INAUGURAL - DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich - Mathematischen

Gesamtfakultät

der Ruprecht - Karls - Universität

Heidelberg

vorgelegt von

Dipl.-Phys. Sven Wanner

aus Sindelfingen

Tag der mündlichen Prüfung: 06.02.2014

Orientation Analysis in 4D Light Fields

Gutachter: **Prof. Dr. Bernd Jähne**
PD. Dr. Christoph Garbe

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Analyse von 4D Lichtfeldern. Als Lichtfeld bezeichnen wir in diesem Zusammenhang eine Serie von digitalen 2D Bilder einer Szene die auf einem planaren regulären Gitter von Kamerapositionen aufgenommen werden. Essenziell ist dabei die Aufnahme einer Szene mittels vieler Kamerapositionen konstanten Abstandes zueinander. Dadurch werden die von einem Punkt der Szene ausgehenden Lichtstrahlen als Funktion der Kameraposition abgetastet. Dadurch ergibt sich die bereits erwähnte Vierdimensionalität der Daten da, im Gegensatz zu einem klassischen Bild, zusätzlich zur Ortsinformation eine Richtungsinformation der Lichtintensität abgebildet wird.

Lichtfelder sind ein relativ neues Forschungsfeld für die Bildverarbeitung, deren moderner Ursprung eher in der Computergrafik zu suchen ist. Dort wurden sie verwendet, um die aufwendige Modellierung der 3D Geometrie zu umgehen und mittels Interpolation der Blickwinkel auch ohne Informationen über die Geometrie einen interaktiven 3D Eindruck zu erzielen. Die vorliegende Arbeit hat die umgekehrte Intention und möchte aufgenommene Lichtfelder dazu verwenden um die Geometrie der Szene zu rekonstruieren. Der Grund ist, dass Lichtfelder im Vergleich zu existierenden Verfahren der 3D Rekonstruktion einen viel reicheren Informationsgehalt besitzen. Durch die reguläre Abtastung des Lichtfeldes werden neben Information über die Geometrie ebenfalls Materialeigenschaften abgebildet. Oberflächen, deren visuelle Erscheinung sich unter Änderung des Betrachtungswinkels nicht konstant verhält, führen bei bekannten passiven Rekonstruktionsverfahren zu großen Problemen. Das Verhalten solcher Oberflächen unter Blickwinkeländerung wird in Lichtfeldern allerdings abgetastet und somit unmittelbar analysierbar.

Der wissenschaftliche Beitrag dieser Arbeit besteht aus verschiedenen Teilbeiträgen. Es wird ein neues Verfahren vorgestellt, das aus den Rohdaten einer Lichtfeldkamera (Plenopik Kamera 2.0) ohne explizite pixelweise Vorberechnung der Tiefeninformation eine 4D Lichtfeldrepräsentation erzeugt. Diese spezielle Repräsentation, auch *Lumigraph* genannt, ermöglicht den Zugang zu *Epipolarebenen* genannten 2D-Unterräumen dieser Datenstruktur. Es wird ein Verfahren vorgestellt das aus einer Analyse dieser *Epipolarebenen* eine robuste Tiefenschätzung unter der Annahme *Lambertscher* Oberflächen ermöglicht. Darauf aufbauend wird eine Erweiterung dieses Verfahrens auf kompliziertere Materialien, zum Beispiel spiegelnder oder teiltransparenter Oberflächen, entwickelt. Als Anwendungsbeispiele für die inherent vorhandene Tiefeninformation in Lichtfeldern werden bekannte Verfahren wie Superresolution oder Objektsegmentierung auf Lichtfelder erweitert und mit Ergebnissen auf Einzelbildern verglichen. Außerdem ist im Laufe dieser Arbeit eine große Benchmark Datenbank, bestehend aus simulierten und realen Lichtfeldern entstanden, mit Hilfe derer die hier vorgestellten Verfahren getestet werden, und die zukünftiger Forschung auf diesem Feld als Vergleichsbasis dienen soll.

Summary

This work is about the analysis of 4D light fields. In the context of this work a light field is a series of 2D digital images of a scene captured on a planar regular grid of camera positions. It is essential that the scene is captured over several camera positions having constant distances to each other. This results in a sampling of light rays emitted by a single scene point as a function of the camera position. In contrast to traditional images – measuring the light intensity in the spatial domain – this approach additionally captures directional information leading to the four dimensionality mentioned above.

For image processing, light fields are a relatively new research area. In computer graphics, they were used to avoid the work-intensive modeling of 3D geometry by instead using view interpolation to achieve interactive 3D experiences without explicit geometry. The intention of this work is vice versa, namely using light fields to reconstruct geometry of a captured scene. The reason is that light fields provide much richer information content compared to existing approaches of 3D reconstruction. Due to the regular and dense sampling of the scene, aside from geometry, material properties are also imaged. Surfaces whose visual appearance change when changing the line of sight causes problems for known approaches of passive 3D reconstruction. Light fields instead sample this change in appearance and thus make analysis possible.

This thesis covers different contributions. We propose a new approach to convert raw data from a light field camera (plenoptic camera 2.0) to a 4D representation without a pre-computation of pixel-wise depth. This special representation – also called *the Lumigraph* – enables an access to epipolar planes which are sub-spaces of the 4D data structure. An approach is proposed analyzing these epipolar plane images to achieve a robust depth estimation on *Lambertian* surfaces. Based on this, an extension is presented also handling reflective and transparent surfaces. As examples for the usefulness of this inherently available depth information we show improvements to well known techniques like super-resolution and object segmentation when extending them to light fields. Additionally a benchmark database was established over time during the research for this thesis. We will test the proposed approaches using this database and hope that it helps to drive future research in this field.

Acknowledgements

I would like to thank Prof. Bernd Jähne for supervising me during this thesis and even more for trusting in me when offering this PhD position. In a PhD project you can often only be as good as the people teaching, supporting and guiding you.

I would also like to thank Dr. Janis Fehr who was my postdoc during the important first months of the project which helped me a lot to find an access to the topic. Unfortunately he left the institute about half a year after I joined the project due to a lucky event in his life, the birth of his first child.

A few months later the former position of Janis Fehr was occupied by Dr. Bastian Goldlücke. During the rest of my thesis I had a very fruitful working relationship with Dr. Goldlücke. Our strengths and interests fit perfectly together resulting in eight publications in two years and I would like to thank him for this nice, knowledge expanding and interesting time and wish him the best for his upcoming professorship.

I thank PD. Dr. Christoph Garbe for agreeing to be my second referee.

I would also like to thank the collaborators from *Robert Bosch GmbH*, especially Dr. Ralf Zink for supporting me and the fruitful discussions we had.

Many thanks to Dr. Bastian Goldlücke, Dr. Harlyn Baker, Christoph Straehle and Dr. Michel Janus for reviewing and the feedback they gave.

My special thank goes to the *Heidelberg Collaboratory for Image Processing* in general, and all the amazing colleagues there made it more than only a working place. The wonderful secretaries Barbara Werner, Karin Kruljac and Evelyn Wilhelm always gave a helping hand when needed. I really enjoyed the time at this institute and will definitely miss it in the future.

Contents

The body of the air is full of an infinite number of radiant pyramids caused by the objects located in it. These pyramids intersect and interweave without interfering with each other during the independent passage throughout the air in which they are infused.

Leonardo da Vinci (1452-1519)

I say that if the front of a building –or any open piazza or field– which is illuminated by the sun has a dwelling opposite to it, and if, in the front which does not face that sun, you make a small round hole, all the illuminated objects will project their images through that hole and be visible inside the dwelling on the opposite wall which may be made white; and there, in fact, they will be upside down, and if you make similar openings in several places in the same wall you will have the same result from each. Hence the images of the illuminated objects are all everywhere on this wall and all in each minutest part of it.

Leonardo da Vinci (1452-1519)

1 Introduction

1.1 Motivation

Depth imaging has been a highly active research area for decades. Considering the vast number of application areas, this is not very surprising. These range from industrial inspection to robotics, from automotive to surveillance – to name only a few that are long established. However, in the last years, new areas of interest have emerge to drive the developments in that field. Recent advances in the mobile and gaming industry offer more and more depth range data and the upcoming era of 3D printing and rapid prototyping is currently opening a new field of interests in 3D reconstruction. This great demand of depth imaging resulted in a wealth of techniques and devices. One of the first established is the so called *stereo imaging* or *triangulation*. Inspired by the visual system of mammals, two cameras can be placed next to each other looking in the same direction. The resulting images can be used to determine the shift between objects in the corresponding images which is related to the distance of the object to the image planes. Stereo imaging is one of the most well developed approaches considering the number of existing setups and algorithms. The reason for this success is the simplicity of the system and that the algorithms are relatively straightforward, at least for the basic approaches.

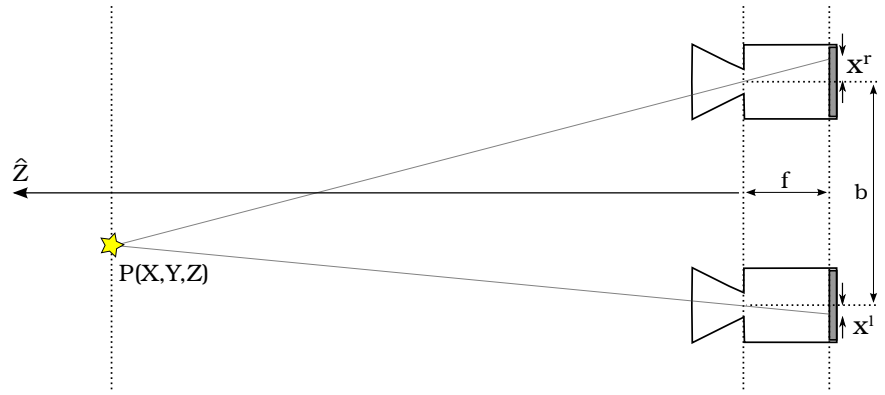


Figure 1: Stereo camera setup. An object at $P = (X, Y, Z)$ is mapped onto two camera sensors. In the left camera at x^l and in the right camera at x^r. The camera sensor centers have a distance b from each other – also called baseline. For objects far away from the camera lens, the parameter f is equivalent to the focal length.

