 2000 px ☑ fix it
5 Pixel Pitch 1 µm ☑ fix it
7 Sensor Size 2 mm ☐ fix it
8 Θ (vert) 50 px m

Bounded Frustum (F$_b$)

horizontal

9 Δd 2 px ☑ fix it
10 Horoptor Disp. d$_H$ (hori) 33.333 px ☐ fix it
Horoptor Disp. d$_H$ (vert) 33.333 px ☐ fix it
11 ΔZ 0.090081 m ☐ fix it
12 Horopter Depth Z$_H$ (hori) 1.5 m ☑ fix it
Horopter Depth Z$_H$ (verti) 1.5 m ☑ fix it
13 Near Disparity Limit (d-) 32.333 px ☐ fix it
14 Far Disparity Limit (d+) 34.333 px ☐ fix it
15 Near Depth Limit (Z+) 1.4563 m ☐ fix it
16 Far Depth Limit (Z-) 1.5464 m ☐ fix it
17 Horopter Depth (ZHopt) 7.0711 m
18 LF Field of View at Z$_H$ 0.405 m
19 Camera Apertur Angle (hori) 17.07 deg

vertical

9 Δd 2 px ☑ fix it
10 ☐ lock
11 ΔZ 0.090081 m ☐ fix it
12 ☐ lock
13 Near Disparity Limit (d-) 32.333 px ☐ fix it
14 Far DisparityLimit (d+) 34.333 px ☐ fix it
15 Near Depth Limit (Z+) 1.4563 m ☐ fix it
16 Far Depth Limit (Z-) 1.5464 m ☐ fix it
17 Horopter Depth (ZHopt) 7.0711 m
18 LF Field of View at Z$_H$ 0.255 m
19 Camera Apertur Angle (vert) 11.427 deg

Depth of Field

20 Subject Distance (s) [illegible] m m ☑ fix it
21 ☐ lock subject distance to Z$_H$
22 F-number (N) 8 ☑ fix it
23 Circle of confusion (c) 0.03 mm ☑ fix it
24 Hyperfocal Distance 0.42667 m
25 Near Limit Distance (Z$_N$) 0.3278 m
26 Far Limit Distance (Z$_F$) inf m
27 Depth of Field (DOF) inf m

Blender

28 Number of Cameras (hori) 10 ☑ fix it
Number of Cameras (vert) 10 ☑ fix it
focal length jitter 0 mm ☑ fix it
29 baseline jitter ☑ X ☑ Y ☑ Z 0 mm ☑ fix it
rotation jitter ☑ X ☑ Y ☑ Z 0 deg ☑ fix it

Precision evaluation:

		ΔD2	ΔDH	ΔD1	
Geometrical Precision:	hori	0	0	0	m
	vert	0	0	0	m
Estimation Precision:	hori	0.00042	0.00045	0.00048	m
	vert	0.00042	0.00045	0.00048	m

34

Structure Tensor Precision:
35 0.01 px ☑ fix it

Figure C.1: *This figure shows the front end of the designed program. The program itself helps the user to find a valid light field camera setup. Thus the user gets informed about wrong placed parameter and is also supported by auto-completion if parameters can be computed.*

C.0.1 Detailed derivation of Theta

For the depth reconstruction of the obtained disparity map the camera parameter focal length f, baseline b and pixel pitch p are needed. These parameters define the relation between the depth Z and the disparity d, as given by the equation

$$Z_{[m]} = \frac{f_{[px]}\, b_{[m]}}{d_{[px]}} \quad \text{with} \quad f_{[px]} = \frac{f_{[m]}}{p_{[m]}}, \tag{C.1}$$

This equation maps the computed disparity information back onto a depth value and is not only valid for light-field imaging but also for stereo or multi-view stereo approaches. In light-field imaging using the Structure Tensor approach, it is only possible to compute the disparity in a 2 px range Δd around a given horopter disparity d_h as seen in figure 3.1. Thus the generalized disparity boundaries become

$$d_+ = d_H + \frac{\Delta d}{2}, \qquad d_- = d_H - \frac{\Delta d}{2} \tag{C.2}$$

which leads to

$$\Delta d = d_+ - d_-. \tag{C.3}$$

This constraint with equation C.1 provides the depth borders and the depth range

$$\Delta Z = Z_- - Z_+ \tag{C.4}$$

where, to keep the depth range positive, the related depth border values swap position. When the target scene exceeds the depth borders the detectable disparity range is also exceeded, and a global shift needs to be applied to keep the slopes in measurable range. This process of global shifting was introduced in Diebold *et al.*[20] and explained in chapter 4 for extending the measurable range of light-field processing for sparsely sampled data sets.

On the other hand, if the target scene depth range is smaller than ΔZ then the disparity resolution attained is less than possible. Thus it can be worth knowing the range of scene depth in order to adapt one's processing to attain the highest quality light-field range estimates.

Knowing the range of depths ΔZ at a certain distance Z_H and knowing the disparity range Δd, a constraint for the principal camera parameter is attained through equation C.1 in equation C.2 and used to establish depth borders, leading to

$$Z_- = \frac{\frac{f_{[m]}\, b_{[m]}}{p_{[m]}} Z_H}{\left(\frac{f_{[m]}\, b_{[m]}}{p_{[m]}} - \Delta d Z_H\right)} \tag{C.5}$$

$$Z_+ = \frac{\frac{f_{[m]}\, b_{[m]}}{p_{[m]}} Z_H}{\left(\frac{f_{[m]}\, b_{[m]}}{p_{[m]}} + \Delta d Z_H\right)} \tag{C.6}$$

These results are inserted in equation C.3 and rearrangement to the principal camera parameter. The resulting equation becomes

$$\frac{f_{[m]}\, b_{[m]}}{p_{[m]}} = \frac{\Delta d Z_H^2 + \Delta d Z_H \sqrt{Z_H^2 + \Delta Z^2}}{\Delta Z} \tag{C.7}$$

where the principal camera parameter reduces to a single conceptual parameter

$$\theta(b, f, p) = \frac{f_{[m]}\, b_{[m]}}{p_{[m]}} := \theta(\Delta Z, \Delta d, Z_H) \tag{C.8}$$

This variable θ defines the linkage between depth range ΔZ at a given distance Z_H and the disparity range Δd constraints with the camera parameters. This makes both directly comparable and simplifies definition of the light-field camera setup, since the depth borders are more often of interest.

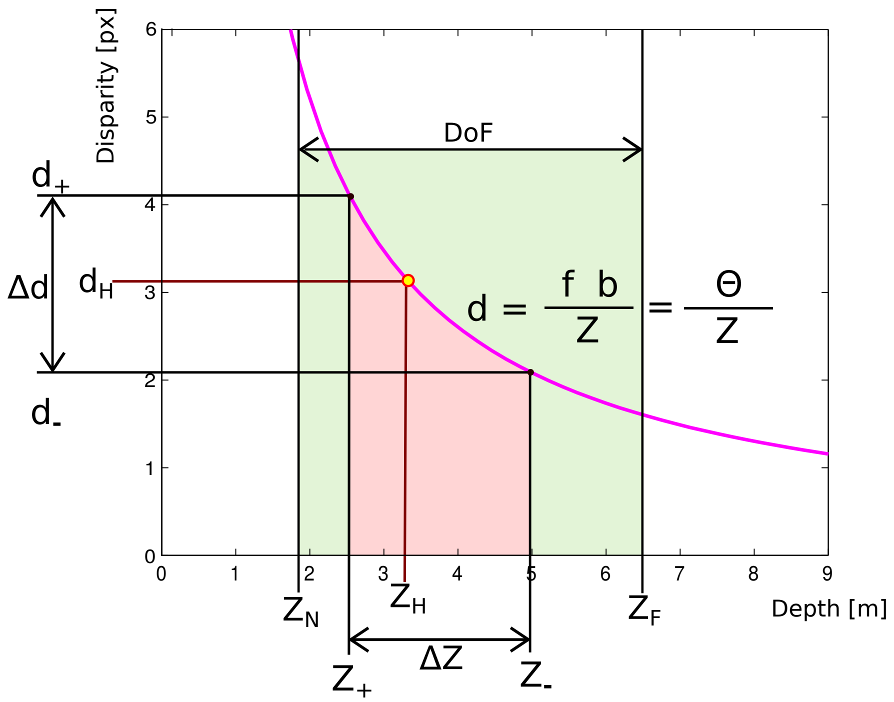


Figure C.2: *Illustrates the projection of the depth domain to the disparity domain for a given camera array setup.*

C.0.2 The bounded frustum

In addition to a given depth range, the frame size F_H of my camera setup is also defined by the principal camera parameter. In light-field imaging the resulting frame size is defined by the visible depth content of all cameras. Thus it becomes

$$F_H = \frac{R_{[px]} Z_H}{f_{[px]}} - b_{[m]} C \tag{C.9}$$

where $R_{[px]}$ is the image resolution and C the number of cameras employed. Together with the depth range constraint, this spans a frustum-shaped volume which defines the measurable extent of the light-field camera. Objects to be measured must lie inside this frustum.

