Fakultät für Physik und Astronomie

Ruprecht-Karls-Universität Heidelberg

Masterarbeit

Im Studiengang Physik

vorgelegt von

(Vor- und Zuname)

geboren in (Geburtsort)

(Jahr der Abgabe)

(Titel)

(der)

(Masterarbeit)

Die Masterarbeit wurde von (Vorname Name)

ausgeführt am

(Institut)

unter der Betreuung von

(Frau/Herrn Prof./Priv.-Doz. Vorname Name)

Department of Physics and Astronomy

University of Heidelberg

Master thesis

in Physics

submitted by

(name and surname)

born in (place of birth)

(year of submission)

(Title)

(of)

(Master thesis)

This Master thesis has been carried out by (Name Surname)

at the

(institute)

under the supervision of

(Frau/Herrn Prof./Priv.-Doz. Name Surname)

(Titel der Masterarbeit - deutsch):

(Abstract in Deutsch, max. 200 Worte. Beispiel: **?**)
Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

(Title of Master thesis - english):

(abstract in english, at most 200 words. Example: **?**)
Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Contents

1 Theory

1.1 Light Field Parametrization

The earliest introduction to Light Fields in literature can be found in Adelson et al. [1991], where they parametrize the field of light as a so-called „plenoptic function “. If we assume that every point in space emits a light ray in a given direction which is characterized by a intensity value P, the whole information in the light field is given as a 6-dimensional function

$$P(V_x, V_y, V_z, \theta, \phi, \lambda), \tag{1.1}$$

where θ and ϕ are the solid angles describing the direction of any light field, λ describes the wavelength dependence. $V_{\{}x, y, z\}$ describe the room coordinates. If we use the pixel of a pinhole camera picture as the coordinate system of our choice, the plenoptic function would be parametrized as

$$P(x, y, V_x, V_y, V_z, \lambda). \tag{1.2}$$

A more generalized model of the plenoptic function could also include a time dependence, leading to a 7-dimensional ray space. In general, the plenoptic function serves as the global funcion that gets mapped into a low-dimensonal space in some form by any camera device, e.g. a pinhole camera mapping the whole ray space down to a 2-dimensional image.

A more detailed introduction to Lightfields can also be found in [15].

1.1.1 The Lumigraph

In the form of 1.2 the plenoptic function is a 7-dimensional function, which is difficult to record and to handle. As described by Wu et al. [2017] this plenoptic function is usually simplified in 3 steps: First, we neglect the time dependence (assuming a static scene), further we neglect the wavelength λ, instead we define the mapped value of the plenoptic function as a vector containing the color channels. Additionally the plenoptic function obtains redundant information due to the fact, that light rays that are lying on one line in space propagate in the same direction, as introduced by Bolles et al. [1987]. The 4D- representation is also known as the *Lumigraph*. Those 4 dimensions can be split to 2 angular dimensions describing the direction of each ray and 2 dimensions for the loaction of the ray: Since every ray would pass an image plane of infinite size once (if the propagation vector is not parallel), two coordinates for localizing the ray are sufficient. Commonly the *two-plane-parametrisation* is used to describe the light field. Every ray is characterized by the intersection of two arbitrary parallel planes Π and Ω. Mathematically spoken the light field L is a function

$$L : \Omega \times \Pi \to R \qquad x, y, s, t \to L(x, y, s, t), \tag{1.3}$$

where x, y are the coordinates in the first plane Ω and s, t are the coordinates in the second plane Π. If we move a camera in a plane and take pictures of a scene orthogonal to the camera plane, the image itself is described by the image coordinates x, y while the position of the camera is denoted as s, t. Both planes are parallel to each other; that way a light field can be measured in a straight forward manner.

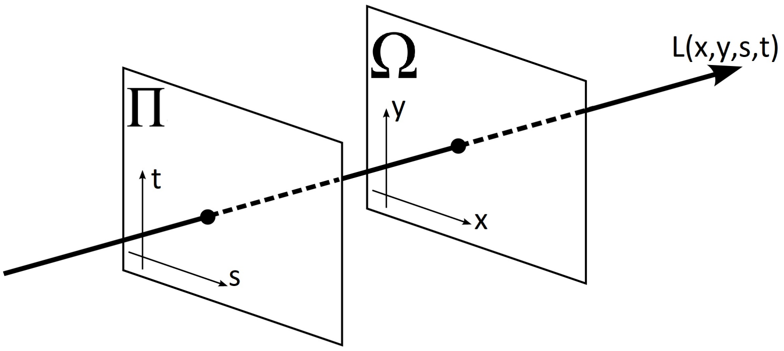

Figure 1.1: In a 4-dimensional two-plane parametrisation a light ray is characterized by the intersection with two parallel planes. we refer to the plane Π as the camera plane and the plane Ω as the image plane. From Xu et al. [2012]

1.1.2 Epipolar Plane Images

We measure the light field in order to obtain as much information about the scene we're looking at as possible. In order to obtain the threedimensional structure one needs to map the light field in a certain manner: We take a look at one 2-dimensional slice through the 4-dimensional space while keeping 2 coordinates constant: one horizontal coordinate x^* in the image plane and one vertical coordinate t^* in the camera plane (or vice versa). The slice Σ_{x^*,t^*} is a 2-dimensonal image called Epipolar Plane Image (EPI). In figure 1.2 the extraction of an EPI from camera array data is visualized. As seen in figure 1.3 it consists of lines with different slopes, each point on one line with the same slope belongs to the same point in the scene und different angles. From the slope Δ of the line at each point we obtain the distance from the camera plane with the equation

$$\text{distance} = \frac{f \cdot b}{\Delta}, \tag{1.4}$$

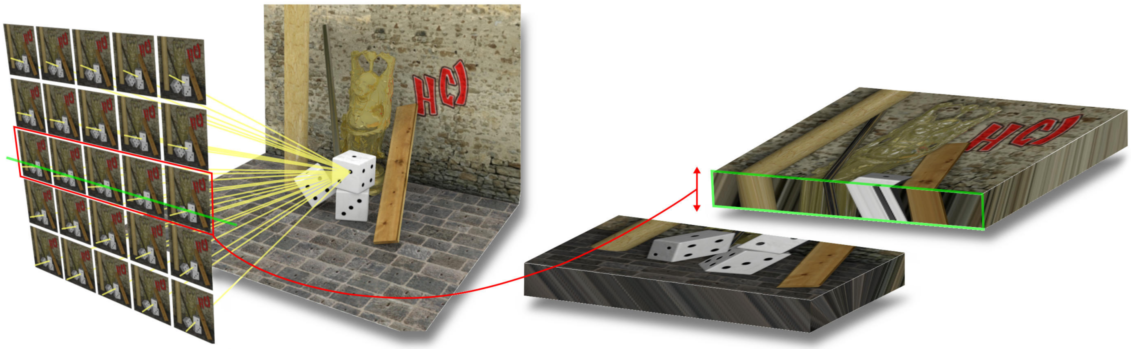

Figure 1.2: Visualization of an Epipolar Plane Image extractoin: A camera array takes images of the same scene from slightly different angles (left array). For a fixed image coordinate y^* (green) and a fixed camera coordinate t^* (red) the pixels are extracted and stacked up resulting in an EPI Σ_{y^*,t^*} (green box on the right). From [2]

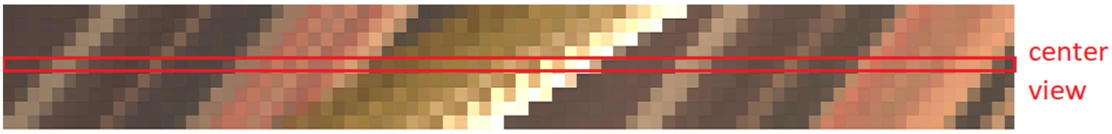

Figure 1.3: An Epipolar Plane Image (EPI) that consists of 9 rows (9 equidistant views or sample points in the camera plane). Points with the same color correspond to the same scene point. Since the viewpoints are slightly shifted, the scene point is also shifted in each view by the disparity d. Marked in red one can identify the center position of the camera viewpoints.

where f is the focal length of the camera and b is the baseline between the camera positions. One may notices that this equation looks equal to distance estimation in stereo vision, where we replace the slope Δ by the disparity shift of a feature in 2 views. In fact, a stereo light field capture would result in an EPI with only 2 pixel rows, the slope of a line consisting of only two points is then defined as the disparity shift between the views.