As depicted in figure 1, a stereo setup consists of two cameras at a distance b. A point P is then projected onto different pixel positions on the image plane. The difference of the relative projections of P is $x^r - x^l$ known as *parallax* or *disparity* d. The disparity is inversely proportional to the distance Z of the object (see equation 1) [45].

$$d = x^r - x^l = f\left(\frac{X + \frac{b}{2}}{Z} - \frac{X - \frac{b}{2}}{Z}\right) = b\frac{f}{Z} \tag{1}$$

The statistical uncertainty can be derived using *Gaussian error propagation*

$$\Delta Z = \frac{Z^2}{bf}\Delta d \tag{2}$$

In equation 2 one can see that the uncertainty ΔZ of the measured depth increases with the square of the distance Z^2 [45].

Algorithmically – due to the vast number of methods handling the stereo problem – we will only discuss a very basic approach as an example. A well known method consists of 3 computation steps [83] [52]:

1. compute matching costs of intensities I_r,I_l at disparity d using for example one of the following cost functions. *Sum of Absolute Differences* (SSD), *Sum of Squared Differences* (SAD), *Normalized Cross Correlation* (NCC).

$$\begin{aligned} SAD(x,y,d) &= \sum_{x,y\in W} |I_l(x,y) - I_r(x,y-d)| \\ SSD(x,y,d) &= \sum_{x,y\in W} (I_l(x,y) - I_r(x,y-d))^2 \\ NCC(x,y,d) &= \frac{\sum_{x,y\in W} I_l(x,y)\cdot I_r(x,y-d)}{(\sum_{x,y\in W} I_l^2(x,y))\cdot(\sum_{x,y\in W}\cdot I_r^2(x,y-d))} \end{aligned} \tag{3}$$

2. for each disparity assumption, sum matching costs over a square window.

3. select optimal disparity as the minimal aggregation cost (AGC) at each pixel.

$$d_{opt}(x,y) = \text{ argmin } AGC(x,y,d) \text{ where } AGC \text{ is } SAD, SSD \text{ or } NCC \tag{4}$$

From equation 3, it is quite obvious that to match correspondences, the presence of texture variance is obligatory. If no high frequency textures are available, other triangulation techniques have been developed which use active light sources to replace the missing texture. Those can be implemented as a series of stripe patterns matching the pattern deformation or as a projection of random patterns onto the objects, to give only two examples. Disadvantages of the active methods are that they do not work under arbitrary lighting conditions or on reflective materials. However, the main problem of all triangulation based algorithms is the fact that the underlying principle is a correspondence search. This means that the same features or regions in all corresponding images need to be found to determine the relative shift between them. However, the basic prerequisite for this is, that the appearance or the color of those regions stays the same from both viewpoints. This so-called *Lambertian* assumption, namely that the observed color of a 3D point is independent of the point of view, is the main problem of correspondence search because most materials do not behave like this.

Another technique to measure the depth of a scene is *Time-of-Flight* (*ToF*) imaging. ToF is an active range estimation method based on measuring the time a light pulse requires to travel from the emitter to the object and back to the camera. A quite old and famous 1D realization is the *LIDAR* (Light Detection And Ranging) [87], often used in the field of self-driving cars.

The important equation for a *ToF* system is

$$\tau = \frac{2z}{c}, \tag{5}$$

where τ is the travel time, z the distance of an object to the camera and c the speed of light. The cameras consist of four main components, an illumination unit, an optic, a sensor and a complex electronic read out unit. The illumination unit often consists of LEDs or laser diodes emitting in the near infrared spectrum. There are two possible operation modes. Either the LEDs emit light pulses or a continuous wave modulation. In the second case, instead of traveling time a phase shift is measured.

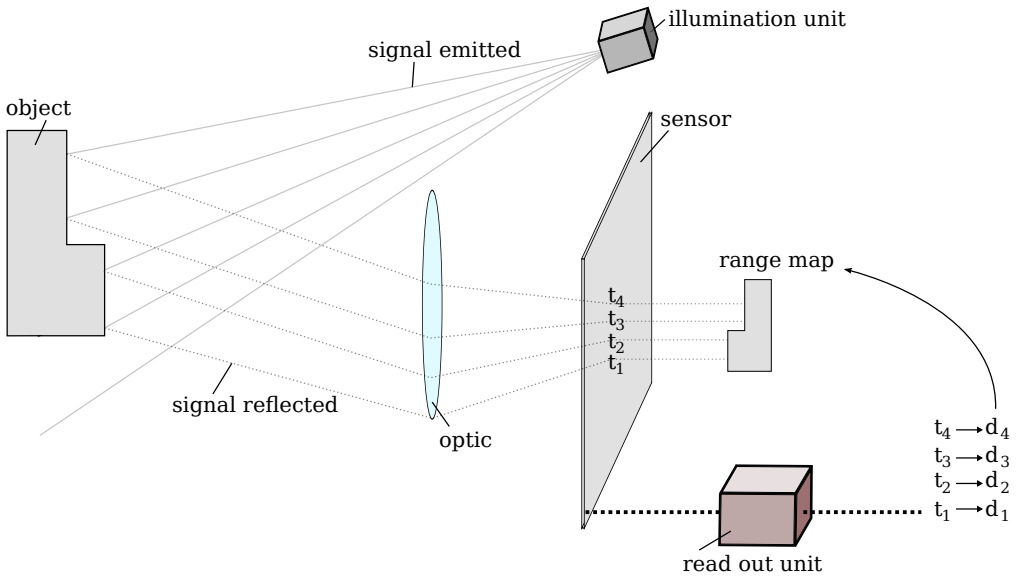


Figure 2: A simplified sketch of a *Time-of-Flight camera*. A signal is emitted by the illumination unit, reflected by an object and measured on the sensor element able to measure time dependent on the incoming intensities. The signal is then processed by the read-out unit to estimate the distance of the reflected signals measured at each pixel.

Time-of-Flight cameras can measure distances between a few centimeter and a few dozen meters. For periodically modulated light sources, there is a natural limitation of the maximum measurable range due to an ambiguity of periodic signals with a phase shift of

$$\Delta\phi = 2\pi k, \quad k \in \mathbb{N}. \tag{6}$$

the maximum range that can be measured then depends on the frequency ν of the light source

$$d_{max} = \frac{c}{2\nu}. \tag{7}$$

These ranges can be extended i.e. by using combined measurements of multiple modulation frequencies [36]. *ToF cameras* using pulsed illumination have similar problems. The depth range limitation is not driven by a non ambiguity of the signal as in equation 6 but by the integration time necessary to wait for back-projected light pulses.

The sensors in *ToF cameras* are much more complicated than in a standard digital camera. Every pixel must be able to measure the light travel time separately. Thus the pixels are huge (around $100\mu m$) compared to the pixels of a *CCD sensor* which are around $10\mu m$. This leads to one of the main disadvantages of this type of range camera, quite poor resolution. Currently they achieve sensor resolutions of around 200×200 pixels. Another disadvantage is that the distance measurement only works on materials able to reflect the light frequency of the illumination unit. Also multiple reflections in the scene as well as mutual interferences between different *ToFs* are known problems. The accuracy of the depth measurement theoretically does not depend on the distance of the objects, but in practice it does. Due to the fact that light intensity I drops off with $1/z^2$, the signal to noise ratio increases for objects with increasing distance to the sensor, which of course affects the accuracy.

A third important technique to estimate depth ranges is the so called *Interferometry*. This is a method based on measuring the interference of a reference beam and the beam reflected by an object. The principle is sketched in figure 3. A source emits light which first goes through an aperture and a collimator lens to create a planar and coherent wavefront. A beam splitter separates the wavefront into a measuring and a reference beam. The reference beam is reflected by a mirror back to the beam splitter where it is reunited with the measuring beam backscattered by the object surface. If both path lengths from the reference and the measuring beam are the same, by constructive interference, the reunited beam causes a maximum intensity signal in the *CCD camera* measuring it. By moving the reference or the object arm, a scanning of the surface can be achieved. The accuracy of measurement is in the range of the wavelength used, which means in the scale on nanometers. A price for this precision is that much effort is necessary to stabilize the system. Mechanical and thermal disturbances are critical. It is also very hard to apply this technology to measure bigger objects, thus interferometry is widely used in scientific and industrial environments to measure small objects very precisely, but it is not a very flexible range estimation technique.