In addition, the depth of field DoF needs to enclose the defined frustum, as shown in figure 3.1. That ensures that all captured images are sharp in the range of interest. These boundary conditions and the fact that just a finite number of objectives and cameras are available limit possible setup configurations in their minimal and maximal achievable baseline b and focal length f. Capturing a light-field scene without knowledge of the setup will lead to poor use of the data. The complexity of the controllable parameters and the constraints coupling depth of field, field of view, and the bounded frustum make it difficult to design a setup satisfying all requirements. In order to prevent unintended results and to provide adequate control over the capturing process,

it is important to put thought into the setup. With this in mind, it became clear that an interactive utility supporting these calculations would be useful.
We designed a toolbox, shown in figure C.1, that automatically completes the setup, where possible, and also informs the user when parameters are inconsistent with the overall setup.

C.0.3 Usage of the light field camera configurator

Parameters which can be derived from entered values are automatically inserted and will be updated when changing related input parameters. Parameters are interpreted as inputs when the "fix it" flag ㉞ is tagged. An active fix it flag locks the related parameter and the program can no longer change its value. Only values not so flagged will be adapted automatically (the flag can be unset at any time).
If a valid value has been entered, the input window ㉜ is set to green, indicating that it is consistent with other related values. When the entered value is not consistent, or it cannot yet be compute, the input window is set to red. Information outputs which require no input are grayed. Next we will introduce one method to define a light-field setup. In general there are several different methods to find a suitable setup, but we focus on the most intuitive one, starting with the principal camera parameter.

Step one: The light-field camera parameter

To define a light-field camera setup we start with the camera parameters (① - ⑦). These parameters can be selected as initial requirement but can also be fixed if the camera is known. Without a known camera, it is better to start with the bounded frustum. Note that for a fully defined camera parameter set, it is not necessary that all input windows ㉜ be set.

C.0.4 Step two: The bounded frustum

First is to establish the disparity range around the selected horopter (⑨, ⑩). Alternatively, the near disparity ⑭ and the far disparity ⑬ can be entered. If a known camera is selected, the bounded frustum is defined. An alternative way to define the bounded frustum is to define the relative depth values instead of the disparity values (⑪, ⑫ or ⑮, ⑯). When the camera setup is not defined, the bounded frustum can be selected first by using the depth range ΔZ ⑪, the disparity range Δd ⑨ and a depth location such as Z_H. After entering these parameters, the value θ ⑧ can be computed, which provides initial constraints on the camera setup.

C.0.5 Step three: Depth of Field

The depth of field ㉗ is an equally important part of the light-field camera configurator. It is defined as the difference between the near limit distance ㉕ and the far limit

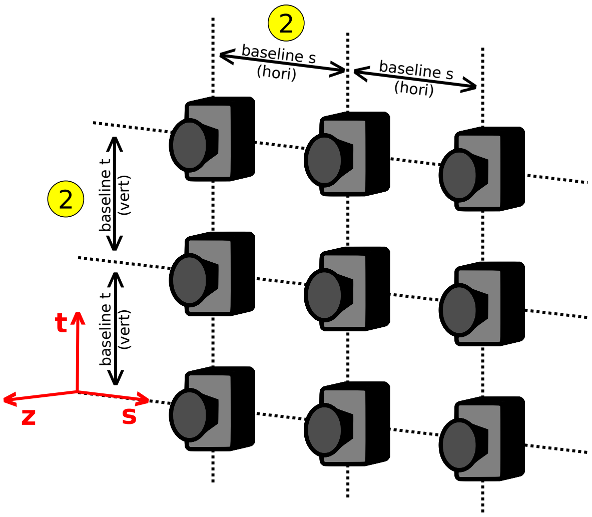


Figure C.3: *Camera array example. The baseline defines the distance from either two neighboring vertical or two horizontal cameras.*

distance ㉖. To compute depth of field, the subject distance ⑳, the f-number ㉒ and the circle of confusion ㉓ are needed. A rule of thumb is to choose the circle of confusion as twice the size of the pixel pitch to guarantee optimal sharpness. Additional information about the depth of field gives the hyperfocal distance ㉔. It is a special subject distance beyond which the far limit distance becomes infinity.

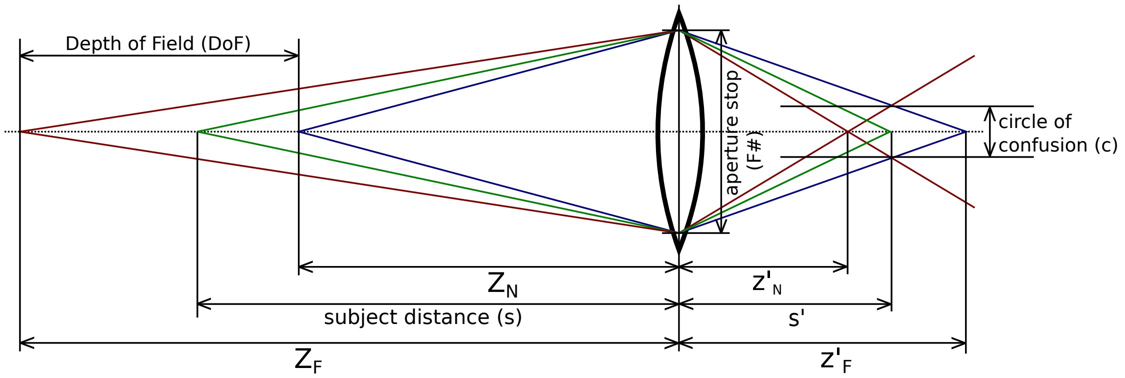


Figure C.4: *Shows the relation between the Depth of Field and the circle of confusion size.*

C.0.6 Step four: Blender and precision evaluation

To export the setup to the blender environment, the number of vertical and horizontal cameras ㉘ are required. Only rectangular grid structures can be exported.

For an evaluation of errors occurring due to misaligned cameras or varying focal lengths, a baseline jitter ㉚ or a focal length jitter ㉙ can be applied. Additionally, a misalignment of the camera orientation ㉛ can be applied. All entered jitter values are normally distributed and added to the mean values of each camera respectively. The entered values define the standard deviation of the normal distribution.

Finally, the geometrical and estimation precisions are computed ㉞. In the first line the geometrical precision is computed, which describes the depth computation error (standard deviation) with respect to a baseline jitter. In the second line the estimation precision is computed, relating to the disparity ㉟ value. For the Structure Tensor method, this describes the minimal distance distinguishable between two disparities.

Parameter list

1	$f_{[m]}$	Focal length for each camera in the array.
2	$b_{[m]}$	The distance of two neighboring cameras.
3	$R_{x[px]}$,$R_{y[px]}$	Image resolution.
4	β	The aspect ratio, computed from the image size.
5	$P_{[m]}$	Distance between two neighboring photo-sensitive areas (same color).
6	α	Pixel aspect ratio, computed from the pixel pitch.
7	$S_{[m]}$	Sensor size resulting from Pixel Pitch and Resolution.
8	$\theta_{[px\cdot m]}$	Theta is defined as the product of baseline (2) and focal length (1).
9	$\Delta d_{[px]}$	The disparity range in which one wants to measure disparities .
10	$d_{H[px]}$	The disparity horopter. Related to depth horopter
11	$\Delta D_{[m]}$	Depth range, which is defined by the disparity range.
12	$Z_{H[m]}$	Horopter depth is the counterpart of the horopter disparity .
13	$d_{-[px]}$	The minimum disparity value.
14	$d_{+[px]}$	The maximum disparity value.
15	$Z_{+[m]}$	The minimum allowed depth value.
16	$Z_{-[m]}$	The maximum allowed depth value.
17	$Z_{Hopt[m]}$	Turning point in depth-disparity-diagram.
18	$F_{[m]}$	Resulting Field of View of light-field camera.
19	$\gamma_{[\mathrm{deg}]}$	Aperture Angle of the camera setup.
20	$s_{[m]}$	Focus point of each camera lens (needed to compute the DoF).
21		Subject Distance can be locked to the horopter depth.
22	$F\#$	focal ratio, f-ratio, f-stop.
23	$c_{[m]}$	Circle of confusion, necessary to compute the depth of field.
24	$H_{[m]}$	Distance to focus where far limit distance is at infinity.
25	$Z_{N[m]}$	Near distance limit of acceptable sharpness.
26	$Z_{F[m]}$	Far distance limit of acceptable sharpness.
27	$DoF_{[m]}$	Depth of field. Difference between near and far distances.
28		Numbers of cameras in a light-field array. (rectangular grids only)
29	Δf,Δb,Δr	Additive normally distributed focal length inaccuracy, baseline jitter (Axis eligible) and rotation jitter (Axis eligible).
30		Input window.
31		Units of the values.
32		”Fix it” flags to define fixed input values.
33		Lock tag, vertical and horizontal direction changes parallel.
34		Depth estimation inaccuracies for Structure Tensor estimation and geometrical inaccuracies.
35	Δd_{geo},Δd_{st}	Structure Tensor precision assumption.
36		Simple mode and extended mode switch.