1.2 Depth from Structure Tensor

Measuring depth from light field data can be done using various approaches. Wu et al. [2017] divide light field depth estimation approaches in three categories:

1. Sub-Aperture Image Matching-based Methods
2. EPI-based methods
3. Learning-based methods

In the following we focus on one specific EPI-based method on which is investigated as part of this work. However, the reader is encouraged to take a look at the cited paper „Light Field Image Processing: An Overview "by Wu et al. [2017] for a state-of-the-art overview of different light field approaches.
Wanner [2014] makes use of the oriented structure of the EPI using image-processing techniques. This idea has first been introduced by Bigun [1987] in 1987.

The 2-dimensional structure tensor J on a function $g : \Omega \rightarrow R, \Omega \subset R^2$ is defined as

$$J = \begin{pmatrix} G * \frac{\partial g}{\partial x}\frac{\partial g}{\partial x} & G * \frac{\partial g}{\partial x}\frac{\partial g}{\partial s} \\ G * \frac{\partial g}{\partial s}\frac{\partial g}{\partial x} & G * \frac{\partial g}{\partial s}\frac{\partial g}{\partial s} \end{pmatrix}, \tag{1.5}$$

where G is a gaussian window function. The derivation can be found in the appendix. The first eigenvector of this tensor J indicates the preferred orientation in the local neighbourhood (defined by the window function). The second eigenvector is orthogonal to the first.

Jähne [2013] proposes a coherence value as a measurement for the anisotropy of the local environment:

$$C = \frac{\sqrt{(J_{11} - J_{22})^2 + 4J_{12}^2}}{J_{11} + J_{22}} \tag{1.6}$$

which obeys the value 1 in case of complete anisotropy, a value of 0 would indicate total isotropy.
In case of an EPI, the structure tensor provides an estimate for the slope of an EPI line, as defined by Bigun [1987]:

$$\Delta = \tan\left(\frac{1}{2}\arctan\left(\frac{J_{22} - J_{11}}{2J_{12}}\right)\right) \tag{1.7}$$

we refer to Δ as the disparity d in the following. Via equation 1.4 one obtains the depth in meters.

1.2.1 Implementation

The following implementation steps have been proposed by Wanner [2014]. The code uses *crosshair* light field data, that can be obtained e.g. by a camera as depicted in figure 1.4. The scope of the light field in the camera coordinates is significantly reduced, in the following the horizontal camera row and the vertical camera column are viewed seperately as 1-dimensional light-field cameras.

From the input images one extracts the EPIs, which have the dimensions

$$\begin{aligned} \text{vertical view} &\rightarrowtail \text{Nr of rows in image coordinates } \times \text{Nr of cameras} \\ \text{horizontal view} &\rightarrowtail \text{Nr of columns in image coordinates } \times \text{Nr of cameras} \end{aligned}$$

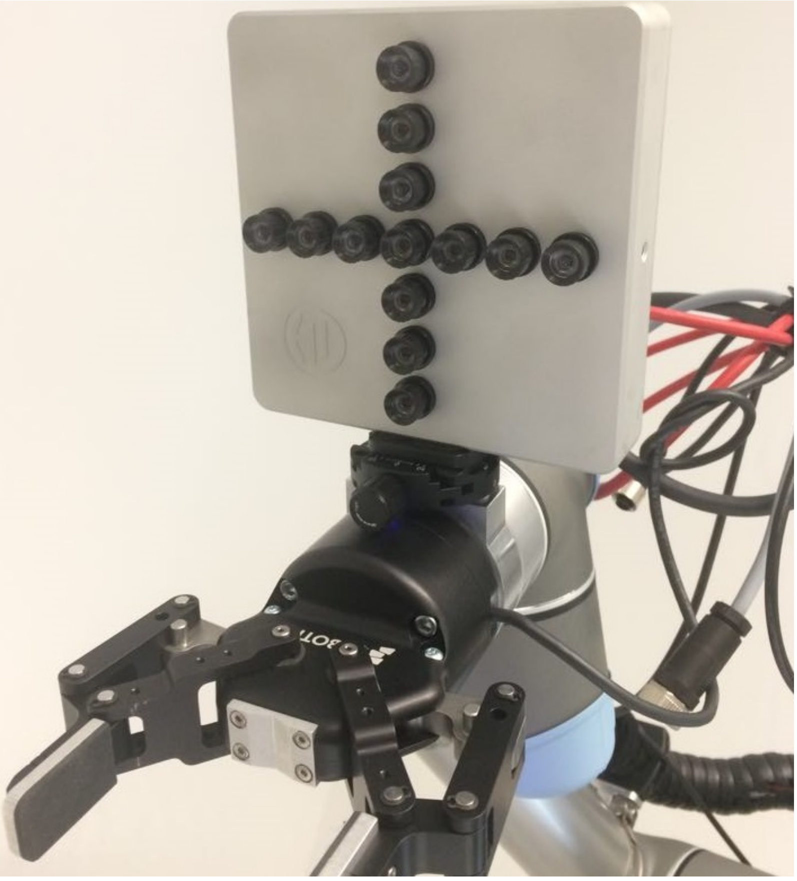

Figure 1.4: Instead of a complete camera array a crosshair camera reduces the number of cameras to one row and one column for faster processing. This Scanner from HDVisionSystems is mounted an a robot arm for industrial applications.

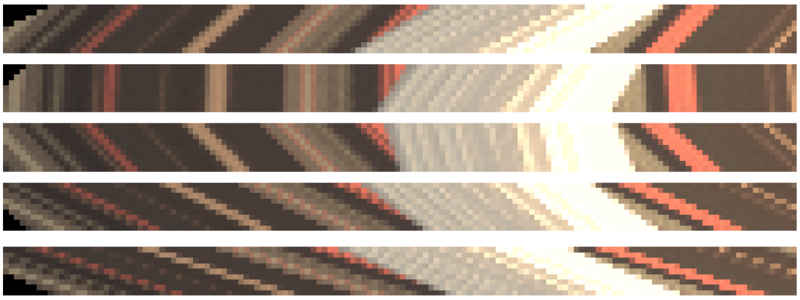

Figure 1.5: The EPI is refocussed by integer disparity steps. If the slope at a scene point is zero (vertical line) the EPI is perfectly focussed on that point. An integration of all views would still result in a sharp image at the corresponding depth.

For each EPI we calculate the structure tensor independently. This is done by first presmoothing the EPI with a 3×3 gaussian kernel to obtain reasonable results when calculating the gradient in the next step. Note that before calculating the gradient, the EPI is converted to grayscale format.
The gradient is calculated via the Scharr-filter, which has the form

$$\text{Scharr}_x = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix} \tag{1.8}$$

$$\text{Scharr}_y = \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}. \tag{1.9}$$

Wanner [2014] and Diebold [2016] both tested out different gradient filters and came to the conclusion that the Scharr-filter performs best. With the gradients the structure tensor and the disparity is obtained via equation 1.5 and 1.7.
Further Wanner found out that the structure tensor results are most accurate when the disparity is close to zero. Therefore the EPI is artificially refocussed by simple linear shifting of the rows such that the slope of any line in the EPI would be increased by the same amount. The disparity shift can simply be subtracted from the disparity later. The refocussing of the EPI is illustrated in 1.5. For each EPI the structure tensor is calculated at each disparity shift, so that the necessary disparity range is fully covered.