Aside from the depth imaging techniques discussed up to now, *light field photography* is developing as a new technology for high quality passive range estimation. A light field is a dense and regular sampling of a scene. This enables a measurement of spatial and direction dependent intensities instead of only spatially measured intensities. In fact, this is a quite old idea which goes back to the early 20th century, and Gabriel Lippmann

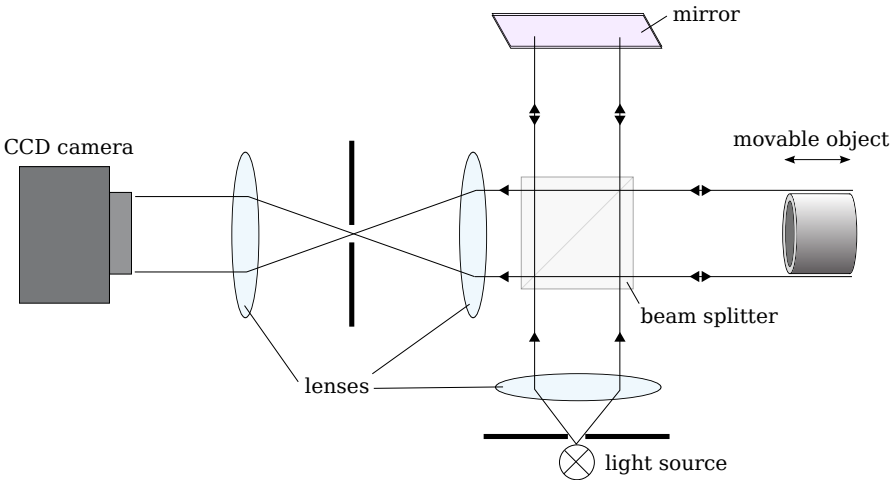


Figure 3: Sketch of an *Interferometer* setup. A light source emits light trough a pinhole and a collimator lens to guarantee planar and coherent wavefronts. The light rays then are separated by a beam splitter into a measuring and a reference beam reunited again in the beam splitter before captured by a *CCD camera*. Same path length of measuring and reference beam causes constructive interference which can be used to measure the object surface elevation.

who first thought of this idea named it *Integral Photography* [61]. His method of light field capture was more or less ignored for 100 years before being rediscovered about 10 years ago. First people researching in *computer graphics* discovered light fields as a possible solution to skip the 3D geometry modeling stage by instead using a collection of images to interpolate intermediate views of a scene resulting in an interactive 3D experience. In fact, this statement is simplified because there are various techniques established in Image-based rendering [23, 24, 41, 54, 58, 65, 85, 86, 89] which can be classified as techniques based on rendering without geometry, with implicit geometry and with explicit geometry. However, the main goal is usually the generation of novel views from existing images of a scene with or without geometry present. Reviews of this techniques can be found in Kang [48] and Shum [88].

In the *computer vision community* people are more interested in sampling a light field of a scene to explicitly reconstruction the geometry. Ways to capture light fields are diverse, but the principle is always to achieve a dense sampling of the cones of light rays emitted by each point on the surface of a captured object. We will see in this work that a dense and regular sampling of a scene, what we call a light field, allows more that just a reconstruction of geometry. Due to the mentioned sampling constraints, also material properties of the captured objects are mapped onto the sensor(s). This becomes clear when realizing that material properties can be described using the *Bidirectional Reflectance Distribution Function* (*BRDF*). The *BRDF* is a function describing the measured intensity of an opaque surface depending on the incoming and outgoing light rays and the normal vector of the surface. The fundamental assumption of algorithms based on triangulation is that the observed scene point behaves *Lambertian*, which is

equivalent to a constant *BRDF*. In other words, the color of an observed object point does not depend on the observation direction. In reality not many existing materials fulfill this assumption. A lot of research has been done in the area of stereo vision to design algorithms more robust against such glossiness effects. Although most objects in natural scenes can be seen as *Lambertian*, the problems increase the higher resolved or the nearer to the camera objects are. Especially for tasks of high quality 3D object reconstruction, playing a role for example in industrial inspection, *non-Lambertian* effects gain in importance and need to be handled. To gain robustness, stereo setups can be extended to multiple cameras, providing more views of the same object point and thus to more possible correspondence. But due to the fact that all such algorithms still are based on searching for corresponding features in different images, this only comes with more and more complexity of the algorithms and increasing computation time. All these methods try to combat a lack of, or an ambiguity, in information with ever improving error handling. If instead a light field camera samples a subset of the light rays emitted by an object, it actually performs a sparse sampling of the *BRDF*. This makes reconstruction of the geometry and also of the *BRDF* possible. Methods analyzing light fields thus inherently should be more robust against *non-Lambertian* effects, because information about the material is really measured and not only causing ambiguities.

The goal of this thesis is an analysis of light fields from the *computer vision* point of view. We use a specific parametrization throughout the entire work, called a 4D light field or *Lumigraph*, which is a well suited representation, for example, giving access to effects caused by the *BRDF*. The main contributions are methods to analyze 4D light fields primarily aimed at geometry reconstruction of objects under *Lambertian* and *non-Lambertian* assumptions.

The work presented in this thesis was funded by *Robert Bosch GmbH*, Stuttgart.

1.2 Outline

Section 2: We give an overview over light fields in the context of image processing. In contrast to computer graphics, where light fields were developed to avoid the necessity of geometry, we are interested in acquiring them to reconstruct geometry as a primary goal. Due to that, we first introduce the most general definition of light fields before discussing the parametrization and data representation we use in this work. After that, we recap different methods of real world light field acquisition as well as simulation using computer graphics.

Section 3: After an introduction of epipolar plane images and their benefits for the analysis of light fields as well as a discussion of the problems of Focused Plenoptic Cameras, we present an algorithm to compute a real 4D representation from Plenoptic 2.0 Camera raw data without a pre-computation of an explicit pixel-wise depth estimation. This section is based on the publication "*Generating EPI representations of 4D Light fields with a single lens focused plenoptic camera*" [104].

Section 4: This section discusses the 4D light field data used in this work. We introduce our benchmark data set consisting of simulated and real world light field data providing ground truth depth at least for the center view. The corresponding publication is "*Datasets and Benchmarks for Densely Sampled 4D Light Fields*" [105], a joint work with Bastian Goldlücke and Stephan Meister.

Section 5: In this section, we discuss geometry reconstruction using light fields, in particular epipolar plane images. Compared to methods based on correspondence search, we propose an orientation analysis on epipolar plane images which can be implemented via fast and robust filtering approaches. The section splits into two parts, single orientation and double orientation estimation. The range estimation using single orientation analysis is based on the Lambertian assumption that the color of a scene point is independent of the viewpoint. This part is based on the publications "*Globally consistent depth labeling of 4D light fields*" [100] and "*Variational Light Field Analysis for Disparity Estimation and Super-Resolution*" [103]. In general the Lambertian assumption is not a valid description for real world objects. If materials show a shininess or, even worse, act as a mirror, this has an effect on the structure of the epipolar plane images. How this effect looks and how to analyze these more complex epipolar planes we discuss in the second part of this section, the double orientation analysis. This part is based on the publication "*Reconstructing Reflective and Transparent Surfaces from Epipolar Plane Images*" [102]. Along with the local analysis of the epipolar planes we discuss variational frameworks to improve the results and to guarantee global consistency. The theoretical part of this optimization techniques as well as fast *CUDA* [71] implementations of the developed algorithms, gathered in the *cocolib* [37], are the work of Bastian Goldlücke.

Section 6: With the readily available range information light fields are providing, further scene analysis can be done. We develop two frameworks based on light field processing. First we discuss a super-resolution framework tailored to light fields. The corresponding publications are "*Spatial and angular variational super-resolution of 4D light fields*" [101] and "*Variational Light Field Analysis for Disparity Estimation and Super-Resolution*" [103]. A second project is object segmentation in light fields. Here, we will see that light fields are highly suitable for segmentation tasks. Problems of classifiers acting on single image domains can be overcome and by labeling rays consistent over the entire light field much better results compared to single pixel labeling can be achieved. The publication corresponding to this part is "*Globally Consistent Multi-Label Assignment on the Ray Space of 4D Light Fields*" [106]. The publication is a joint work with Bastian Goldlücke and Christoph Straehle, whereby also here the theoretical background as well as fast implementations on the GPU of the variational methods are the work of Dr. Goldlücke.

1.3 Contribution

The following is a list of what the author believes to be the novel contributions of this thesis:

- a novel algorithm to convert raw data of *Plenoptic 2.0 Cameras* into the *Lumigraph* representation without pre-computing a pixel-wise distance measure.
- a benchmark database consisting of real-world and simulated 4D light fields providing ground truth depth and partly ground truth object labels.
- a new approach for range estimation using orientation analysis in light fields
- an extension of the single orientation analysis to double orientation patterns for reconstructing reflections and transparencies.
- an evaluation of applications of the orientation analysis such as super-resolution of light fields and ray-space segmentation.

2 Light Fields

2.1 The Plenoptic Function

One of the fundamental papers introducing the concept of light fields is "*The Plenoptic Function and Elements of Early Vision*" from Adelson and Bergen [2]. They ask the question what the actual information about the world is which is contained in the light filling the space an observer is looking at. Starting from this question they develop a theory of the *plenoptic function*.