C.0.7 Equations List:

9	$\Delta d = d_+ - d_H$	(C.10)
9	$\Delta d = d_H - d_-$	(C.11)
11	$\Delta Z = Z_- - Z_H$	(C.12)
11	$\Delta Z = Z_H - Z_+$	(C.13)
11	$\Delta Z = Z_- - Z_+$	(C.14)
5	$P_{[m]} = \frac{S}{R}$	(C.15)
5	$P_y = \frac{P_x}{\alpha}$	(C.16)
5	$P_x = P_y \alpha$	(C.17)
24	$H = \frac{f_{[m]}^2}{Fc} + f_{[m]}$	(C.18)
25	$Z_N = \frac{\left(H - f_{[m]}\right) s}{H + \left(s - 2f_{[m]}\right)}$	(C.19)
26	$Z_F = \frac{\left(H - f_{[m]}\right) s}{H - s}$	(C.20)
27	$DoF = Z_F - Z_N$	(C.21)
18	$F_H = \frac{SZ_H}{f} - b_{[m]}C$	(C.22)
7	$S = RP_{[m]}$	(C.23)
7	$S = \frac{R_x S}{R_y}$	(C.24)
7	$S = \frac{R_y S}{R_x}$	(C.25)
4	$\beta = \frac{R_x}{R_y}$	(C.26)
2	$b_m = \frac{\theta_{px} P_{[m]}}{f_{[m]}}$	(C.27)
6	$\alpha = \frac{P_x}{P_y}$	(C.28)
17	$Z_{Hopt} = \sqrt{f_{[m]} b_{[m]}}$	(C.29)
14	$d_+ = 2\Delta d + d_-$	(C.30)
14	$d_+ = \Delta d + d_H$	(C.31)
14	$d_+ = \frac{\theta_{px} P_{[m]} R}{SZ_+}$	(C.32)
13	$d_- = d_+ - 2\Delta d + d_-$	(C.33)
13	$d_- = d_H - \Delta d$	(C.34)
13	$d_- = \frac{\theta_{px} P_{[m]} R}{SZ_-}$	(C.35)
	$f_{px} = \frac{f_{[m]} R}{S}$	(C.36)
	$f_{px} = \frac{\theta_{px} P_{[m]} R}{bS}$	(C.37)
1	$f_{[m]} = \frac{\theta_{px} P_{[m]}}{b}$	(C.38)
10	$d_H = \frac{d1 - d_+}{2}$	(C.39)
10	$d_H = \frac{\theta_{px} P_{[m]} R}{Z_H S}$	(C.40)
16	$Z_- = \Delta Z + Z_+$	(C.41)
15	$Z_+ = Z_- - \Delta Z$	(C.42)
3	$R = \frac{S}{P_{[m]}}$	(C.43)
3	$R = \beta R_y$	(C.44)
3	$R = \frac{R_x}{Asp}$	(C.45)
3	$b_m = \frac{\theta_{px} P}{f_{[m]}}$	(C.46)
8	$\theta_{px} = \frac{f_{[m]} b_{[m]}}{P_{[m]}}$	(C.47)

9 $$\Delta d = \frac{\theta_{px}\left(Z_H - Z_+\right)}{Z_H Z_+} \tag{C.48}$$

9 $$\Delta d = \frac{\theta_{px}\left(Z_- - Z_H\right)}{Z_H Z_-} \tag{C.49}$$

16 $$Z_- = \frac{\theta_{px} Z_H}{\left(\theta_{px} - \Delta d Z_H\right)} \tag{C.50}$$

15 $$Z_+ = \frac{\theta_{px} Z_H}{\left(\theta_{px} + \Delta d Z_H\right)} \tag{C.51}$$

12 $$Z_H = \frac{Z_- \theta_{px}}{\left(\theta_{px} + \Delta d Z_-\right)} \tag{C.52}$$

12 $$Z_H = \frac{Z_+ \theta_{px}}{\left(\theta_{px} - \Delta d Z_+\right)} \tag{C.53}$$

12 $$Z_H = \frac{\theta_{px}}{d_H} \tag{C.54}$$

19 $$\gamma = \frac{360}{\Pi} \cdot \arctan\left(\frac{S}{2 f_{[m]}}\right) \tag{C.55}$$

$$\theta_{px} = \frac{\Delta d Z_H^2 + \Delta d Z_H \sqrt{Z_H^2 + \Delta Z^2}}{\Delta Z} \tag{C.56}$$

34 $$\Delta Z_{geo} = Z \sqrt{\left(\frac{\Delta b}{b}\right)^2 + \left(\frac{\Delta f}{f}\right)^2} \quad D \in \{Z_+, ..., Z_-\} \tag{C.57}$$

34 $$\Delta Z_{st} = \frac{Z^2 \Delta d_{st}}{b f_{px}} \quad Z \in \{Z_+, ..., Z_-\} \tag{C.58}$$

3 $$R = \frac{Z_- Z_H \Delta Z S}{\theta_{px} P_{[m]} \left(Z_H - Z_+\right)} \tag{C.59}$$

3 $$R = \frac{Z_H^2 \Delta d S + \Delta d S Z_H \sqrt{Z_H + \Delta Z}}{\Delta Z \theta_{px} P_{[m]}} \tag{C.60}$$

3 $$R = \frac{Z_- Z_H \Delta Z S}{\theta_{px} P_{[m]} \left(Z_1 - Z_H\right)} \tag{C.61}$$

7 $$S = \frac{\theta_{px} P_{[m]} R \left(Z_- - Z_H\right)}{Z_H Z_- \Delta d} \tag{C.62}$$

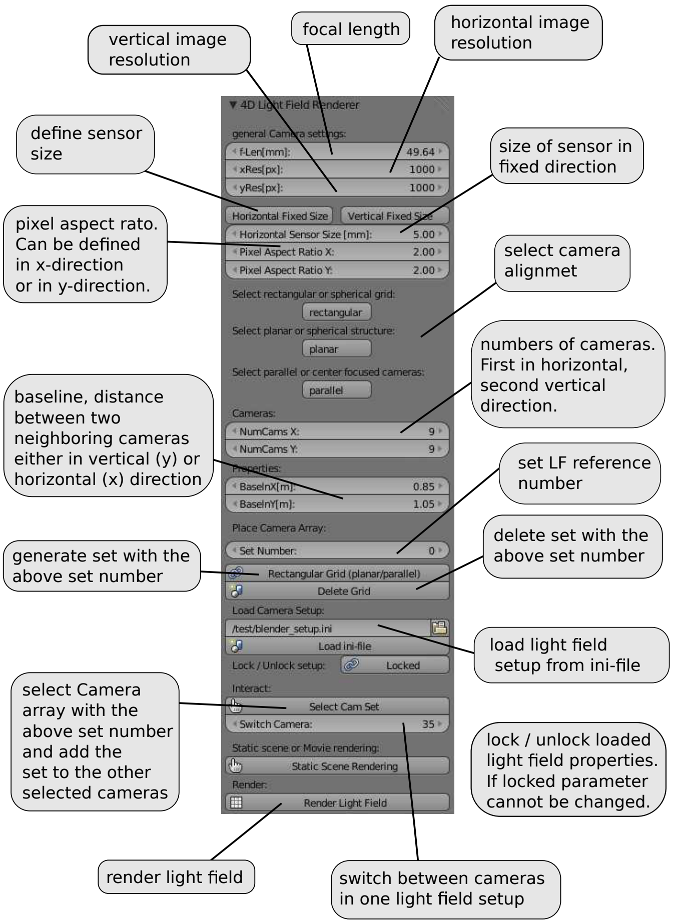
7 $$S = \frac{\theta_{px} P_{[m]} R \left(Z_H - Z_+\right)}{Z_H Z_+ \Delta d} \tag{C.63}$$

7 $$S = \frac{\theta_{px} P_{[m]} \Delta Z R}{\Delta d \left(Z_H^2 + Z_H \sqrt{Z_H + \Delta Z}\right)} \tag{C.64}$$

D Light-Field Acquisition Toolbox

For the evaluation of synthetic generated images we implemented a light-field acquisition toolbox for bender. Blender [13] is a free software to render images or movies of 3D modeled scenes. This software is used because it provides depth information with respect to the rendered images and the possibility to write own extensions in python 3. The programmed blender toolbox is also written in python 3 and simplifies the placement of camera arrays to capture light fields of static or dynamic scenes. An image of the control terminal of the programmed toolbox is shown in figure D.1.

The toolbox is suitable to set all light-field camera related parameters. Additional it is possible to load predefine setups generated with the light-field camera configurator as introduced in C. That enables the visualization of the frustum for the defined setup and makes it possible to place scene content at the computed distance as shown in figure D.6. Next we want to give some examples about different possible light field camera configurations possible to create directly with the toolbox.