From all disparity shifts the value with the highest coherence measure (equation 1.6) is selected. Hence one obtains two depth estimations for each image coordinate in the center view: one from the vertical EPI, one from the horizontal EPI. The merging of both direction follows

$$d(x,y) = \begin{cases} d_{\text{horizontal}}(x,y) & C_{\text{horizontal}}(x,y) > C_{\text{vertical}}(x,y) \\ d_{\text{vertical}}(x,y) & C_{\text{horizontal}}(x,y) < C_{\text{vertical}}(x,y) \end{cases}. \tag{1.10}$$

1.2.2 The occlusion problem of the structure Tensor

Having a look at the results of the the benchmark test of Honauer et al. [2016] one realizes that most Light field depth estimation algorithms suffer from large errors near depth discontinuities. Since the center view pixels close to the edge of a depth discontinuity are at least partly occluded, this behaviour is to be expected. The ST almost always produces a systematic error near discontinuities, leading to a „magnification“of the object closer to the camera in the depth map, see figure 1.7. We refer to this as „edge fattening “. The reason for this error has its origin in the smoothing of the EPI as part of the algorithm. At the occlusion, the edge between the fore- and background follows the same orientation as the foreground does.

Furthermore the occlusion mostly comes with a a color change, resulting in a high gradient which dominates the local environment orientation: Edge fattening is the consequence. In figure 1.6 one can recognize two effects leading to edge fattening: Firstly the EPI is smoothed with a 3×3 gaussian kernel which already will enlarge the foreground structure (b). This becomes clear when we illustrate the norm of the gradients (c), since the strong (white) gradients all measure the forground (green) structure. A clear shift to the left is visible from (b) to (c). However, this bias error is small compared to the second error that is indicated in (d): The red dot lies clearly in the background structure. However, calculating the structure tensor components from the local environment (green square) one would obtain the foreground structure at the red dot, since the strong gradients in the square dominate. In (e) the total bias in labeling the two structures is illustrated.

1.3 Modifications on the structure tensor

In the following diffent methods will be explained which modify the Structure tensor algorithm proposed by Wanner [2014]. In section 2 implementations of those modifications will be evaluated and discussed.

1.3.1 Kernelsizes

The Structure tensor algorithm calculates the depth based on the preffered orientation of the EPI in a local neighbourhood. If one chooses a bigger neighbourhood, the resulting depth map is likely to be smoother, however the accuracy at edges in the scene will rapidly decrease.

Even though one often wants to smoothen the depth map in a postprocessing step to improve the overall result and cancel out outlyiers, the structure tensor algorithm by Wanner is forced to smoothen the depth map beforehand.

In the work of Wanner [2014] the effect of the kernel size has been examined extensively and he calculated the best parameters for different scenes by grid search.

Note that if the kernel is smaller then the height of the EPI, the outer cameras are neglected and do not enter in the calculation of the depth. To maintain the information of all cameras and vary the kernel size at the same time, Diebold [2016] proposes

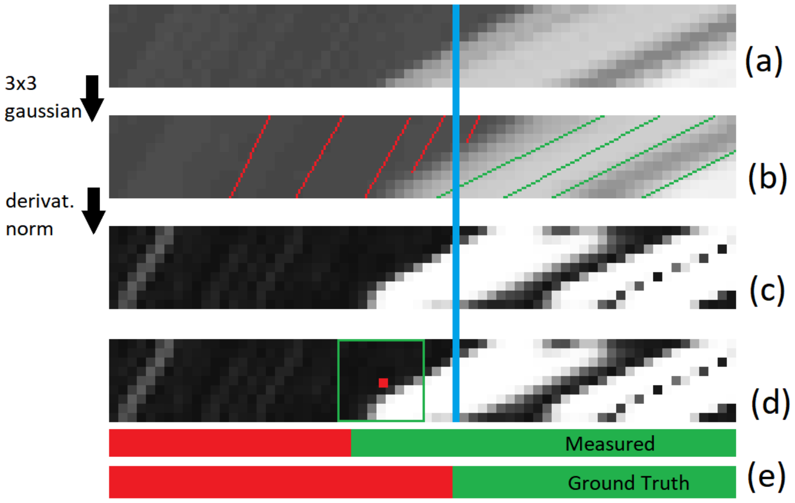

Figure 1.6: (a) The structure of an occlusion border in a gray-value EPI. (b) Before calculating the gradient, the EPI is smoothed with a 3 × 3 gaussian kernel. One sees the smoothed EPI with colored lines indicating the Ground Truth orientation. (c) shows the norm of the gradient calculated via the Scharr-filter. White significates a high gradient, black significates low gradient. In (d) the local environment around the red dot (x_r, y_r) as an example point is marked to show which gradient values go into the structure tensor components $J(x_r, y_r)$. (e) marks the orientation labeling measured and ground truth. The blue line marks the transition in the center view that corresponds with the ground thruth labeling.

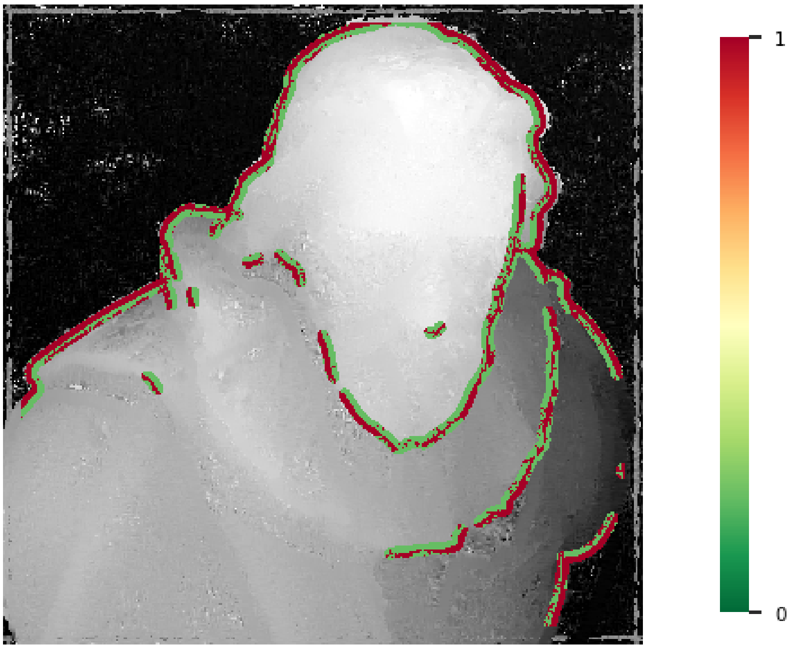

Figure 1.7: Evaluation of the deviation from Ground truth at the depth discontinuity for scene "cotton". The red border indicates that the depth map is errorneous at the outside of the edge.

asymmetrical kernelsizes to eventually reduce edge-fattening effects. However, assymetrical kernels still need to find a trade off between denoising and avoiding edge fattening and do not significantly increase the quality of the depth map result.
Another approach is to choose a custom kernel form which only weights the local gradients that could possibly be part of the „correct"line in the EPI. Such a kernel is illustrated in figure (FIGURE), it has the form of as sandclock. in y-direction, it still follows a gaussian distribution.