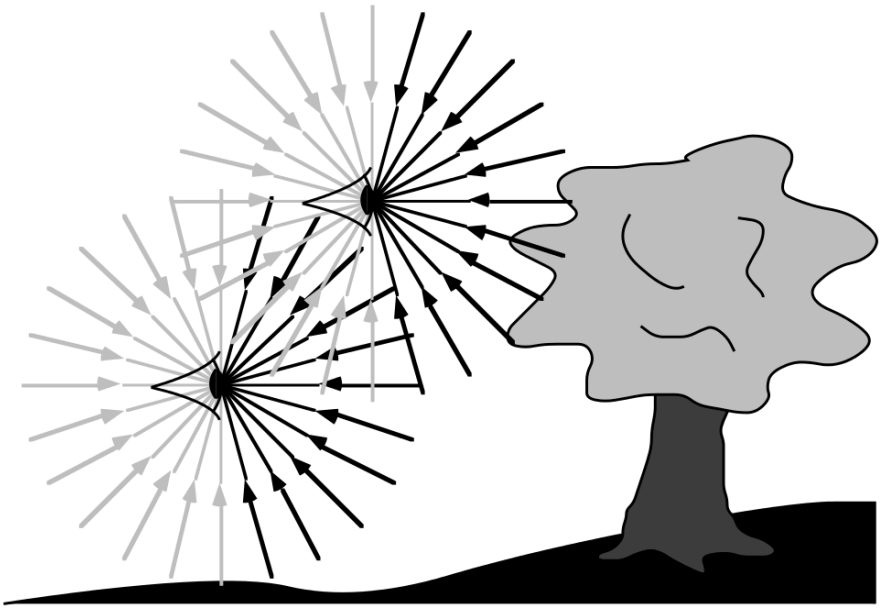


Figure 4: A widely spread visualization of the *plenoptic function*. We cite the original caption of the figure: "*The plenoptic function describes the information available to an observer at any point in space and time. Shown here are two schematic eyes-which one should consider to have punctuate pupils-gathering pencils of light rays. A real observer cannot see the light rays coming from behind, but the plenoptic function does include these rays.*" (Adelson and Bergen [2])

.

If we capture a gray value image of a scene – using a pinhole camera– we select a cone shaped bundle of rays at a specific position in space V_0 and accumulate their intensities on the sensor of our camera. Thus we measure an intensity distribution $P(\theta, \phi)$ or $P(x, y)$, depending on the type of coordinate system we use. Taking the lights wavelength λ into account we can add another dimension $P(x, y, \lambda)$. If in a next step we measure the whole space $V \in \mathbb{R}^3$ instantaneously we gain three dimensions more, and when also including the time t, we end up with a seven dimensional function describing the entire information about the light filling the space over time

$$P(x, y, \lambda, V_x, V_y, V_z, t). \tag{8}$$

There might be even more dimensions if we consider the polarization states of the light rays, but we neglect this here and concentrate in the following on the *plenoptic function* in equation 8. In general, this function definition seems a bit abstract, but in fact, every imaging device samples sparse subsets of this function. Considering this, the concept of a *plenoptic function* can serve as a general framework to think about possible imaging modalities. Another example, besides the standard pinhole camera which can be described as $P(x, y)$, this work is about a sampling of the *plenoptic function* of type $P(x, y, V_x, V_y)$ [107].

2.2 The Lumigraph Parametrization

In the previous section 2.1, we discussed the *plenoptic function* as describing the entire information on the light filling the space around an object. Due to the fact that all imaging techniques are sparse samples of this general function, in this section we introduce the sparse sampling or parametrization this work concentrates on.

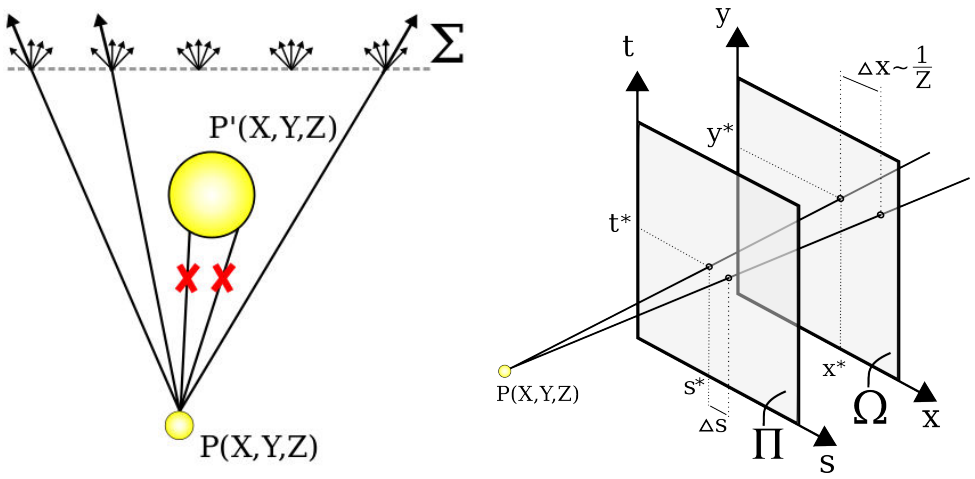
If we assume a sampling of the *plenoptic function* using a gray value camera, we can first neglect the wavelength λ dependency in equation 8. Furthermore this work is about static scene reconstruction, so we are not interested in optical flow estimation or any other time dependent properties of the scene and thus can cancel out the dimension t as well. Another reduction in dimensionality can be achieved if we assume that the intensity of a light ray does not depend on the actual position on the ray, which is equivalent to the assumption that we parametrize the light field on a surface Σ outside of the convex hull of the scene (compare figure 5 left).

Several ways to represent light fields have been proposed. Here, we adopt the light field parametrization from early works in motion analysis from Bolles et al. [16] and the work about light field sampling from Gortler et al. [41]. The idea of a convex hull to reduce the *plenoptic function* has also been used by Benton [10] and similar ideas can be found in Ashton [5], where the movement of the camera is restricted to a spherical surface for an illumination analysis.

One way to look at a 4D light field is to consider it as a collection of pinhole views from several view points parallel to a common image plane (see figure 5). The 2D plane Π contains the focal points of the views, which we parametrize by the coordinates (s, t), and the image plane Ω is parametrized by the coordinates (x, y). A 4D light field or *Lumigraph* [41] then is a map

$$L : \Omega \times \Pi \to \mathbb{R}, \qquad (x, y, s, t) \mapsto L(x, y, s, t). \tag{9}$$

It can be viewed as an assignment of an intensity value to the ray passing through $(x, y) \in \Omega$ and $(s, t) \in \Pi$.



(a) Opaque surface only assumption (b) Two plane parametrization

Figure 5: (a) Dimensionality reduction of the *plenoptic function* through parameterizing the light field on a surface Σ by assuming that the intensity of light rays does not change in the free space between object surfaces and the imaging device. In other words, the intensity of the rays from the object at P which are occluded by the object at P' is not of interest. (b) Each light ray can be parametrized by the intersection point with two planes. Each camera location (s^*, t^*) in the 2D plane Π yields a different pinhole view of the scene. Together with the second intersection (x^*, y^*) at the image plane Ω we can parametrize a light field as a four dimensional subspace $L(x, y, s, t)$ of the *plenoptic function* (see equation 9).

2.3 Epipolar Plane Images

For the problem of estimating the 3D structure of a sampled scene, we consider the structure of the light field, in particular on 2D slices through the field. We fix a horizontal line of constant y^* in the image plane and a constant camera coordinate t^*, and restrict the light field to an (x, s)-slice Σ_{y^*,t^*}, respectively to an (y, t)-slice $\Sigma_{x^*s^*}$. The resulting map is called an epipolar plane image (EPI). This idea goes back to Bolles et al. [16].

$$\begin{aligned} S_{y^*,t^*} : \Sigma_{y^*,t^*} &\to \mathbb{R}, \\ (x, s) &\mapsto S_{y^*,t^*}(x, s) := L(x, y^*, s, t^*). \end{aligned} \tag{10}$$

Let us consider the geometry of this map (compare figures 5 and 6). A point $P = (X, Y, Z)$ within the epipolar plane corresponding to the slice projects to a point in Ω depending on the chosen camera center in Π. If we vary s, the coordinate x changes according to

$$\Delta x = \frac{f}{Z}\Delta s, \tag{11}$$

where f is the distance between the parallel planes. Note that to obtain this formula Δx has to be corrected by the translation Δs to account for the different local coordinate systems of the views. Interestingly, a point in 3D space is thus projected onto a line in Σ_{y^*,t^*}, where the slope of the line is related to its depth. This means that the intensity

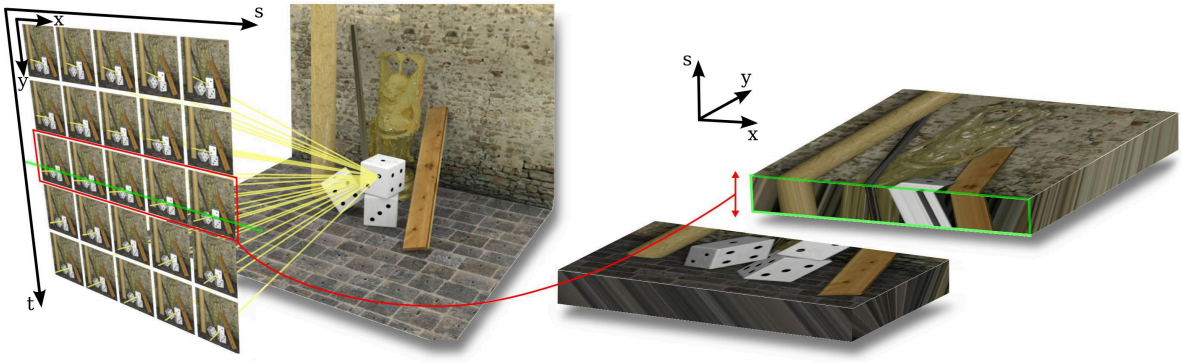


Figure 6: The left side depicts a collection of images sampling a 3D scene. The images are captured on planar 2D grid with constant baselines. This is what is called the *Lumigraph* parametrization (section 2.2). By fixing an angular dimension (visualized via a red box) we extract a 3D subspace of the *Lumigraph*. If we imagine this image sequence as a volume (x,y,s) and cut out a slice along the s-axis, which is equivalent to fixing another spatial dimension (visualized via a green line), the result is an epipolar plane image. In this subspace a point in the world is mapped onto a line whose slope corresponds to the distance of the point to the camera.

of the light field should not change along such a line, provided that the objects in the scene are *Lambertian*. Thus, computing depth is essentially equivalent to computing the slope of level lines in the epipolar plane images. Of course, this is a well-known fact, which has already been used for depth reconstruction in previous works [16, 27]. In sections 5.1 and 5.2, we describe and evaluate novel approaches on how to obtain slope estimates for *Lambertian* and for non *Lambertian* assumptions.