▼ 4D Light Field Renderer

general Camera settings:

f-Len[mm]: 49.64

xRes[px]: 1000

yRes[px]: 1000

Horizontal Fixed Size | Vertical Fixed Size

Horizontal Sensor Size [mm]: 5.00

Pixel Aspect Ratio X: 2.00

Pixel Aspect Ratio Y: 2.00

Select rectangular or spherical grid: rectangular

Select planar or spherical structure: planar

Select parallel or center focused cameras: parallel

Cameras:

NumCams X: 9

NumCams Y: 9

Properties:

BaselnX[m]: 0.85

BaselnY[m]: 1.05

Place Camera Array:

Set Number: 0

Rectangular Grid (planar/parallel)

Delete Grid

Load Camera Setup:

/test/blender_setup.ini

Load ini-file

Lock / Unlock setup: Locked

Interact:

Select Cam Set

Switch Camera: 35

Static scene or Movie rendering:

Static Scene Rendering

Render:

Render Light Field

vertical image resolution

focal length

horizontal image resolution

define sensor size

size of sensor in fixed direction

pixel aspect rato. Can be defined in x-direction or in y-direction.

select camera alignmet

numbers of cameras. First in horizontal, second vertical direction.

baseline, distance between two neighboring cameras either in vertical (y) or horizontal (x) direction

set LF reference number

generate set with the above set number

delete set with the above set number

load light field setup from ini-file

select Camera array with the above set number and add the set to the other selected cameras

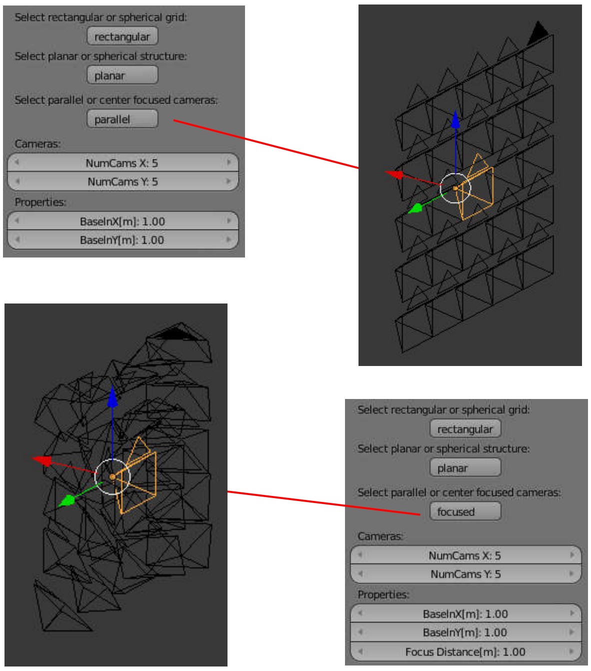
lock / unlock loaded light field properties. If locked parameter cannot be changed.

render light field

switch between cameras in one light field setup

Figure D.1: *Shows the toolbox embedded in blender to define light-field camera setups.*

Regular light-field configuration

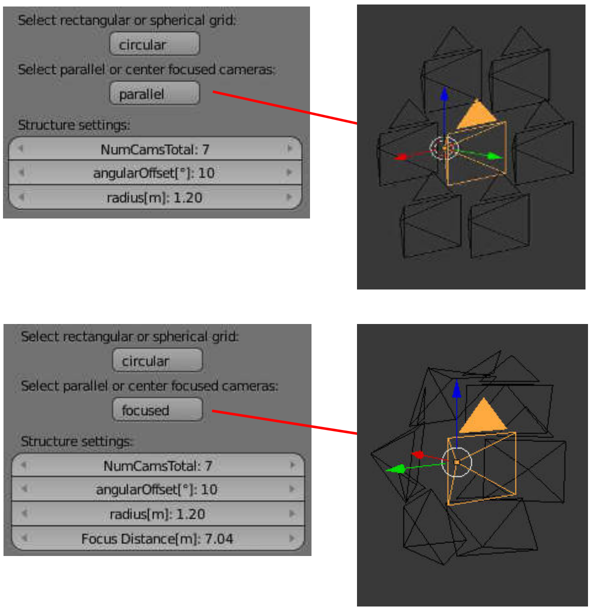


Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
planar
Select parallel or center focused cameras:
parallel
Cameras:
NumCams X: 5
NumCams Y: 5
Properties:
BaseInX[m]: 1.00
BaseInY[m]: 1.00

Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
planar
Select parallel or center focused cameras:
focused
Cameras:
NumCams X: 5
NumCams Y: 5
Properties:
BaseInX[m]: 1.00
BaseInY[m]: 1.00
Focus Distance[m]: 1.00

Figure D.2: *This images show two setup examples of a regular light-field configuration. The first for a parallel aligned cameras and the second with focused cameras.*

Circular light-field configuration

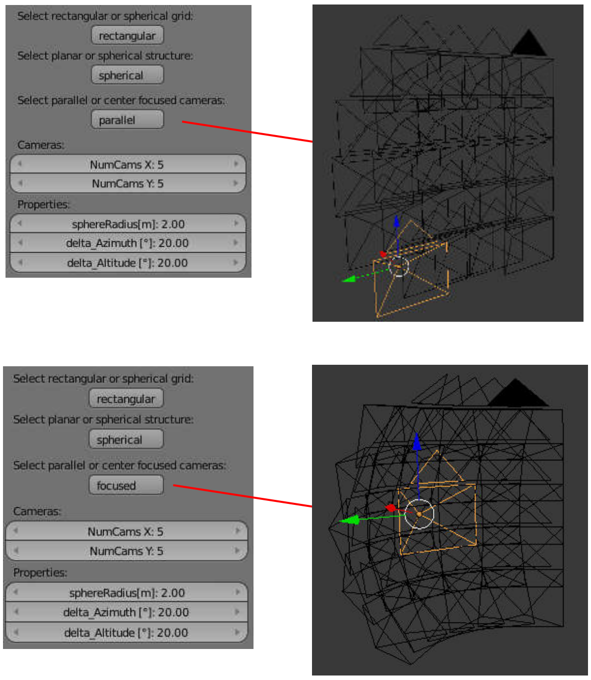


Select rectangular or spherical grid:
circular
Select parallel or center focused cameras:
parallel
Structure settings:
NumCamsTotal: 7
angularOffset[°]: 10
radius[m]: 1.20

Select rectangular or spherical grid:
circular
Select parallel or center focused cameras:
focused
Structure settings:
NumCamsTotal: 7
angularOffset[°]: 10
radius[m]: 1.20
Focus Distance[m]: 7.04

Figure D.3: *This images show two setup examples of a circular light-field configuration. The first for a parallel aligned cameras and the second with focused cameras.*

Spherical light-field configuration

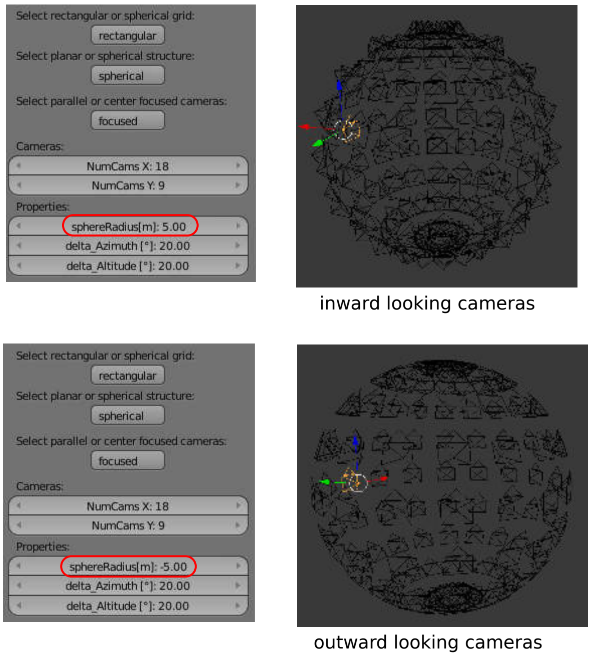


Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
spherical
Select parallel or center focused cameras:
parallel
Cameras:
NumCams X: 5
NumCams Y: 5
Properties:
sphereRadius[m]: 2.00
delta_Azimuth [°]: 20.00
delta_Altitude [°]: 20.00

Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
spherical
Select parallel or center focused cameras:
focused
Cameras:
NumCams X: 5
NumCams Y: 5
Properties:
sphereRadius[m]: 2.00
delta_Azimuth [°]: 20.00
delta_Altitude [°]: 20.00

Figure D.4: *This images show two setup examples of a spherical light-field configuration. The first for a parallel aligned cameras and the second with focused cameras.*

Special light-fields configuration



Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
spherical
Select parallel or center focused cameras:
focused
Cameras:
NumCams X: 18
NumCams Y: 9
Properties:
sphereRadius[m]: 5.00
delta_Azimuth [°]: 20.00
delta_Altitude [°]: 20.00

inward looking cameras

Select rectangular or spherical grid:
rectangular
Select planar or spherical structure:
spherical
Select parallel or center focused cameras:
focused
Cameras:
NumCams X: 18
NumCams Y: 9
Properties:
sphereRadius[m]: -5.00
delta_Azimuth [°]: 20.00
delta_Altitude [°]: 20.00

outward looking cameras

Figure D.5: *This images show two setup examples of a special light-field configuration. The first for inward looking cameras and the second for outward looking cameras.*

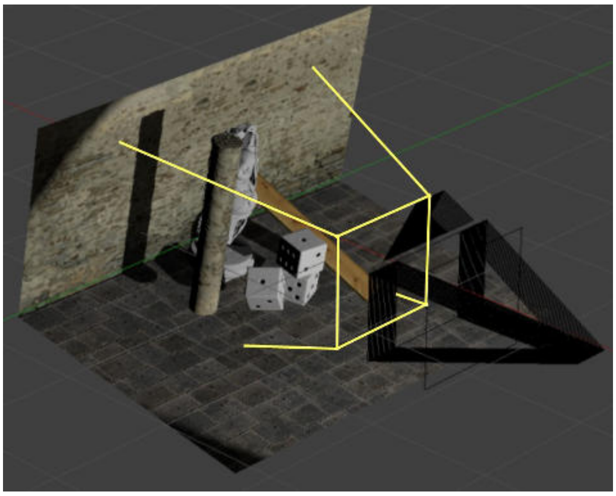


__Figure D.6:__ Shows a loaded setup defined with the light-field camera configurator. Now aside the camera array also the related frustum is displayed. Scenes can now be positioned inside the frustum.