1.3.2 Color-awareness with bilateral filtering

In its original form the structure tensor pipeline weights the square of all gradients in a local environment by the distance to the reference point. Since all local gradients in the EPI that belong to the same color in the EPI are very likely to belong to the same orientation, in theory one only has to find all points with the same color which then lie on the same line in the EPI. The preferred orientation of the gradients at those points will then correspond to their slope with high confidence. However, most objects are not perfectly lambertian such that not all points on the correct line have to be of the exact same color. Only filtering the exact color will lead to noisy results. Though one could implement a filter weighting not only the distance in a gaussian manner but also the distance from the EPI kernel origin in the color RGB space; such a filter is called „bilateral filter "(Tomasi and Manduchi [1998]). Pixels with completely different color will be excluded from the Neighbourhood when calculating the structure tensor. In equation 1.5 the window function G is then defined as

$$G_P = \frac{1}{W_p} \sum_{q \in N} g_{\sigma_d}(||p - q||) g_{\sigma_c}(||I_{p,\mathrm{EPI}} - I_{q,\mathrm{EPI}}||), \tag{1.11}$$

where

p is the position of the pixel,

q is the position of the pixel in a neighbourhood N,

g_{σ_d} is a gaussian function weighting the distance between p and q using σ_d as standard deviation,

g_{σ_c} is a gaussian function weighting the color distance in RGB space between the RGB values of the original EPI at p and q using σ_c as standard deviation,

W_P is a normalization factor so that the sum over all the neighbourhood always is 1.

It is important to notice that the color distance is extracted from the EPI, though we are filtering the components of the second derivatives, see equation 1.5. In fact, the EPI only serves as a „guide“for the filtering.

1.3.3 Thresholding the gradients

In section 1.2.2 it is explained why the structure tensor struggles to serve with a good estimation at depth transitions. It is referred to two errors, one coming from the first smoothing, the other one as a result from the second gaussian smoothing. To avoid the second error, one could in principle normalize all local gradients such that in figure 1.6 (d) the foreground gradients in a local environment obey the same weight in the structure tensor calculation as the background gradients around the red dot.
The gradient components of the EPI would be given by

$$\tilde{\vec{\nabla}}(x,y) = \frac{1}{\sqrt{\nabla_x(x,y)^2 + \nabla_y(x,y)^2}} \vec{\nabla}_x(x,y) \tag{1.12}$$

Nevertheless in some occasions one wants the stronger gradients to play a bigger role in the preferred orientation, since small gradients are more likely to be noisy. The only way to damp the gradients at occlusions while not strengthening the small noisy gradients in an EPI is to implement a trunctation threshold. The gradient is then given by

$$\tilde{\vec{\nabla}}(x,y) = \begin{cases} \frac{\text{threshold}}{\text{norm}}\vec{\nabla}(x,y) & \text{if} \quad \text{norm} > \text{threshold} \\ \tilde{\vec{\nabla}}(x,y) & \text{else} \end{cases} \tag{1.13}$$

$$\text{with} \quad \text{norm} = \sqrt{\nabla_x(x,y)^2 + \nabla_y(x,y)^2}. \tag{1.14}$$

$$\tag{1.15}$$

The threshold should be chosen the lowest possible in a manner that gradients higher than the threshold are barely affected by noise. In section **??** different thresholds are tested, if not mentioned otherwise a threshold of 0.1 is used assuming that the underlying EPI is a gray-value image with range 0-1.

1.3.4 Occlusion-awareness using segmentation of the EPI

In the section a method is proposed that attempts to handle both occlusion errors mentioned in section 1.2.2.
Bilateral filtering (section 1.3.2) and thresholding the gradients (section 1.3.3) may improve the depth map estimation, however they come with trade-offs as one can see in the evaluation. See section **??** for more details. An alternative way to maintain sharp transitions is to find those transitions in the EPI and calculate them seperately. By extracting them from the EPI, background and foreground structure are calculated without being biased by the transition itself. Therefor the EPI is segmented into occlusion transistions and areas, whereby the norm vector is thresholded to create a binary mask on the EPI. Possibly one could think of a better criterion that is segmenting occlusion/non-occlusion areas, however this approach is fastly implemented and is sufficient for our needs. In principle the algorithm works as follows:

1. Segmentate the transitions and the rest of the EPI with a binary mask.
2. Calculate the structure tensor components on the masked gradient of the EPI. All masked gradients are zero and do not affect the local environment structure.
3. Calculate the structure tensor components again, now on the inverted masked gradient of the EPI.

In figure 1.8 the segmentation process is illustrated step-by-step. The gradient of the gray-valued EPI (a) is calculated in x- and y-direction (b), from which one calculates and thresholds the norm (c, d). The threshold itself is chosen to be 0.5 if not mentioned otherwise. It is essential to modify resulting mask from the binary thresholding (e) using morphological image-processing operators as explained in detail in hom. It is made use of Morphological Closing (structured filling in of image region boundary pixels), which is explained in figure 1.9.

Having a look at the different scenarios in the given EPI that can happen when segmenting, the necessarity for morphological closing becomes clear. The EPI has two visible disparity transitions, one at Pixel [133] and one at Pixel [185]. At [133] (a) it is seen that the transition affects multiple neighbouring pixels even though the EPI isn't smoothed yet. Our aim is to mask those transtions. However at Pixel [63] and[174] the threshold is also crossed, since big gradients do not necessarily have to lie on a transition. Still the false-positive masked transitions are less uncomfortable to handle then a false negative result. It is clearly visible at [63], that zero crossings in the gradient lead to holes in the mask which need to be closed by morphological closing. Since the local environment of each pixel is only given by near pixels in the same segment (masked/unmasked), small holes lead to errorneous results at those pixels. After morphological closing, a last postprocessing step has to be made to handle the smearing due to the 3×3-gaussian presmoothing. In figure 1.9 (d) one sees that the transition peak is gaussian-formed. If one simply cuts out the segment at the threshold transition, the background segment gradient is still dominated by the gradients directly next to the segment transtions. This becomes visible in figure 1.10, where the inverted mask of the EPI gradient is shown. Thus morphological dilation with a 3×3 kernel is applied on the inverted filter, resulting in a complete seperation of any peak crossing the threshold.

1.3.5 An alternative to the coherence as the confidence measure

The coherence describes, how strong the anisotropy of a structure in an image is. In the structure tensor pipeline (section 1.4) this value is used as a confidence meausure for the corresponding depth value. In this section we try to define a more profound definition for the confidence based on the distribution of the gradients in a small environment. From the structure tensor components we obtain the disparity

$$d = \tan\left(\frac{1}{2}\arctan\left(\frac{J_{22} - J_{11}}{2J_{12}}\right)\right), \tag{1.16}$$

whose error Δd is then given by

$$\Delta d = \left(\left(\frac{\partial d}{J_{11}}\Delta J_{11}\right)^2 + \left(\frac{\partial d}{J_{22}}\Delta J_{22}\right)^2 + \left(\frac{\partial d}{J_{12}}\Delta J_{12}\right)^2\right) \tag{1.17}$$

When substituting

$$d = \tan\left(\frac{1}{2}\arctan(x)\right), \quad \text{with} \quad x = \frac{J_{22} - J_{11}}{2J_{12}} \tag{1.18}$$