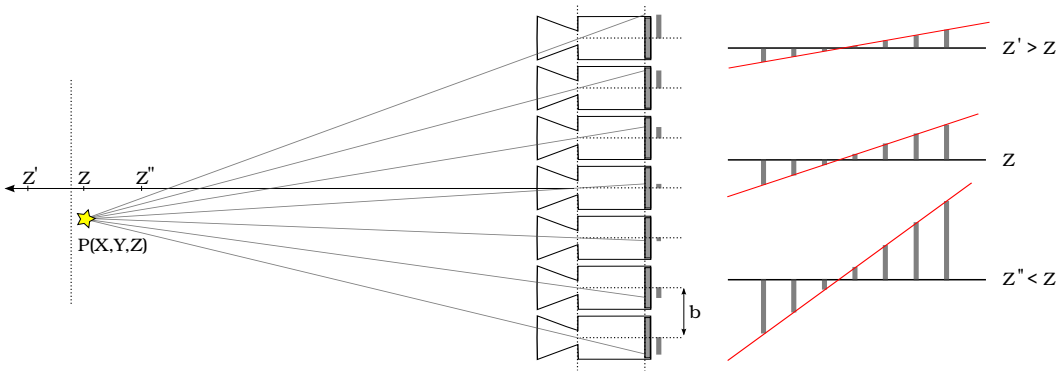


Figure 7: Sketch of a linear camera array. Cameras are lined up with constant baselines b. This leads to a linear mapping of a 3D point onto the sensors. The slope of these lines depends on the distance Z of P to the image plane.

2.4 Acquisition Of Light Fields

2.4.1 The Plenoptic Camera (1.0)

The early beginnings of light field imaging or *Plenoptic Cameras* are strongly connected to Ives (1903) [44] and Lippmann (1908) [61]. Lippmann realized that classical photography, like drawings, only shows one part of the whole and dreamed of an imaging device able to render – "*the full variety offered by the direct observation of objects*". One of his drawings from 1908, depicted in figure 8, already gives some insight in todays realization of *Plenoptic Cameras*.

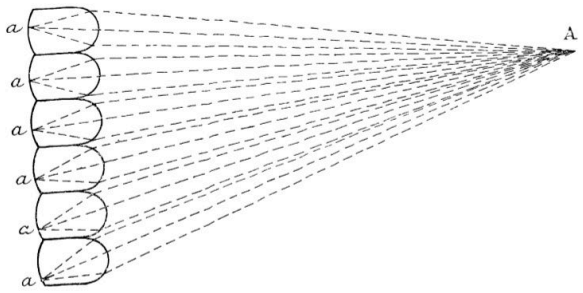


Figure 8: Early drawing of Lippmans so-called integral camera (1908) [61]

The modern approaches of building *Plenoptic Cameras* are mainly influenced by the works of Adelson et al. [3] and Ng et al. [70]. Due to the fact that the principles in detail are well described in those publications and that this work does not deal with data of early versions of *Plenoptic Cameras*, we will here only give a short overview of the basic concepts of the realization and the algorithmic of rendering images from the sensor raw data following Ng et al. [70].

2.4.1.1 Optical Design. The *Plenoptic Camera 1.0* is based on a usual camera with a digital sensor, a main optics and an aperture. The difference from a normal camera is a micro-lens array placed on the focal plane of the main lens exactly at a distance f_{MLA} from the sensor. (see figure 9). This means the micro-lenses themselves are focused at infinity. In contrast with a usual camera which integrates the focused light of the main lens on a single sensor element – the micro-lenses split the incoming light cone by the direction of the rays mapping them onto the sensor area below the corresponding micro-lens.

This means that one has direct access to the intensity of a light ray $L(x^*, y^*, s^*, t^*)$ of the light field by choosing the micro-image of the micro-lens at (x^*, y^*) – encoding the spatial position– and a pixel of the corresponding micro-image (s^*, t^*) – encoding the direction. It should be noted that the size of each micro-lens is coupled to the *aperture* or *f-number* of the main optics. If the micro-lenses are too small compared to the main

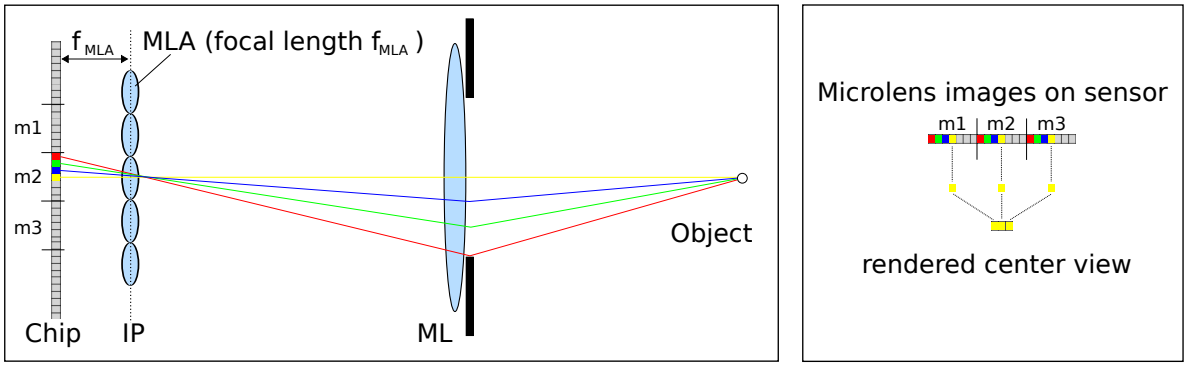


Figure 9: Left: One dimensional sketch of a Plenoptic Camera 1.0 setup. Light rays emitted by the object are focused by the main lens (ML). The microlens array (MLA) is placed at the image plane (IP) of the main lens and thus separates the rays by their direction, mapping them onto the sensor. **Right:** Illustrates the rendering of a single view point, here the center view, by collecting the center pixels of each micro image m_i.

aperture the micro-images overlap each other or – the other way around – it is a waste of sensor area if the micro-lenses are too big. Due to the fact that light passing the main aperture also has to pass a micro-lens before getting integrated on a squared pixel, what actually happens is that the camera measures small 4D boxes of the light field entering the camera instead of single rays.

2.4.1.2 Rendering Views From Raw Data.

Rendering a projective view from sensor raw data is quite simple as depicted in figure 9.

1. Determining a specific projective view means determining a relative position p_{s^*,t^*} within the micro-images, for example the center position p_{center}.

2. Define an output image I_{s^*,t^*} as $M \times N$ matrix where $M, N \in \mathbb{N}$ are the number of micro-lenses in vertical and horizontal directions.

3. Assigning the pixel (x, y) in I_{s^*,t^*} the intensity of the pixel p_{s^*,t^*} in the micro-image corresponding to the micro-lens $q_{x,y}$.

Changing the relative position p_{s^*,t^*} for rendering means changing the virtual aperture which results in an projective view from a slightly different viewpoint. An integration of images from neighboring viewpoints is used to create a depth of field and enables computationally refocusing by varying the relative positions of these neighbored images. "*In quantized form, this corresponds to shifting and adding the sub-aperture images ...*" Ng et. al [70].

In fact the description above neglects a very important calibration step necessary beforehand. Rendering views from the camera raw data is only that easy if the data are rectified and distorted to satisfy the conditions necessary for a successful rendering. A

detailed description of a possible calibration process is out of the scope of this work but can be found in Dansereau et al. [28].

2.4.2 The Focused Plenoptic Camera (2.0)

Besides the *Plenoptic Camera* (see section 2.4.1), another optical setup for a compact light field camera has recently been developed, the *Focused Plenoptic Camera*, often also called the *Plenoptic Camera 2.0* [62, 64, 73]. The main disadvantage of the *Plenoptic Camera 1.0* is the poor spatial resolution of the rendered views, which is equal to the number of micro-lenses. By changing the optical setup a little bit one can increase the spatial resolution dramatically.

2.4.2.1 Optical Design. The main difference in the optical setup between the *Plenoptic Camera 1.0* and *2.0* is the relative position of the micro-lens array. The micro-lenses are no longer placed at the principal plane of the main lens and focused to infinity, but are now focused onto the image plane of the main lens. The result is that each micro-lens then acts as a single pinhole camera, "looking" at a small part of the virtual image inside the camera. This small part is then imaged with a high spatial resolution onto the sensor as long as the imaged scene point is in the valid region between the principal plane of the main lens and the image sensor. Scene features behind the principal plane cannot be resolved. The effect is that scene points – that are not in focus of the main lens but within this valid region – are imaged multiple times over several neighboring micro-lenses, thus encoding the angular information over several micro-images(see also figure 11 and Lumsdaine et al. [62] or Perwass [73]). This makes it possible to encode angular information and preserve high resolution at the same time. But this comes with a price. First, light field encoding is complicated and, second, due to the multiple imaging of scene features, rendered images from this camera have also a much lower resolution than the inherent sensor resolution promises.