E Heterogeneous Light Fields

E.1 Color filtered light field image data

The supplementary material contains all captured image data with respect to the real captured color filtered light fields used for the RGB reconstruction. Additionally we provide RGB captured images of the objects as reference for the colors. The RGB images are taken with a Sony ILCE-7E camera while the light-field data is captured with a PCO edge 5.5 camera.

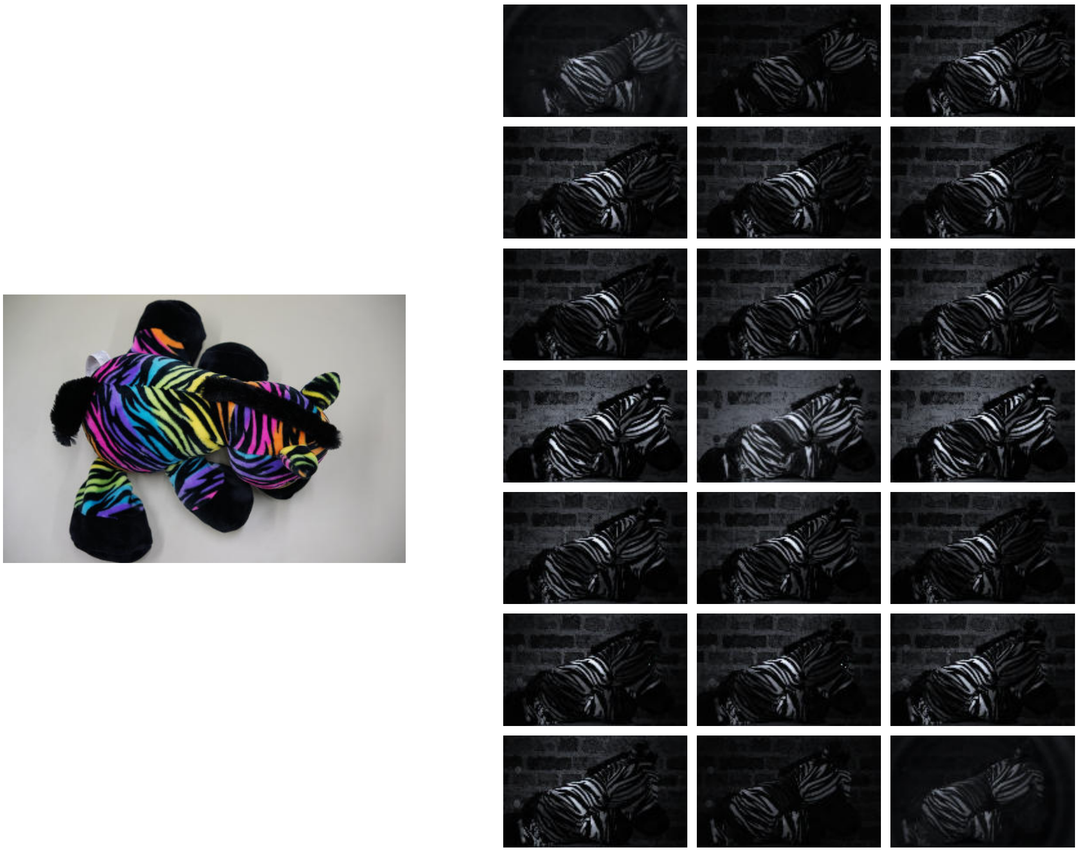


Figure E.1: *The horse scene. The RGB image is taken with a Sony ILCE-7E camera. It is not representing the ground truth color but illustrates the color distribution of the object. The heterogeneous light field is captured with a PCO edge 5.5 camera. The captured heterogeneous light field contains 21 images (top left to bottom right). The filters employed have a full width half mean value of* 10 *nm with the given mean values:* 400 *nm,* 450 *nm,* 500 *nm,* 515 *nm,* 532 *nm,* 550 *nm,* 560 *nm,* 589 *nm,* 600 *nm,* 650 *nm,* 700 *nm and back down to* 400 *nm.*

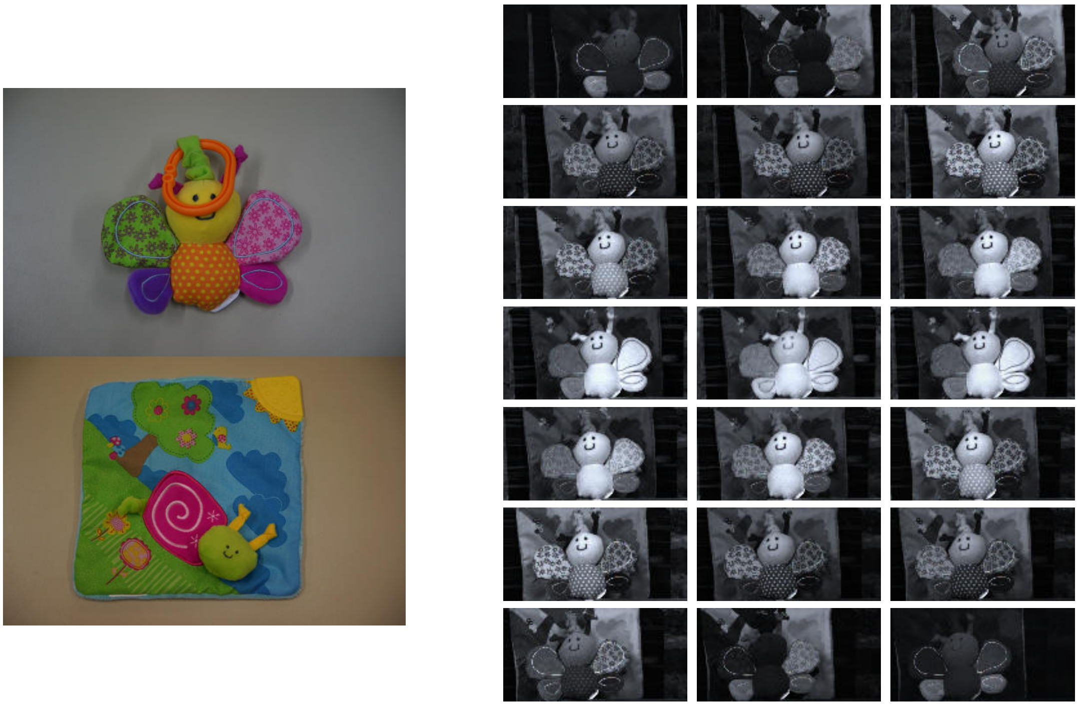


Figure E.2: *The butterfly scene. The RGB image is taken with a Sony ILCE-7E camera. It is not representing the ground truth color but illustrates the color distribution of the object. The captured heterogeneous light field contains 21 images (top left to bottom right). The filters employed have a full width half mean value of* 10 *nm with the given mean values:* 400 *nm,* 450 *nm,* 500 *nm,* 515 *nm,* 532 *nm,* 550 *nm,* 560 *nm,* 589 *nm,* 600 *nm,* 650 *nm,* 700 *nm and back down to* 400 *nm.*