we obtain

$$\frac{\partial d}{\partial x} = 0.5 \cdot \frac{d^2 + 1}{x^2 + 1} \tag{1.19}$$

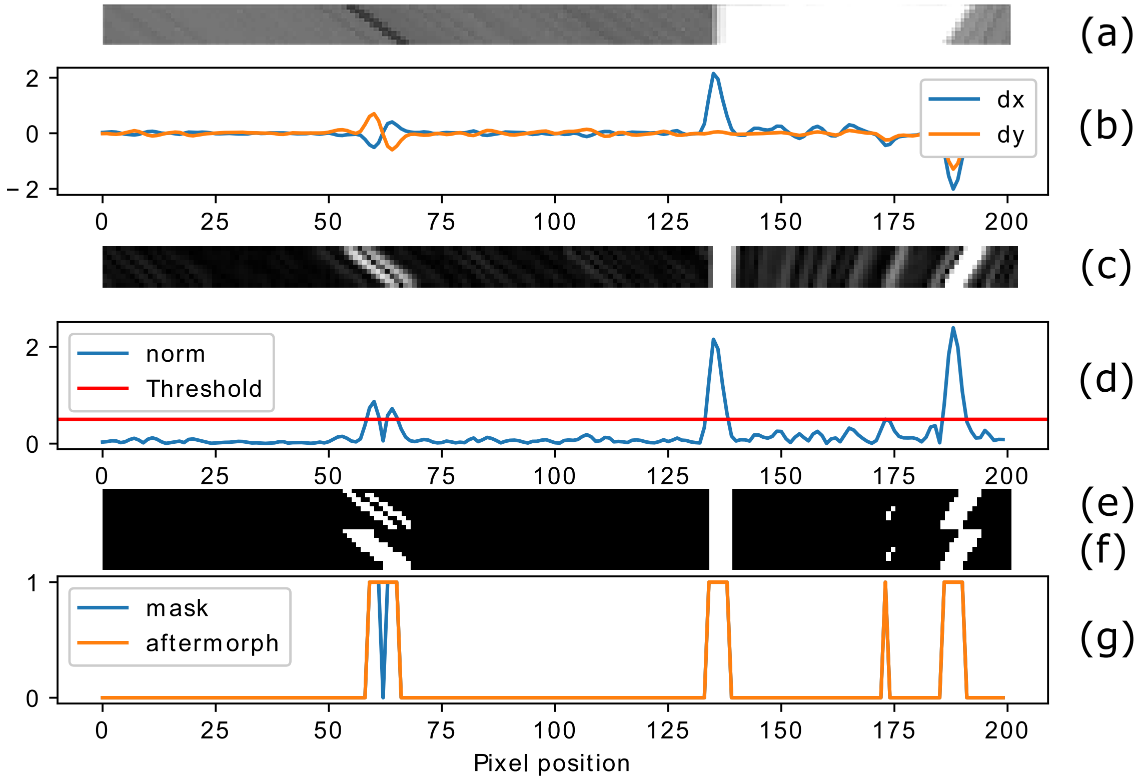

Figure 1.8: (a) Input EPI to be segmented. (b) show the local derivatives along the center view line. (c) shows the vector norm of the derivative. (d) shows the center view norm with the threshold that is applyed to mask transitions. (e) shows the resulting mask. (f) shows the improved mask using morphological image processing. (g) shows the mask at the center view before and after using morph. image processing.

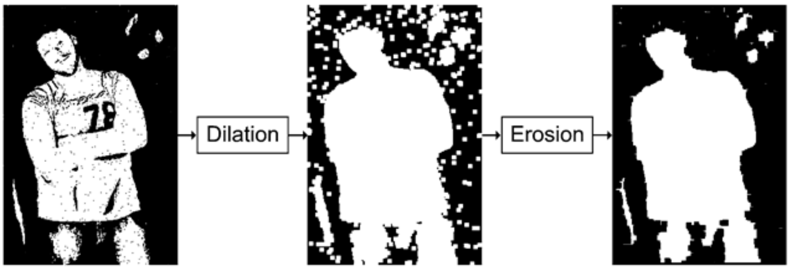

Figure 1.9: Morphological closing is a combination of Dilation and Erosion. Dilation uses a custom-sized kernel and turnes any 0 to 1, if at least one 1 isfound in the local environment. Erosion turnes any 1 to 0, if at least one 0 is found in the local environment. Image from wha

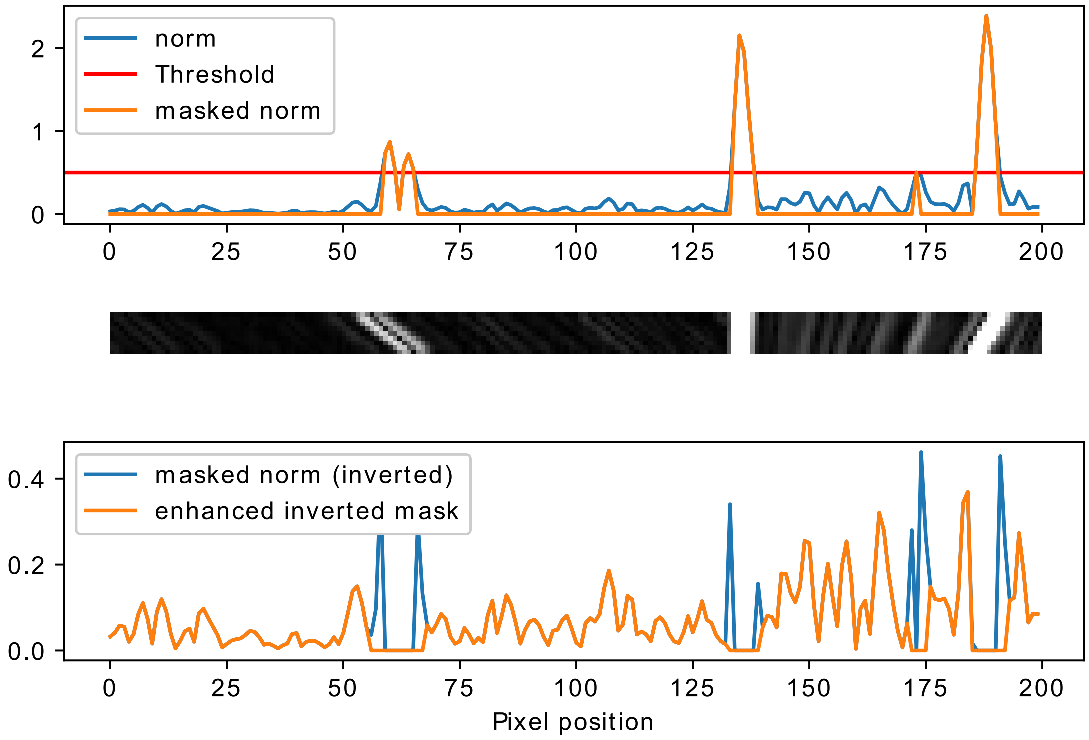

Figure 1.10: In the upper diagram the thresholded gradient norm of the EPI (middle) is seen. The norm (masked with the inverted mask) is depicted in the lower figure (blue), while the inverted mask with a dilation filter is depicted in orange.

1.4 Depth from focus

One advantage of using lightfields for depth measure is its ability to get a two-dimensional mapping of the scene with focus at any depth. Integrating the views of the light field camera array has the same effect as the integration of a focussed lense camera, as the lense is simply integrating slightly different viewpoints of the same scene point when focussed on the correct depth.
Obtaining the refocussed integrated image is a synthetic process that only requires shifting the view coordinates artificially. Given a full four-dimensional light field $L(u, v, x, y)$ we can refocus the light field as described in [12] :

$$L'(u, v, x, y) = L(u(1 - d'), v(1 - d'), x, y), \tag{1.20}$$

where d' describes the relative pixel shift. The disparity is directly related to the absolute depth of the focus (relate to PICTURE) if the relevant camera parameters are known. Given the baseline b in meters and the focal length f in pixels, the depth Z is given as

$$Z = \frac{f \cdot b}{d}. \tag{1.21}$$

We obtain

$$\bar{L}(x, y) = \frac{1}{N_{u,v}} \int \int L'(u, v, x, y) du dv = \frac{1}{N_{u,v}} \sum_u \sum_v L'(u, v, x, y) \tag{1.22}$$