2.4.2.2 Rendering Views From Raw Data. The rendering process requires a one time scene independent calibration, which extracts for all micro-lens images (micro-images) the position as well as their diameter d_{ML}. In this work, we use a commercially available camera [72], which has a micro-lens array where the lenses are arranged in a hexagonal pattern.

Due to this lens layout, we also use a hexagonal shape for the micro-images and address them with coordinates (i, j) on the sensor plane. We define an image patch $\widehat{p}_{ij}$ as a micro-image or a subset of it. Projective views are rendered by tiling these patches together [33, 63].

The center of a micro-image (i, j), determined in the coordinate system given by the initial camera calibration process, is denoted by $\vec{c}_{ij}$. The corresponding patch images

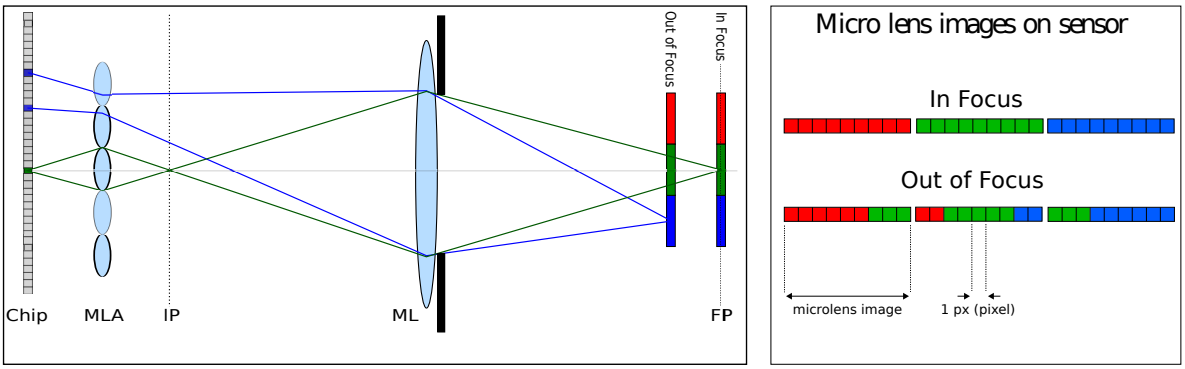


Figure 10: Left: One-dimensional sketch of a *Plenoptic Camera 2.0* setup. Light rays emitted by the object are focused by the main lens (ML) onto the image plane (IP). The micro-lens array (MLA) is placed so that the micro-lenses are focused onto the image plane of the main lens, mapping fractions of the virtual image onto the sensor. Green rays are coming from an object in focus of the main lens (FP), blue rays of an object away from the principal plane of the main lens.
Right: Illustrates the resulting micro-images of an object in and out of focus.

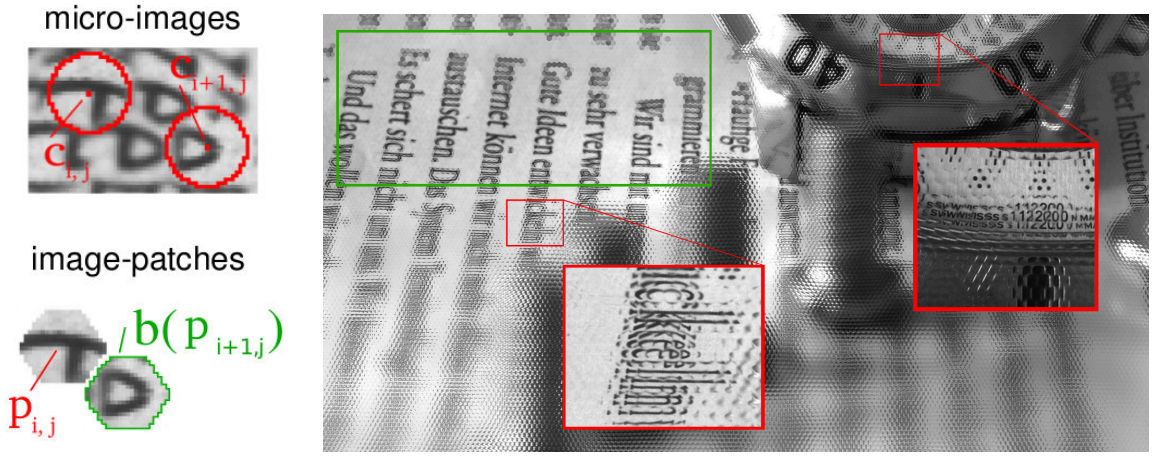
are defined as $\omega_{ij}(\delta, \vec{o})$, where δ denotes the size of the micro-image patch $\widehat{p}_{ij}(\delta, \vec{o})$ in pixels and $\vec{o}$ is the offset on the sensor plane of the micro-image patch center from $\vec{c}_{ij}$. We define $\omega_{ij}(\delta, \vec{o})$ as an $m \times n$ matrix, which is zero except for the positions of the pixels of the corresponding micro-image patch $\widehat{p}_{ij}(\delta, \vec{o})$:

$$\omega_{ij}\left(\delta, \vec{o}\right) = \begin{pmatrix} 0 & & \cdots & & 0 \\ & \ddots & & & \\ \vdots & & \widehat{p}_{ij}\left(\delta, \vec{0}\right) & & \vdots \\ & & & \ddots & \\ 0 & & \cdots & & 0 \end{pmatrix} \tag{12}$$

$m \times n$ is the rendered image resolution and (i, j) is the index of a specific image patch, imaged from microlens (i, j) (see figure 11). A projective view $\Omega\left(\delta, \vec{o}\right)$ of a scene is then rendered as:

$$\begin{aligned} \Omega\left(\delta, \vec{o}\right) &= \sum_{i=1}^{N_y} \sum_{j=1}^{N_x} \omega_{ij}\left(\delta, \vec{o}\right) \\ \delta \in &\mathbb{N} \,|\, 1 < \delta \leq d_{ML} \\ \vec{o} \in &\mathbb{N}^2 \,|\, 0 \leq \| \vec{o} \| \leq \frac{d_{ML}}{2} - \frac{\delta}{2}, \end{aligned} \tag{13}$$

where (N_x, N_y) is the number of micro-lenses on the sensor in x- and y-directions. The choice of the parameters δ and $\vec{o}$ directly controls the image plane and point of view of the rendered view.



(a) Micro images (b) Plenoptic 2.0 Camera raw data example

Figure 11: (a) Micro-images and their centers $\vec{c}_{ij}$ are indicated as well as the resulting image patches $\widehat{p}_{ij}$ and their border pixels $b(\widehat{p}_{ij})$. (b) Raw data from a *Plenoptic Camera 2.0* [73]. All possible optical effects are visible here. The green box in the upper left corner shows the transition from a region in the scene behind the principal plane of the camera main lens to a region exactly on the focal plane so that the imaged fragments perfectly fit together. The red boxes show magnified regions of the scene between the principal plane of the main lens and the sensor so that scene features are imaged multiple times over several neighbored micro images. The amount of multiple feature occurrence depends on the distance to the image plane.

2.4.2.3 Refocusing. It is obvious that rendering a projective view here is a bit more complex than it is for the *Plenoptic Camera 1.0*, where one can simply extract single pixels from the raw data (see section 2.4.1). The reason is the different sampling of the light field in the devices. While a micro-lens in a *Plenoptic Camera 1.0* is focused at infinity and thus decomposes the light rays emitted by a 3D point into their directions, a micro-lens of a *Focused Plenoptic Camera* acts as a single pinhole camera looking at a small subset of the virtual image of the scene. This leads to a much higher resolution but spreads the directional information over multiple micro-images. This causes so-called *plenoptic artifacts* during rendering. The choice of the patch size δ defines a specific image or virtual depth plane in the 3D scene. Neighbored patches $\widehat{p}_{ij}$ with a size *delta* fit perfectly together for all imaged scene features from the corresponding virtual depth plane. Patches whose content is a imaged region not lying on this virtual depth plane, either lack information or the multiple occurrence of scene features is still present. These are the mentioned *plenoptic artifacts* which occur for a fixed patch size δ all over the rendered image, except for the specific virtual depth plane (compare figure 12, or as another example, Lumsdaine et al. [64] Fig. 11). Due to the fact that the multiple occurrence of image features over the micro-images depends on the distance to the camera, image planes nearer to the camera need to be rendered with smaller patch sizes δ and thus show a loss in resolution. Full resolution is only present at the image plane of maximum δ, which is the principal plane of the main lens. A possibility to handle