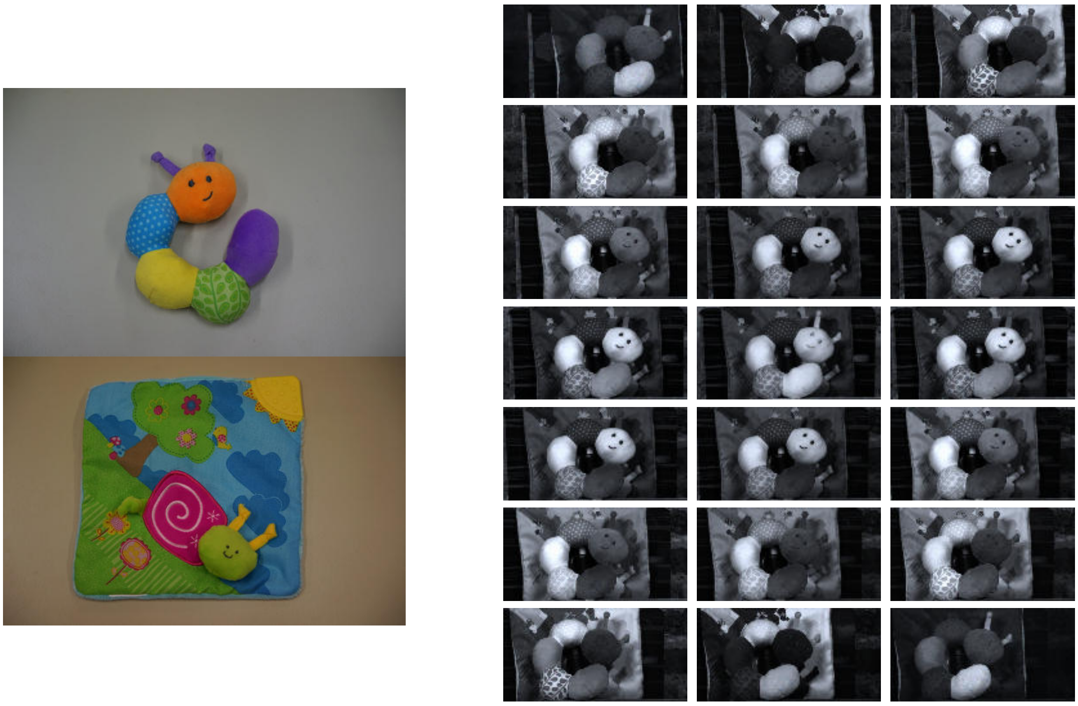


Figure E.3: *The worm scene. The RGB image is taken with a Sony ILCE-7E camera. It is not representing the ground truth color but illustrates the color distribution of the object. The heterogeneous light field is captured with a PCO edge 5.5 camera. The captured heterogeneous light field contains 21 images (top left to bottom right). The filters employed have a full width half mean value of* 10 *nm with the given mean values:* 400 *nm,* 450 *nm,* 500 *nm,* 515 *nm,* 532 *nm,* 550 *nm,* 560 *nm,* 589 *nm,* 600 *nm,* 650 *nm,* 700 *nm and back down to* 400 *nm.*

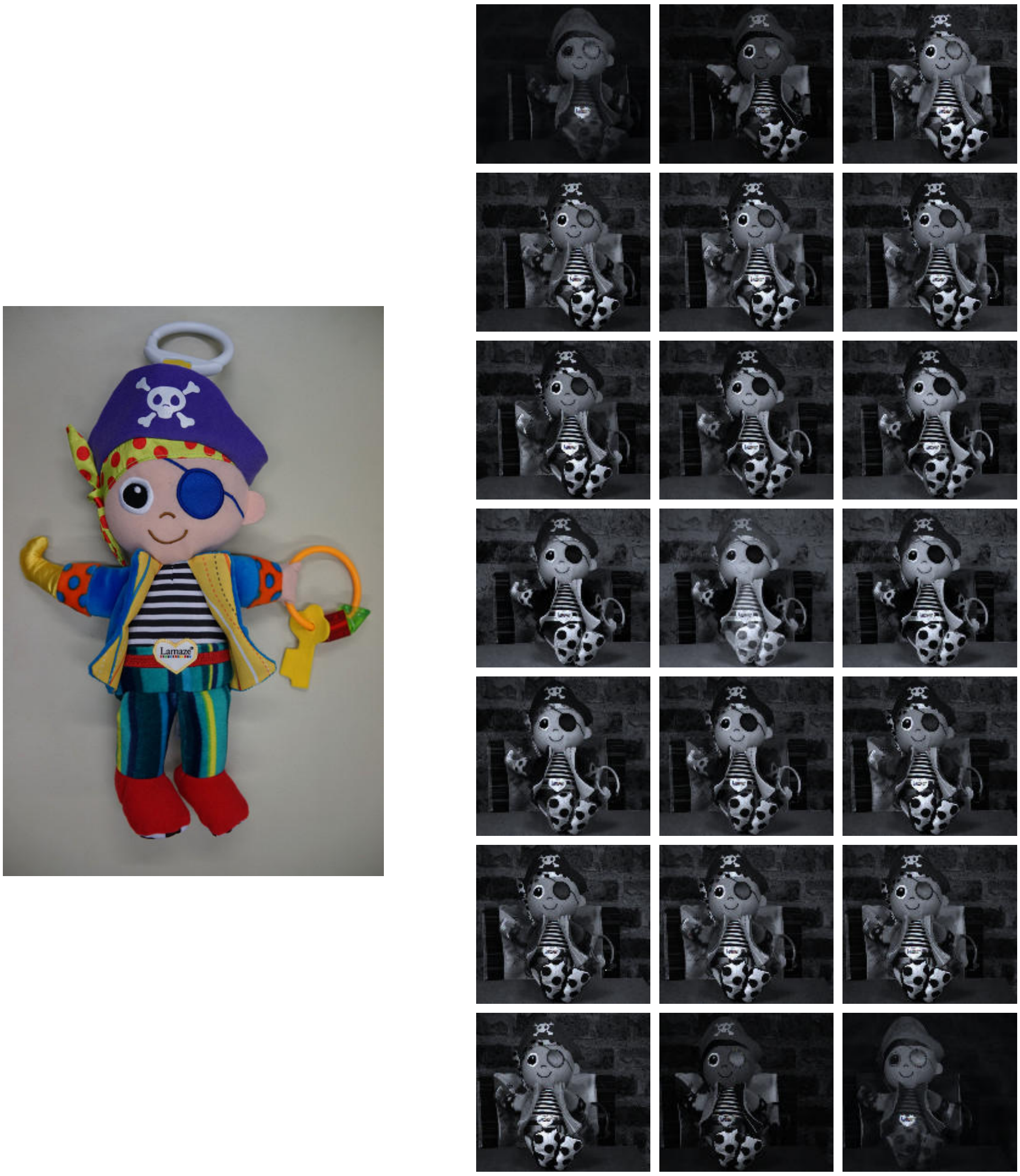


Figure E.4: *The pirate scene. The RGB image is taken with a Sony ILCE-7E camera. It is not representing the ground truth color but illustrates the color distribution of the object. The heterogeneous light field is captured with a PCO edge 5.5 camera. The captured heterogeneous light field contains 21 images (top left to bottom right). The filters employed have a full width half mean value of* 10 *nm with the given mean values:* 400 *nm,* 450 *nm,* 500 *nm,* 515 *nm,* 532 *nm,* 550 *nm,* 560 *nm,* 589 *nm,* 600 *nm,* 650 *nm,* 700 *nm and back down to* 400 *nm.*