Once we can focus at any range, one can adopt *depth-from-focus*-techniques as described in Watanabe and Nayar [1998] for depth measure. If the scene point at a given image coordinate (x, y) in the center view is in focus, the contrast in the integrated image $\bar{L}(x, y)$ is high, thus a contrast measure at each pixel combined with stepwise refocussing yields a depth map.
For measuring the contrast, one has different options: The most straight forward approach is calculating the first derivative of the grey-value image. At high contrast structure the local intensity changes are expected to be high. Alternatively one could measure the second derivative laplacian that eventually results in higher robustness. The implementation and tests of those techniques for the benchmark dataset can be found in section **??**.
Using a pinhole camera array allows us to go further and find a response value that shows higher consistency. Taking the absolute difference between the center view of the camera array and the refocussed image yields to promising results as shown in Tao et al. [2017]. Under the assumption of lambertian surfaces the RGB- value of any scene point should be the same under all angles. Thus when refocussed on the correct depth, summing over all angles should result in a value that ideally is the same as in the center view alone. This is referred as *photo consistency*; for more information read Tao et al. [2017]. The response value at a given depth is obtained

from

$$D'(x,y) = \frac{1}{|W_D|} \sum_{x',y' \in W_D} \left|\bar{L}(x',y') - P(x',y')\right|, \tag{1.23}$$

where $P(x,y)$ is the center view. For more robustness, it is averaged over a small window. We refer to this measuring technique as *photo consistency* in the following. Note that calculating the absolute results in a 1-channel-image while the input images are RGB-images.
Tao et al. propose another measure that they refer to as *angular correspondence*. It follows the same principle, but instead of integrating the refocussed lightfield followed by comparing it to the center view, they directly take the difference of each viewpoint to the center view and sum up those differences:

$$D'(x,y) = \frac{1}{N_{u,v}} \sum_u \sum_v |L'(u,v,x,y) - P(x,y)|. \tag{1.24}$$

We tested those methods against the common contrast measures mentioned above, the results are found in section results.

1.5 Semi-Global Matching

1.5.1 Semi - Global Matching for Stereo Vision

In contrast to Light field depth estimation techniques Stereo systems often suffer from mismatching pixels between the left and right images. Many attemps have been made to smoothen bad pixels, resulting in blurred edges or long calculation times. One promising attempt to imporove matching results was published in 2005 by Heiko Hirschmüller (Hirschmuller [2005]) that was described as „a very good trade off between runtime and accuracy “ (Hirschmüller [2011]): we speak of Semi-Global Matching.
In general, matching of two stereo images means shifting the disparity over the predefined disparity range and comparing both images (pixel- or blockwise) until we have a cost value at each image point for each discrete disparity. We assign to each pixel $\vec{p}$ the disparity value $D_{\vec{p}}$ which is related to the lowest cost $C(\vec{p}, D_{\vec{p}})$. This matching does not have to be unique, resulting in errorneous pixel disparities. To overcome this one wants to minimize a global cost function of the form

$$E(D) = \sum_{\vec{p}} \left(C(\vec{p}, D_{\vec{p}}) + \sum_{q \in N_p} \begin{cases} P1 & \text{if } |D_{\vec{p}} - D_{\vec{q}}| = 1 \\ P2 & \text{if } |D_{\vec{p}} - D_{\vec{q}}| \geq 1 \\ 0 & \text{else} \end{cases} \right). \tag{1.25}$$

The fist term sums all matching costs over the whole image, while the second term forces continuity by comparing the disparity of all neighbour pixels N_q to the disparity D_p; if a small discontinuity is detected $(D_{\vec{p}} - D_{\vec{q}} = 1)$, a small penalty is added

to the global cost function. Since a small discontinuity can be found essentially at any tilted plane, only a small error is added. A bigger disparity difference indices a clear discontinuity in the disparity map. Note that the penalty $P2$ can be divided by the gradient of the original image to allow a disparity discontinuity when we find edges in the image; at these points we expect the disparity to be discontinuous.
However, minimizing the global cost function involves computational cumbersome algorithms as it is a NP-complete Problem (Hirschmüller [2011]). Semi-Global Matching however chooses another approach by minimizing the global cost function along one-dimensional lines – this can indeed be calculated in polynomial time. The new smoothed cost function $S(\vec{p}, D_{\vec{p}})$ at pixel $\vec{p}$ is then given as the sum of all 1D minimum cost paths that are ending in $\vec{p}$. The minimal cost L'_r along the path r is defined recursively as

$$L'_r(\vec{p}, D) = C(\vec{p}, D) + \min \begin{cases} L'_r(\vec{p_{\text{before}}}, D) \\ L'_r(\vec{p_{\text{before}}}, D+1) + P1 \\ L'_r(\vec{p_{\text{before}}}, D-1) + P1 \\ \min_i L'_r(\vec{p_{\text{before}}}, i) + P2 \end{cases} \tag{1.26}$$

By always adding the minimum path cost of the previous pixel on the scanline we are looking at, we solve equation 1.24 in one dimension. It is to mention that the rolling sum can reach quite high numbers that are unpleasant to handle on the computer; a normalization is implemented by substracting $\min_D L'_r(\vec{p_{\text{before}}}, D)$ from all pixel cost values $L'_r(\vec{p}, D)$. The position of the minimum cost function at pixel $\vec{p}$ is unaffected by that normalization.
Summing along at least 8 path directions (crosshair + diagonals) results in disparity maps with reduced error pixel while maintaining clean edges. Neither a blur filter, a median filter or a bilateral filter would preserve those features.

1.5.2 Semi - Global Matching for Light fields

Even though Hirschmüller describes Semi - Global Matching (SGM) as a complete algorithm to obtain a disparity map from a stereo image input, we further refer to SGM as the true novelty of his work: the implementation of an approximation to the global solution of the cost function (equation 1.24). Indepent from the method one uses to calculate a disparity map, one needs a cost function defined in disparity space for each pixel to make use of SGM. Similar to the Stereo Matching depth estimation, the structure tensor depth estimation pipeline for Lightfield data sets produces a disparity map and a coherence value at each disparity shift. This implies, that the SGM algorithm can be adapted to improve the results of the structure tensor pipeline. However, there are some significant differences between those two methods:

1. The structure tensor algorithm is tuned to a much smaller disparity range. While in Hirschmuller [2005] Hirschmüller scans a disparity range of 32 pixels , The benchmark data sets for light fields mostly include close-up views of

objects, with a disparity range between 2 and 10 pixels. In figure 1.11 one can see the different values that are allocated in memory for each pixel of the image.

2. The subpixel accuracy using the structure tensor is a lot higher than the stereo matching subpixel accuracy. A simple adaption of the algorithm to the structure tensor pipeline would require to give up the best feature that is provided by the ST, its subpixel accuracy.

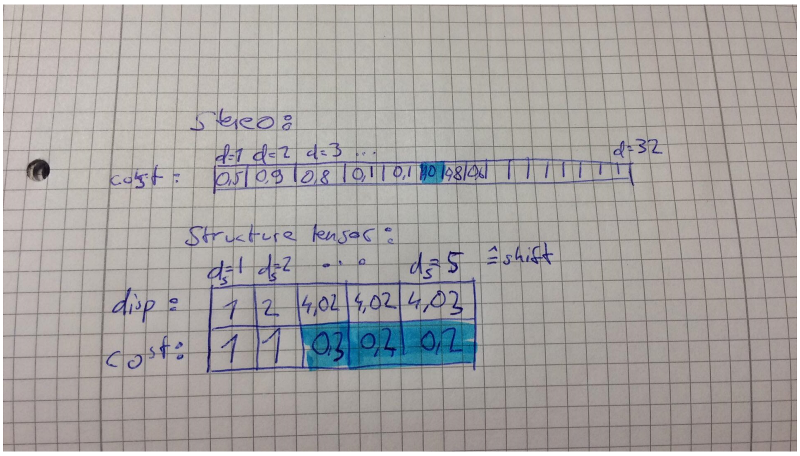

Figure 1.11: One point $\vec{p}$ contains different values (values here: example values): For Stereo matching, the resolution is given by the discrete disparity steps. Each disparity value has a cost value assigned to it. Using the ST, we have a different subpixel accuracy for every disparity shift, while the subpixel accuracy can differ from the shift by up to 1.2

To handle those problems, we do not throw away the subpixel accuracy: instead we use the float-value disparities to decide whether we penalize a disparity discontinuity or not. As one can see in figure 1.11, we have to process an additional information, since the exact disparity value is not implicitely given by the index of the allocated cost value (in contrast to the original algorithm). Switching to a continuous space as depicted in figure 1.12 requires a new definition of the error propagation defined in equation 1.24.