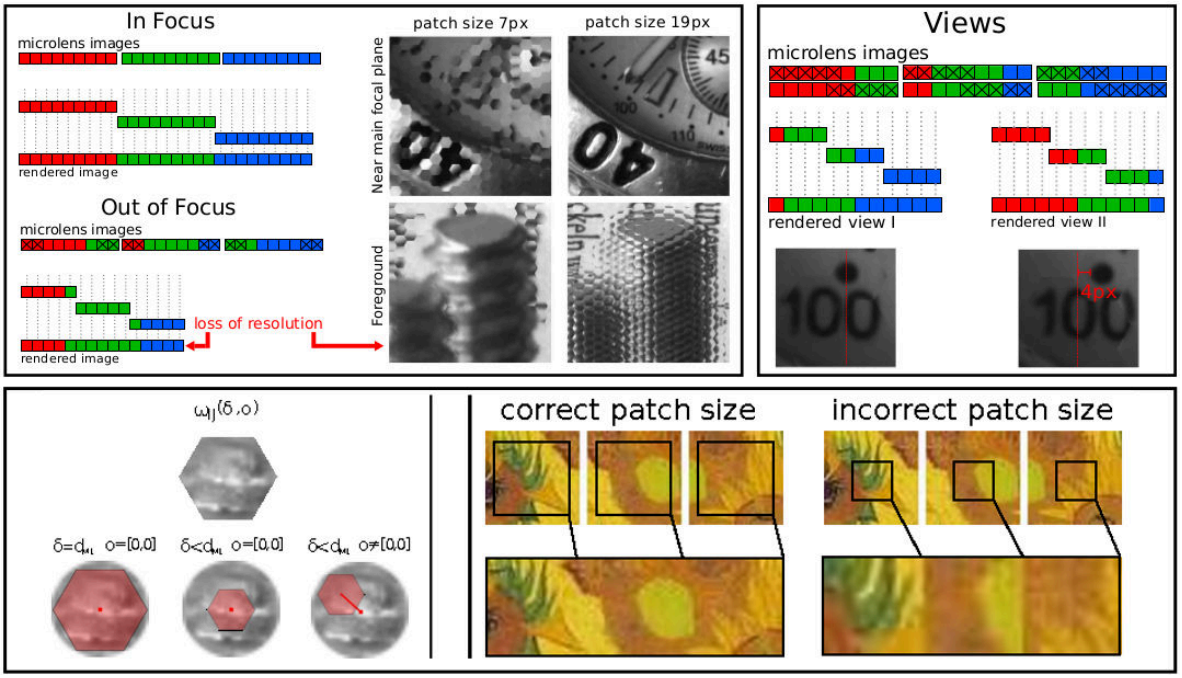


Figure 12: Top: Rendering different image planes and views illustrated on the basis of the 1D sketch depicted in figure 10. On the left side the effect of different patch sizes δ is depicted, on the right the effect of changing the offset $\vec{o}$. **Bottom:** Illustrates on the left a micro-image patch as well as the effect of different values of δ and offsets $\vec{o}$. On the right the reason for plenoptic artifacts is visualized.

the plenoptic artifacts is described in Lumsdaine et al. [25]. They call it blending and achieve with this technique much more realistic looking refocusing results. Another approach can be found in Georgiev et al. [34]. We will propose our approach in this work in section 3.

2.4.2.4 Generating All-In-Focus Views An important aspect of the *Focused Plenoptic Cameras* is that beside the opportunity of computationally refocusing, one could also be interested in rendering images with the largest possible depth of field. This means removing the *plenoptic artifacts* or in other words, eliminating all duplicated scene features captured by the individual micro-lenses.

The common thread of this work is an analysis of light fields based on the analysis of epipolar plane images (see section 2.3). We will see in the following sections how to create and analyze them in detail. In this context it is only important to know that from a *Focused Plenoptic Camera*, we need to render all possible All-In-Focus Views to get access to them. Therefore we will recap in this section some related work to render those full depth of field views and will discuss a new approach in section 3.

We will now quickly discuss two approaches treating this issue. The first is from Perwass

et al. [73]. It is mainly based on triangulation to obtain a pixel-wise depth map over the micro-images (an overview of existing triangulation algorithms can be found in Scharstein and Szeliski [83]). For this, the correlation or sum-of-absolute differences (SAD) is computed over micro-image pairs. This of course works only where a local contrast is present. The computed virtual depth per pixel value gives a hypothesis for a projection cone defining the occurrence of the same image feature in neighboring micro-images. By integrating all connected pixels over those cones, a final image without multiple occurrence of scene features can be rendered. A quite similar approach using *multi-view stereo* is described in Bishop et al. [15].

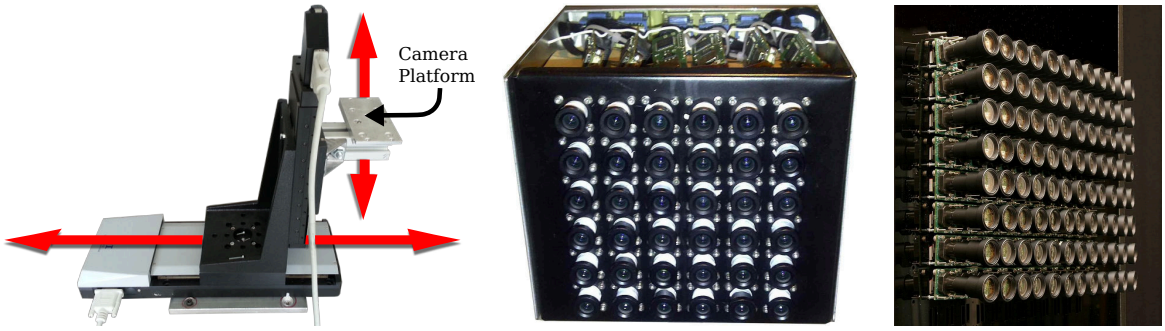
Another approach is from Georgiev et al. [33]. They define a sub window patch of a micro-image – similar to those depicted in figure 11 – and compute the cross correlation of this patch along the x-axis of the left and right neighbored micro-images as well as along the y-axis of the top and bottom neighbored micro-images. This results in a shift from one micro-image to the other. Knowing this shift and the chosen window size of the initial patch used for searching, an optimal patch size for this micro-image can be computed. By tiling all patches with optimal patch size together, a full depth-of-field view can be rendered.

2.4.3 Gantry

One of the most simple and inexpensive opportunities to sample a light field is a *gantry* (see figure 13 left). This is a precise xy-axis stepper motor driving a normal camera along a regular 2D grid. The benefits besides the simplicity are that baselines down to a millimeter are realizable and no color or optics correction over several cameras are necessary. A disadvantage is that only static scenes under static lighting conditions can be captured, and the mobility is restricted.

2.4.4 Camera Arrays

Due to the fact that the analysis in this work deals with epipolar plane images, the output of camera arrays is the most convenient data structure. They offer a fast and direct access to the 4D *Lumigraph* (compare section 2.2) and the EPIs (section 2.3). Additionally camera arrays are also suitable for capturing dynamic scenes. Another benefit is that a camera array not necessarily limits the spacial resolution as *Plenoptic Cameras* (sections 2.4.1 and 2.4.2) do. The main disadvantages are that they are costly due to the amount of cameras but also due to the hardware necessary for synchronization and data access. Additionally they also need a more complicated calibration process. Besides the external and internal calibration it is also important to apply a color and noise calibration due to the different sensor behavior of individual cameras.



(a) Gantry - xy stepper motor

(b) Two possible realizations of a camera array

Figure 13: (a) Depicts our precise *gantry* xy-axis stepper motor. Indicated are the x and y axis through the red arrows as well as the camera platform. Many thanks to the *Robert Bosch GmbH* for loaning this device. (b) Depicts on the left a prototype of a 6×6 array camera. Many thanks to Harlyn Baker for providing this image. Right side shows a camera array from Stanford [97].

2.4.5 Simulation

In this work, we often make use of simulated light fields. They offer, besides an inexpensive data generation, an easy access to interesting properties like ground truth for the geometry or the objects themselves, the material properties, as well as the opportunity to simulate the sensor's noise behavior. This makes simulation a great tool for algorithm development and for evaluation. We use the open source software *Blender* [77]. *Blender* offers an API accessible via *Python* [98]. This allows a scripting of all objects in the 3D environment. We simulate the light fields by scripting the blender camera to sample the 3D scene on a regular 2D grid. This is exactly the data format of a *gantry* (section 2.4.3) or a camera array (section 2.4.4) device. It should be noted her that with the rendered *Lumigraph* (compare section 2.2) and the ground truth depth provided through *Blender*, a simulation of *Plenoptic Camera 1.0* (section 2.4.1) data is also quite easy to achieve. Simulation of a *Focused Plenoptic Camera* (section 2.4.2) is not so trivial, and needs a more complex simulation of the optics.

Together with light fields captured using a *gantry* mentioned in section 2.4.3 we offer a benchmark database for light field analysis consisting of real world and simulated 4D light fields (see section 4 and figures 18 and 19).

3 Lumigraph Representation from Plenoptic Camera Images

In the following sections of this work, we will see that having access to the epipolar plane images (see section 2.3) of a light field can be very valuable for the analysis of a captured scene. But, if our light field is sampled with a *Plenoptic Camera 2.0* (see section 2.4.2), this access is not trivial.

Basically, the generation of a *Lumigraph* representation from a sampled 4D Light Field is simple – at least using camera arrays [108] (see also sections 2.3, 2.4.3 and 2.4.4) – where the projective transformations of the views of the individual cameras only have to be rectified and unified into one epipolar coordinate system requiring a precise calibration of all cameras.

Due to the optical properties of the micro-lenses – with the image plane of the main lens defining the epipolar coordinate system – these projective transformations are, in the case of *Focused Plenoptic Cameras*, reduced to simple translations [63] of the patches $\widehat{p}_{ij}$ within each micro-image, given by an offset $\vec{o}$ (see section 2.4.2). Hence, one simply has to rearrange the viewpoint-dependent rendered views from plenoptic raw data into the 4D EPI representation (see equation 9).

However, the necessarily small depth of field of the micro-lenses causes other problems. For most algorithms, the EPI structure can only be effectively evaluated in areas with high-frequency textures - which of course is only possible for parts of a scene which are in focus.