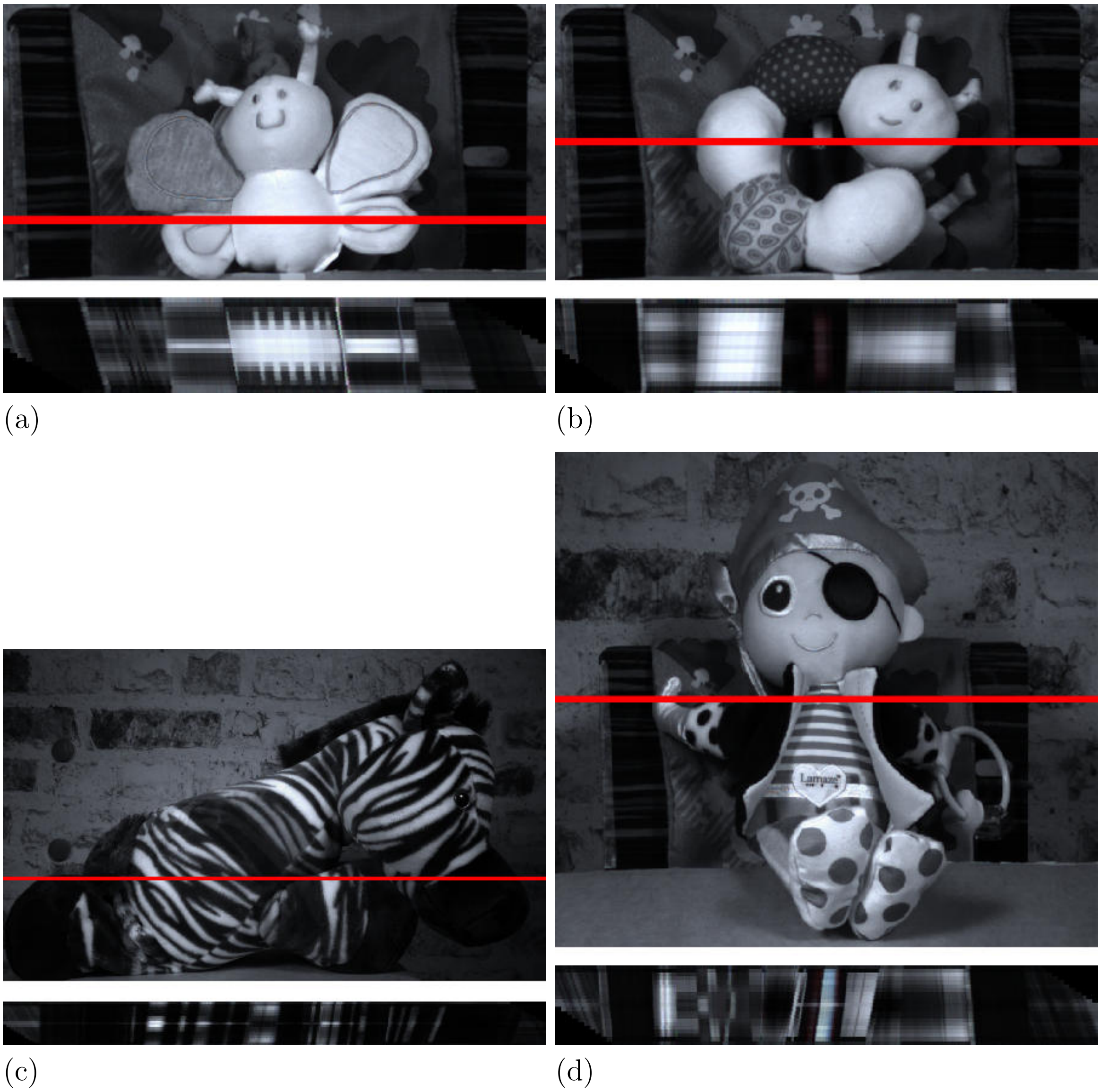


Figure E.5: *(a) shows the central image of the heterogeneous light field given in figure E.3. An EPI slice, related to the red line is shown underneath. The same for the worm light field in (b) which relates to figure E.2.(c) shows the central image of the heterogeneous light field given in figure E.1. An EPI slice, related to the red line is shown underneath. The same for the pirate light field in (d) which relates to figure E.4.*

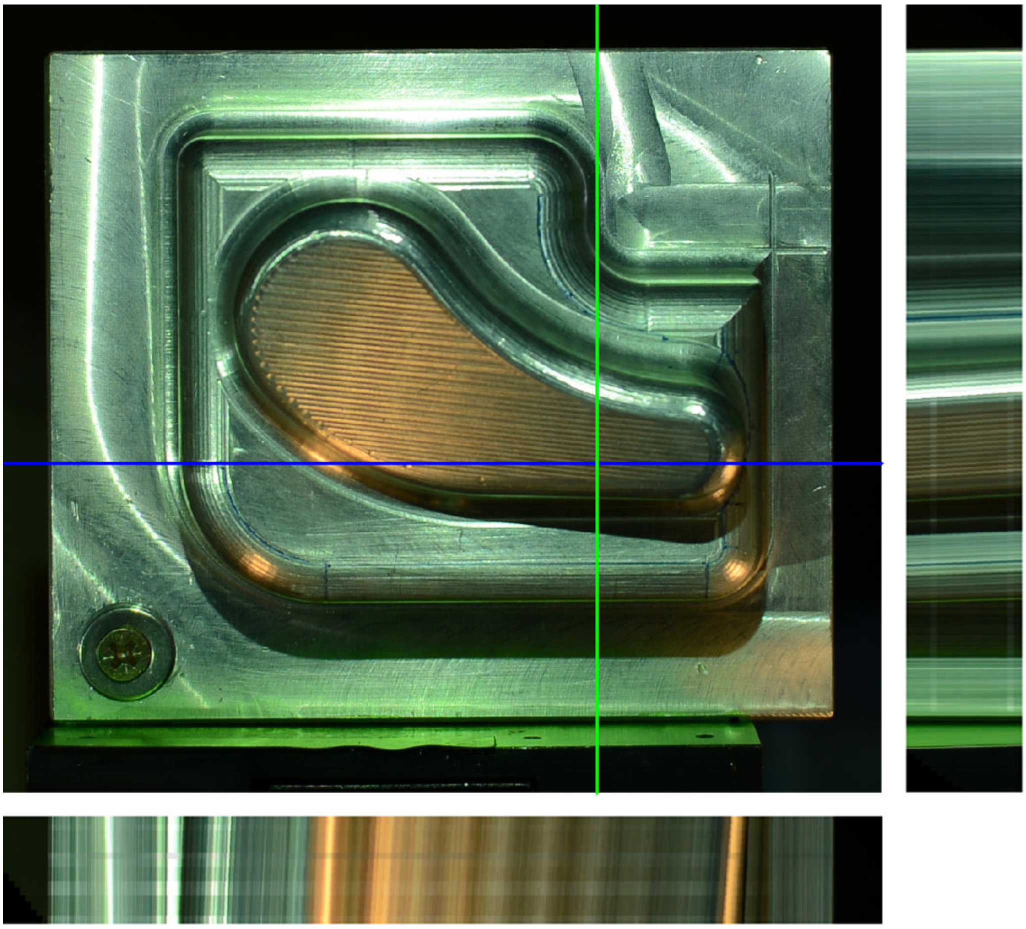


Figure E.6: *shows the center view of a metal test part and a vertical and horizontal EPI slice related to the position of the green and blue line. The EPIs illustrate the influence of illumination flickering.*

List of Publications

Parts of the thesis are extracts from the following publications.

Conference Paper

Epipopar Plane Image Refocusing for Improved Depth Estimation and Occlusion Handling
M. Diebold, B.Goldlücke
Vision, Modelling and Visualization Workshop (VMV), 2013

Spherical Light Fields
B. Krolla, M. Diebold, B. Goldluecke, D. Stricker
British Machine Vision Conference (BMVC), 2014

Light Field from Smartphone-based Dual Video
Bernd Krolla, Maximilian Diebold, Didier Stricker
Light Field for Computer Vision (LF4CV), 2014

Light-field camera design for high-accuracy depth estimation
M. Diebold, O.Blum, M.Gutsche, S.Wanner, C. Garbe, H.Baker, B.Jähne
Proc. SPIE9528, Videometrics, Range Imaging and Applications XIII, 2015

Heterogeneous Light Field
CVPR submission 2016

Journals

Dreidimensionales Körper-Tracking mit Hilfe eines evolutionären Algorithmus
K. Back, P. Hernández Mesa, M. Diebold, F. Puente León
In Technisches Messen, Vol. 80(10):335-342, 2013

Bibliography

[1] T. Aach, I. Stuke, C. Mota, and E. Barth. Estimation of multiple local orientations in image signals. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–553, 2004.

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

[3] M. Aly and J.-Y. Bouguet. Street view goes indoors: Automatic pose estimation from uncalibrated unordered spherical panoramas. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 1–8, 2012.

[4] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google Street View: Capturing the world at street level. *IEEE Computer*, 43(6):32–38, 2010.

[5] A.Pagani and D.Stricker. Structure from motion using full sherical panoramic cameras. In *OMNIVIS2011*, 2011.

[6] Argos 3D-P100 Image. Published by Daniela Hübsch under Creative Commons License (CC BY-SA3.0). `https://de.wikipedia.org/wiki/TOF-Kamera#/media/File:Argos3D-P100_pers_2_W3200x2000.png`.

[7] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. In *In IJCV*, 1989.

[8] S. Baker and T. Kanade. Limits on Super-Resolution and How to Break Them. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.

[9] S. Barry. *Fixing My Gaze: A Scientist's Journey into Seeing in Three Dimensions.* Basic Books; First Trade Paper Edition edition, 2010.

[10] M. Bentsen, G. Evensen, H. Drange, and A. Jenkins. Coordinate transformation on a sphere using conformal mapping. *Monthly Weather Review*, pages 2733–2740, 1999.

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

[12] C. Birklbauer and O. Bimber. Panorama light-field imaging. In *SIGGRAPH Posters*, page 61, 2012.

[13] Blender Foundation. Blender. http://www.blender.org/, 2014.

[14] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, pages 7–55, 1987.

[15] Boxplot Image. Published by Jhguch under Creative Commons License (CC BY-SA 2.5). https://en.wikipedia.org/wiki/Box_plot#/media/File:Boxplot_vs_PDF.svg.

[16] K. Bredies and D. Lorenz. *Mathematische Bildverarbeitung Einführung in Grundlagen und moderne Theorie*. Springer, 2011.

[17] B. H. Bunch and A. Hellemans. *The History of Science and Technology*. Houghton Mifflin Harcourt, first edition edition, 2004.

[18] A. Criminisi, S. 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.

[19] M. Diebold, O. Blum, M. Gutsche, S. Wanner, C. Garbe, H. Baker, and B. Jähne. Light-field camera design for high-accuracy depth estimation. In *Proc. SPIE9528, Videometrics, Range Imaging and Applications XIII*, 2015.

[20] M. Diebold and B. Goldluecke. Epipolar Plane Image Refocusing for Improved Depth Estimation and Occlusion Handling. In *Vision, Modeling and Visualization Workshop VMV*, 2013.

[21] First Look at the Rift. oculus.com, 2015.

[22] A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *Asian Conf. on Computer Vision*, 2010.

[23] T. Georgiev. New results on the plenoptic 2.0 camera. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 1243–1247. IEEE, 2009.

[24] T. Georgiev and A. Lumsdaine. Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, 19:021106, 2010.

[25] T. Georgiev, A. Lumsdaine, and G. Chunev. Using Focused Plenoptic Cameras for Rich Image Capture. *CGA*, 31(1):62–73, 2011.