In the following we refer to s as the disparity shift in the ST algorithm. Note that we replaced D by d to clarify that the disparity is no longer discrete:

$$E(d) = \sum_{\vec{p}} \left(C(\vec{p}, d_{\vec{p}}) + \sum_{q \in N_p} \begin{cases} P1 \cdot |d_{\vec{p}} - d_{\vec{q}}| & \text{if } |d_{\vec{p}} - d_{\vec{q}}| \leq 1 \\ P2 & \text{if } |d_{\vec{p}} - d_{\vec{q}}| > 1 \end{cases} \right). \tag{1.27}$$

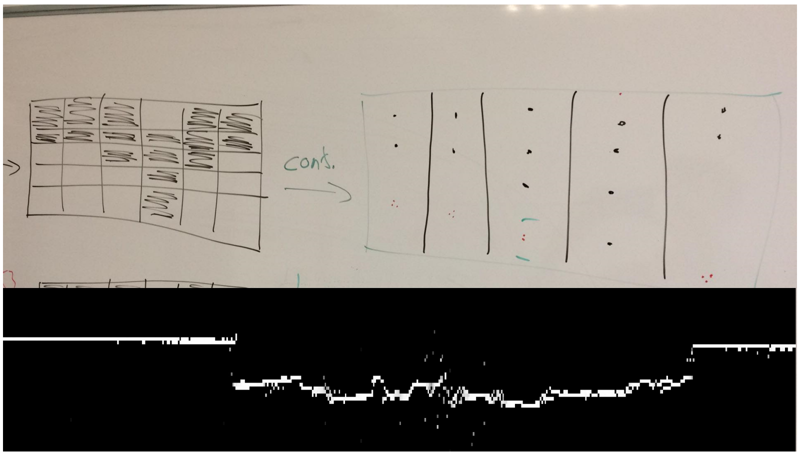

Figure 1.12: In this figure one can see the skizze of one arbitrary scanline of the structure tensor method. Under the Assumption that the ST algorithm recognizes the structure of the EPI perfectly, we either have two or three disparity shifts that have a high coherence (colored white) (a) and result in approximately the same final disparity. This can be seen if we plot the exact disparity values in a continuous space(b). In (c) real data scanline cost is plotted with a resolution of ca. 100 pixels.

The recursive 1-d form to solve the global constraint on a scanline then changes to:

$$L'_r(\vec{p}, s) = C(\vec{p}, s) + \min_i \begin{cases} L'_r(\vec{p}_{\text{before}}, s_i) + P1 \cdot |d_{\vec{p},s} - d_{\vec{p}_{\text{before}},s_i}| & \text{if } |d_{\vec{p},s} - d_{\vec{p}_{\text{before}},s_i}| \leq 1 \\ L'_r(\vec{p}_{\text{before}}, s_i) + P2 & \text{if } |d_{\vec{p},s} - d_{\vec{p}_{\text{before}},s_i}| > 1 \end{cases} \tag{1.28}$$

The biggest difference lies in the fact that the small factor that is smoothing the image linearely increases with the distance. This change is necessary under the assumption that the disparity space is continuous. In other words we cluster disparity differences between two neighbouring points as either part of one surface ($|d_{\vec{p},s} - d_{\vec{p}_{\text{before}},s_i}| \leq 1$) that gets smoothed by the linearly increasing penalty, or assume a real disparity discontinuity that is penalized regardless of the size of the jump - the second error remains constant as it is in Stereo matching. Note that in our implementation, $P2$ is modified by

$$P2' = \frac{P2}{\sqrt{(Im_b^2 + Im_r^2 + Im_g^2)}}, \tag{1.29}$$

with Im being the center view of the lightfield and $Im_{b,g,r}$ being the 3 color channels. If the color intensity changes, the penalty for a disparity discontinuity is lowered.

1.5.3 Occlusion awareness in SGM for light Fields

If we take a close look at figure 1.12, one can see that at least at some discontinuities the ST pipeline manages to calculate the depth of the background structure near boundaries with good coherence, but the foreground structure is overlapping and quantitatively measured with higher coherence. Once we know that at least at some edges an improvement can be made by adapting the evaluation function in a sense that the highest coherence does not necessarily measure the right depth, we realize that SGM is doing the job already. The simple heuristic approach is to change the global minimization function 1.26 such that a positive disparity jump is less punished than a negative one. In fact, the function changes to

$$E(d) = \sum_{\vec{p}} \left(C(\vec{p}, d_{\vec{p}}) + \sum_{q \in N_p} \begin{cases} P1 \cdot |d_{\vec{p}} - d_{\vec{q}}| & \text{if } |d_{\vec{p}} - d_{\vec{q}}| \leq 1 \\ P2 & \text{if } d_{\vec{p}} - d_{\vec{q}} > 1 \\ P3 & \text{if } d_{\vec{p}} - d_{\vec{q}} < -1 \end{cases} \right). \tag{1.30}$$

1.5.4 SGM as postprocessing smoothing

2 Evaluation

2.1 Depth from focus

The depth measure using epipolar plane analysis requires iterative calculation of the structure tensor for each EPI at each disparity. A way to overcome this is to generate a preestimate of the depth before actually calculating the correct depth. This could also help to prevent possible errors due to periodic scene characteristics which can lead to mismatch errors when calculating the structure tensor. Therefore the depth pre-estimate should fulfil the following criteria:

1. It should be *consistent*, meaning that the number of pixels with low confidence should be the lowest possible.
2. It should result in a *fast* measure, ideally faster then it would take to do the full iterative structure tensor algorithm.
3. It does not have to be subpixel accurate, since it only serves as a pre-estimate.

The methods that are tested are described in section 1.4. We test four different ways to obtain a depth map using depth from focus:

Photo consistency This measure takes advantage of the fact that the difference between the refocussed two-dimensional image and the center view is close to zero when refocussed to the correct depth. Response value:

$$D'(x,y) = \frac{1}{|W_D|} \sum_{x',y' \in W_D} \left| \bar{L}(x',y') - P(x',y') \right|, \tag{2.1}$$

Angular correspondence In contrast to the *Photo consistency* - measure, it first calculates the absolute difference between each camera array view and the center view followed by the summation of those deviations. The response value is given as in equation (1.23)

$$D'(x,y) = \frac{1}{N_{u,v}} \sum_{u} \sum_{v} |L'(u,v,x,y) - P(x,y)| \tag{2.2}$$

First derivative The first derivative is calculated for contrast measure by applying the sobel filter onto the refocussed image I:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \cdot I \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot I \tag{2.3}$$

The directional gradients are simply added up to the response value

$$D'(x, y) = |G_x(x, y)| + |G_y(x, y)| \tag{2.4}$$

Laplace Here we calculate the second derivative laplacian by appling the sobel operator twice:

$$D'(x, y) = \text{Laplace}(I)(x, y) = \frac{\partial^2 I}{\partial x^2}(x, y) + \frac{\partial^2 I}{\partial y^2}(x, y) \tag{2.5}$$

In the following we are going to compare the method qualitatively and quantitatively. In figure 2.1 one can see the pixel response value refocussed at different disparities for all the methods at example points in the testscene „complextestscene“. We chose points close to edges as well as points on clear surface with less structure on it. One can see that the pixel response of the Angular correspondence and the Photoconsistency method for those points show a more consistent behaviour, meaning that only one clear maximum can be seen. The derivative method as well as the Laplace method both seem to have trouble especially on pixel coordinates close to edges. This can be seen most clearly at the edge of the melon, where The first derivative shows 2 Maxima, while the Angular correspondence and Photo consistency measure find a clean maximum indicating at which depth the point can be found. However, on surfaces with less structure as for example on the potato, the Photo consistency measure struggles to find a clear maximum – still it has one at the right position. Having a look at the actual disparity maps produced by the different techniques we can already capture that the angular correspondence and photo consistency method produce the more consistent, smooth disparity maps. A quantitative evaluation confirms this impression: We measure the mean relative error (MRE), which is defined as the mean squared error over all pixels $\vec{p}$ divided by the maximum disparity range (the maximum error possible).

$$MRE = \sum_{\vec{p}} (d_{\text{ground truth}} - d_{\vec{p}})^2 / (\text{max. disp - min. disp}) \tag{2.6}$$

2.1.1 Using depth-from-refocus as a preestimate for the ST

The long-term aim of this work is to make the structure tensor pipeline more robust. We achieve this by using the depth-from-refocus method as a pre

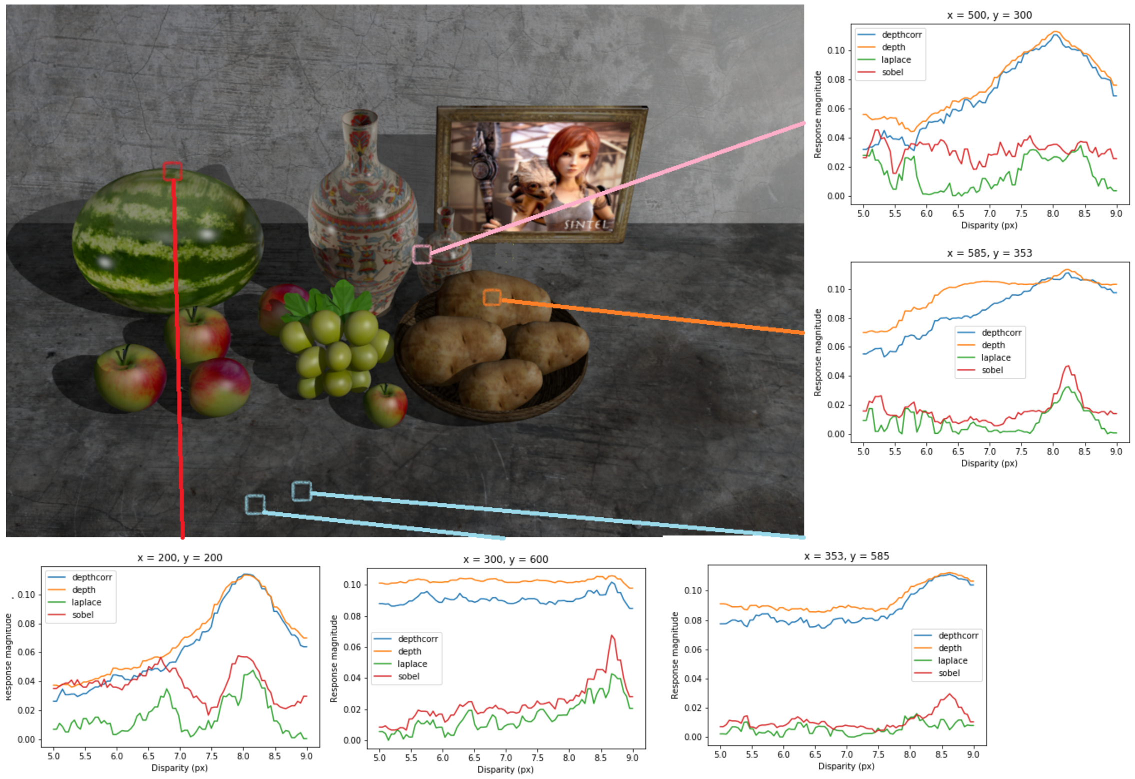

Figure 2.1: At different pixel positions we take a look on how the Pixel response value behaves for the four different depth-from-focus techniques: Photo consistency (orange), Angular correspondence (blue), First derivative (red) and The Laplace method (green) A high value means high confidence (low cost).

Part I

Appendix

.1 Derivation of the structure tensor

The derivation is taken from Jähne [2013]. Taking a function $g : \Omega \rightarrow R, \Omega \subset R^D$, the pereferred local direction $\vec{n} \subset R^D$ must satisfy the following equation:

$$(g^T \vec{n})^2 = |\nabla g|^2 \cos^2(\sphericalangle(g, \vec{n})) \tag{.7}$$

If ∇g is parallel or antiparallel to $\vec{n}$, the expression on the right side reaches a maximum. Therefor one needs to maximise the left hand expression in a local environment:

$$\vec{n}_{\text{preferred}} = \operatorname*{argmax}_n \left(\int w(\vec{x} - \vec{x}') \left(\nabla g(\vec{x'})^T \vec{n} \right)^2 d^D x' \right), \tag{.8}$$

w is a window function defining the size of the local environment. Multipling with $\vec{n}$ we obtain:

$$\vec{n}_{\text{preferred}} = \operatorname*{argmax}_n (\vec{n} J \vec{n}) \tag{.9}$$

$$J = \int w(\vec{x} - \vec{x}') \left(\nabla g(\vec{x'}) \nabla g(\vec{x'})^T \right) d^D x' \tag{.10}$$

This results in a $D \times D$ tensor of the form

$$J_{pq} = \int_{-\infty}^{\infty} w(\vec{x} - \vec{x}') \left(\frac{g(\partial \vec{x'})}{\partial x'_p} \frac{g(\partial \vec{x'})}{\partial x'_q} \right) d^D x'. \tag{.11}$$

In two dimensions we can write

$$J = \begin{pmatrix} w * \frac{\partial g}{\partial x} \frac{\partial g}{\partial x} & w * \frac{\partial g}{\partial x} \frac{\partial g}{\partial s} \\ w * \frac{\partial g}{\partial s} \frac{\partial g}{\partial x} & w * \frac{\partial g}{\partial s} \frac{\partial g}{\partial s} \end{pmatrix}, \tag{.12}$$

where „$*$ “describes a convolution.

A Lists

A.1 List of Figures

A.2 List of Tables

B Bibliography

[1] Website of the hypermedia image processing. http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm. Accessed: 2018-04-16.

[2] Website of the iwr heidelberg. https://klimt.iwr.uni-heidelberg.de/Staff/swanner/. Accessed: 2018-04-06.

[3] what-when-how.com. http://what-when-how.com/introduction-to-video-and-image-processing/morphology-introduction-to-video-and-image-processing-part-2/. Accessed: 2018-04-16.

[4] Edward H Adelson, James R Bergen, et al. The plenoptic function and the elements of early vision. 1991.

[5] Josef Bigun. Optimal orientation detection of linear symmetry, 1987.

[6] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International journal of computer vision*, 1(1):7–55, 1987.

[7] Maximilian Diebold. *Light-Field Imaging and Heterogeneous Light Fields*. PhD thesis, 2016.

[8] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.

[9] Heiko Hirschmüller. Semi-global matching-motivation, developments and applications. *Photogrammetric Week 11*, pages 173–184, 2011.

[10] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*. Springer, 2016.

[11] Bernd Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, 2013.

[12] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera.

[13] Michael W Tao, Pratul P Srinivasan, Sunil Hadap, Szymon Rusinkiewicz, Jitendra Malik, and Ravi Ramamoorthi. Shape estimation from shading, defocus, and correspondence using light-field angular coherence. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):546–560, 2017.

[14] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[15] Sven Wanner. *Orientation Analysis in 4D Light Fields*. PhD thesis, 2014.

[16] Masahiro Watanabe and Shree K Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 27(3):203–225, 1998.

[17] Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):926–954, 2017.

[18] Zhimin Xu, Jun Ke, and Edmund Y. Lam. High-resolution lightfield photography using two masks. *Opt. Express*, 20(10):10971–10983, May 2012. doi: 10.1364/OE.20.010971. URL http://www.opticsexpress.org/abstract.cfm?URI=oe-20-10-10971.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den (Datum) ....