[26] T. Georgiev, Z. Yu, A. Lumsdaine, and S. Goma. Lytro camera technology: theory, algorithms, performance analysis. In *IS&T/SPIE Electronic Imaging*, volume 8667, pages 1J1–1J10. International Society for Optics and Photonics, 2013.

[27] A. Gershun. The Light Field. *J. Math. and Physics*, 18:51–151, 1936.

[28] B. Goldluecke and D. Cremers. Superresolution Texture Maps for Multiview Reconstruction. In *Proc. International Conference on Computer Vision*, 2009.

[29] J. W. Goodman, editor. *Introduction to fourier optics.* Roberts & Company Publishers, 2005.

[30] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, 1996.

[31] D. Gutierrez, A. Rituerto, J. Montiel, and J. J. Guerrero. Adapting a real-time monocular visual SLAM from conventional to omnidirectional cameras. In *IEEE International Conference on Computer Vision Workshops*, pages 343–350, 2011.

[32] K. Hansung and A. Hilton. 3D Modelling of Static Environments Using Multiple Spherical Stereo. In *Proc. European Conference on Computer Vision*, page 2, 2010.

[33] R. I. Hartley. In Defense of the Eight-Point Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, June 1997.

[34] Y. S. Heo, K. M. Lee, and S. U. Lee. Mutual Information-based Stereo Matching Combined with SIFT Descriptor in Log-chromaticity Color Space. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[35] H. Hirschmller and D. Scharstein. Evaluation of cost functions for stereo matching. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[36] I. P. Howard and B. J. Rogers. *Binocular vision and stereopsis.* Oxford University Press, 1996.

[37] Howard I P. Alhazen's neglected discoveries of visual phenomena. *Perception*, 25(10):1203–1217, 1996.

[38] B. Jähne. *Digital Image Processing.* Springer, 2005.

[39] H. Jin, D. Cremers, D. Wang, A. Yezzi, E. Prados, and S. Soatto. 3-D Reconstruction of Shaded Objects from Multiple Images Under Unknown Illumination. *International Journal of Computer Vision*, 76(3):245–256, March 2008.

[40] H. Jin, S. Soatto, and A. Yezzi. Multi-View Stereo Reconstruction of Dense Shape and Complex Appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005.

[41] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 103–110, 2001.

[42] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura. Viewpoint-dependent stereoscopic display using interpolation of multiviewpoint images. In *Proceedings of SPIE*, volume 2409, page 11, 1995.

[43] 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.

[44] B. Krolla, M. Diebold, B. Goldlücke, and D. Stricker. Spherical Light Fields. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[45] A. Kubota, K. Aizawa, and T. Chen. Reconstructing Dense Light Field From Array of Multifocus Images for Novel View Synthesis. *IEEE Transactions on Image Processing*, 16(1):269–279, 2007.

[46] G. Kurillo, H. Baker, L. Zeyu, and B. Ruzena. Geometric and Color Calibration of Multiview Panoramic Cameras for Life-Size 3D Immersive Video. In *International Conference on 3D Vision*, 2013.

[47] 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*, volume 7893 of *LNCS*, pages 61–73. Springer, 2013. 1.

[48] F. Lenzen, H. Schäfer, and C. Garbe. Denoising Time-Of-Flight Data with Adaptive Total Variation. In *Proceedings ISVC*, pages 337–346. Springer, 2011.

[49] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996.

[50] R. Li, Z. Bing, and L. Ming. A New Three-Step Search Algorithm for Block Motion Estimation. In *IEEE Trans. Circuits And Systems For Video Technology*, 2004.

[51] I. Light and Magic. OpenEXR fileformat. http://www.openexr.com/, 2013.

[52] G. Lippmann. Épreuves réversibles donnant la sensation du relief. In *J. Phys. Theor. Appl.*, pages 821–825, 1908.

[53] LizardQ GbR. LizardQ. http://www.lizardq.com/, 2014.

[54] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[55] A. Lumsdaine and T. Georgiev. Full resolution lightfield rendering. Technical report, 2008.

[56] A. Lumsdaine and T. Georgiev. The Focused Plenoptic Camera. In *In Proc. IEEE International Conference on Computational Photography*, pages 1–8, 2009.

[57] Q.-T. Luong and O. Faugeras. The Fundamental Matrix: Theory, Algorithms, and Stability Analysis. *International Journal of Computer Vision*, 17:43–75, 1996.

[58] Lytro Inc. Lytro. https://store.lytro.com/, 2014.

[59] S. Mann. Compositing multiple pictures of the same scene. In *Proceedings of the 46th Annual IS&T Conference*, volume 2, 1993.

[60] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proc. SIGGRAPH*, pages 39–46, 1995.

[61] Microsoft. Microsoft Streetside. http://www.microsoft.com/maps/streetside.aspx.

[62] M. Mühlich and T. Aach. A Theory of Multiple Orientation Estimation. In H. Bischof and A. Leonardis, editors, *9th European Conference on Computer Vision (ECCV)*, page to appear, Graz, May 7-13 2006. Springer Lecture Notes on Computer Science.

[63] 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.

[64] 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.

[65] R. Paschotta. Optical heterodyne detection. https://www.rp-photonics.com/optical_heterodyne_detection.html.

[66] C. Perwass and L. Wietzke. The Next Generation of Photography, 2010. www.raytrix.de.

[67] PMDs CamCube Image. Published by Magnus Manske under Creative Commons License (CC BY-SA 3.0). https://de.wikipedia.org/wiki/TOF-Kamera#/media/File:PMDvision_CamCube.jpg.

[68] T. Pock, A. Chambolle, H. Bischof, and D. Cremers. A Convex Relaxation Approach for Computing Minimal Partitions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 810–817, 2009.

[69] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An Algorithm for Minimizing the Piecewise Smooth Mumford-Shah Functional. In *Proc. International Conference on Computer Vision*, 2009.

[70] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global Solutions of Variational Models with Convex Regularization. *SIAM Journal on Imaging Sciences*, 2010.

[71] Pointgrey. Ladybug 2. http://www.ptgrey.com/products/ladybug2/ladybug2_360_video_camera.asp, 2014.

[72] N. Qian. *Binocular Disparity and the Perception of Depth.* Neuron, Vol. 18, 359368, 1997.

[73] L. Rayleigh. V. investigations in optics, with special reference to the spectroscope. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9(53):40–55, 1880.

[74] Raytrix GmbH. Raytrix. http://www.raytrix.de/, 2014.

[75] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski. *High dynamic range imaging: Acquisition, display, and image-based lighting: 2nd ed.* Elsevier, 2010.

[76] Ricoh Company. Ricoh theta. http://theta360.com/en/, 2013.

[77] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[78] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Institute of Electrical and Electronics Engineers, 2008.

[79] SceneCam®HDR. https://www.spheron.com/products/visual-asset-management.html, 2014.

[80] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet. http://www.w3.org/Graphics/Color/sRGB.html, 1996.

[81] SwissRanger 4000 Image. Published by Captaindistance under Creative Commons License (CC BY-SA3.0). https://de.wikipedia.org/wiki/TOF-Kamera#/media/File:TOF_Kamera.jpg.

[82] Y. Taguchi, A. Agrawal, A. Veeraraghavan, S. Ramalingam, and R. Raskar. Axialcones: Modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Transactions on Graphics-TOG*, 29(6):172, 2010.

[83] Tamaggo Inc. Ibi 360. http://ibi360.tamaggo.com/, 2014.

[84] Tof Camera Concept. Published by Captaindistance as public domain release. https://commons.wikimedia.org/wiki/File:TOF-Kamera-Prinzip.jpg.

[85] S. Tominaga. Spectral imaging by a multichannel camera. In *Proc. SPIE 3648, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV, 38*, 1998.

[86] A. Torii, A. Imiya, and N. Ohnishi. Two- and three-view geometry for spherical cameras. In *Proceedings of the sixth workshop on omnidirectional vision, camera networks and non-classical cameras.* Citeseer, 2005.

[87] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and Rendering With Incident Light Fields. In *Proc. Eurographics Workshop on Rendering*, pages 141–149, 2003.

[88] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 141–149. Eurographics Association, 2003.

[89] V. Vaish, R. Szeliski, C. Zitnick, S. Kang, and M. Levoy. Reconstruction Occluded Surfaces using Synthetic Apertures: Stereo, Focus and Robust Mea- sures. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2006.

[90] S. Wanner and B. Goldluecke. Spatial and angular variational super-resolution of 4D light fields. In *Proc. European Conference on Computer Vision*, 2012.

[91] S. Wanner and B. Goldluecke. Reconstructing Reflective and Transparent Surfaces from Epipolar Plane Images. In *Proc. German Conference on Pattern Recognition*, 2013.

[92] S. Wanner and B. Goldluecke. Variational Light Field Analysis for Disparity Estimation and Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[93] Civetta 360° Camera. http://www.weiss-ag.org/solutions/civetta/.

[94] 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.

[95] Y. Xu, K. Maeno, H. Nagahara, and R. Taniguchi. Mobile Camera Array Calibration for Light Field Acquisition. In *Proc. International Conference on Computer Vision and Pattern Recognition*, volume 6, 2014.