Another problem are different focal lengths of the micro-lenses the camera vendor uses to increase the depth of field [73]. One last, but also most important problem is, that *Focused Plenoptic Cameras* suffer from imaging artifacts in out-of-focus areas. Hence, in order to generate EPIs which can be used to analyze the entire scene at once, we have to generate the EPIs from all-in-focus (i.e. full depth-of-field) views for each focal length separately.

3.1 Rendering All In Focus Views Without Pixel-wise Depth

We already discussed some existing methods addressing the topic of rendering all-in-focus views in section 2.4.2. Now we discuss our contribution based on the publication [104].

To generate *plenoptic artifact* free *Lumigraph* (section 2.2, equation 9) from raw data of a *Focused Plenoptic Camera*, we need images of all available viewpoints without *plenoptic artifacts* and thus we need to render all full depth-of-field images from the raw data.

The primary objective in analyzing light fields is to reconstruct the inherently available depth information since further computations can benefit from available range data. Here we want to make use of the epipolar plane images (section 2.3) to reconstruct the scene geometry (section 5.1). Thus the generation of a *Lumigraph* from plenoptic raw data is a *chicken-and-egg-problem* because we already need the depth to render the all-in-focus views (section 2.4.2) and generate artifact free epipolar plane images.

The computation of full depth-of-field images from a series of views with different image planes usually requires depth information of the given scene: [15], [73] and [13] applied a depth estimation based on cross-correlation. The main disadvantage of this approach is that one would have to solve a major problem, namely the depth estimation for *non-Lambertian* scenes, in order to generate the EPI representation, which is intended to be used to solve the problem in the first place - as already mentioned a classical *chicken-and-egg-problem*. To overcome this dilemma, we propose an alternative approach. We actually do not need to determine the depths explicitly - all we need are the correct patch sizes δ_m to ensure a continuous view texturing without *plenoptic artifacts*.

We propose to find the best δ_m via a local minimization of the gradient magnitude at the patch borders $b(\widehat{p}_{ij})$ (see figure 11, section 2.4.2) over all possible focal images Ω_m. Since the effective patch resolution changes with δ_m, we have to apply a low-pass filtering to ensure a fair comparison. In practice, this is achieved by downscaling each patch to the smallest size δ_{min}, using bilinear interpolation. We denote the band-pass filtered focal images by $\bar{\Omega}_m$. Assuming a set of patch sizes $\vec{\delta} = [\delta_0, \ldots, \delta_m, \ldots, \delta_M]$, we render a set Γ of border images using a *Laplacian filter* (see figure 14):

$$\vec{\Gamma} = \left[\nabla^2\bar{\Omega}_0, \ldots, \nabla^2\bar{\Omega}_m, \ldots, \nabla^2\bar{\Omega}_M\right], \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{14}$$

From Γ, we determine the gradients for each hexagon patch by integrating along its borders $b(\widehat{p}_{ij})$, considering only gradient directions orthogonal to the edges of the patch (see figure 14). The norm of the gradients orthogonal to the border of each micro-image patch $\widehat{p}_{ij}$ and each image plane m is computed as

$$\Sigma\left(m, i, j\right) = \oint_{b(p_{ij})} \vec{n}_b \cdot \nabla\Gamma_m ds. \tag{15}$$

Here, $\vec{n}_b$ denotes the normal vector of each hexagon border b (compare figure 11). Furthermore, we define the lens specific image plane map $z\left[i, j\right]$ as a minimum of Σ_m for each micro-lens image (i, j)

$$z\left(i, j\right) = \underset{m}{argmin}\ \Sigma\left[m, i, j\right]. \tag{16}$$

The image plane map $z\left(i, j\right)$ has a resolution of (N_y, N_x) (number of micro-lenses) and encodes the patch size value δ_{ij} for each microimage (i, j). Using $z\left(i, j\right)$, we render full depth of field views $\Omega\left(z\right)$. This approach works nicely for all textured regions of

a scene. Evaluating the standard deviation of Σ for each (i, j) can serve to further improve $z\,(i, j)$: micro-images without (or with very little) texture are characterized by a small standard deviation in Σ. We use a threshold to replace the affected $z\,(i, j)$ with the maximum patch size δ_{max}. This is a valid approach, since the patch size (i.e. the focal length) does not matter for untextured regions. Additionally, we apply a gentle median filter to remove outliers from $z\,(i, j)$.

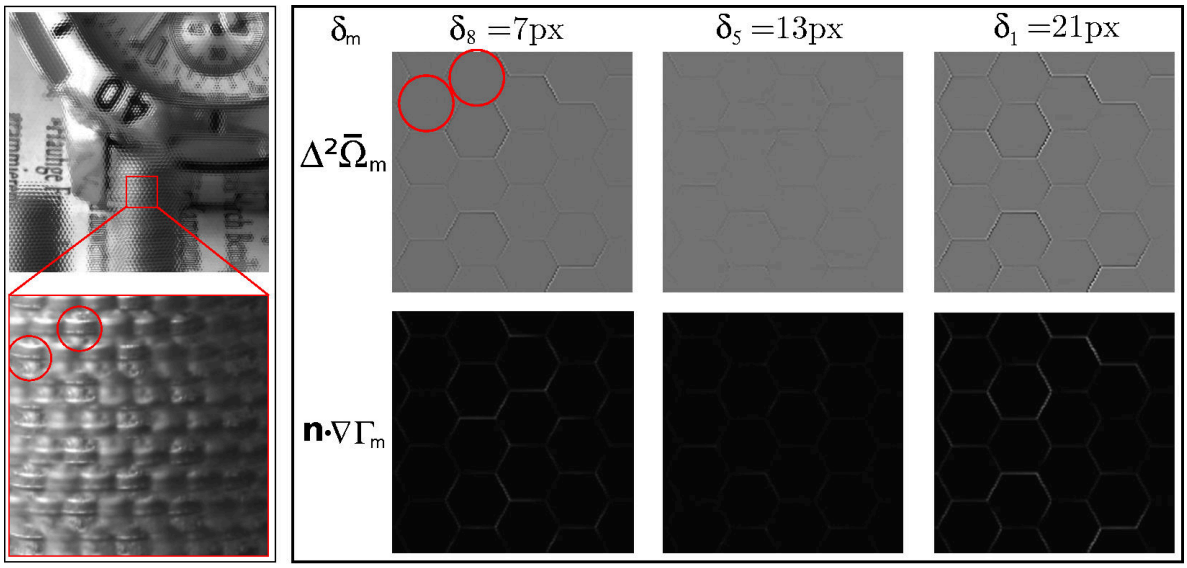


Figure 14: Left: Part of a raw data image from a *Focused Plenoptic Camera* and a zoomed part of it. **Right:** Three examples of the border image set (see eq. 14) and the gradient magnitude set (equation. 15) with different patch sizes δ are depicted. The example in the center shows the correct focal length.

3.2 Merging Views from Different Lens Types

The full depth-of-field views for each lens type have the same angular distribution (if the same offsets $\vec{o_q}$ have been used), but are translated relative to each other. We neglect that these translations are not completely independent of the depth of the scene. Due to the very small offset (baseline), theses effects are in the order of sub-pixel fractions. The results shown in figures 6, 7 and 8 are merged by determining the relative shifts T_n via normalized cross-correlation and averaging over the views with the same offset.

$$\Omega_{merged}\,(z, \vec{o_q}) = \frac{1}{3}\sum_{n=1}^{3} T_n \Omega_n\,(z, \vec{o_q}) \quad T_n \in \mathbb{N} \times \mathbb{N} \tag{17}$$

Due to the fact that each lens type has an individual focal length, the sharpness of the results can be improved by a weighted averaging depending on the optimal focal range

of each lens type and the information from the focal maps $z[i,j]$

$$\Omega_{merged}\left(z,\vec{o}_q\right)=\sum_{n=1}^{3}\alpha_n\left(z\right)T_n\Omega_n\left(z,\vec{o}_q\right),\quad \alpha_n\in\mathbb{R}\quad\text{and}\quad\sum_{n=1}^{3}\alpha_n=1. \tag{18}$$

3.3 The EPI Generation Pipeline

1. **View rendering:** Rendering of all possible full depth-of-field images $\Omega(z,\vec{o})$ for different view points of the scene using the focal plane map $z\left(i,j\right)$ of optimal patch sizes and the patch offset vector $\vec{o}$.

2. **View merging:** Merging of the corresponding views of different lens types. This step is only necessary for cameras with several micro-lens types, such as the camera used in our experiments [72].

3. **View stacking:** After the merging process, a single set of rendered views remains. These have to be arranged in a 4D volume according to their view angles resulting in the EPI structure $L(x,y,s,t)$ (section 2.2, equation 9).

3.4 Results

For the experimental evaluation, we use a commercially available *Focused Plenoptic Camera* (the *R11* by the camera manufacturer *Raytrix GmbH* [72]). The camera captures raw images with a resolution of 10 Mega-pixels and is equipped with an array of roughly 11000 micro-lenses. The effective micro-image diameter is 23 pixels. The array holds three types of lenses with different focal lengths, nested in a $3\times65\times57$ hexagon layout, which leads to an effective maximum resolution of 1495×1311 pixels for rendered projective views at the focal length of the main lens. Due to this setup with different micro-lens types, we compute the full depth of field view for each lens type independently and then apply a merging algorithm.

A qualitative evaluation is shown in figure 15. We compare the results of our proposed algorithm with the output of commercial software from the camera vendor, which computes the full depth of field projective views via an explicit depth estimation based on stereo matching on the camera raw data [72]. We present the raw output of both methods. It should be noted, that the results of the depth estimates are not directly comparable - the emphasis of our qualitative evaluation lies in the full depth of scene reconstruction.

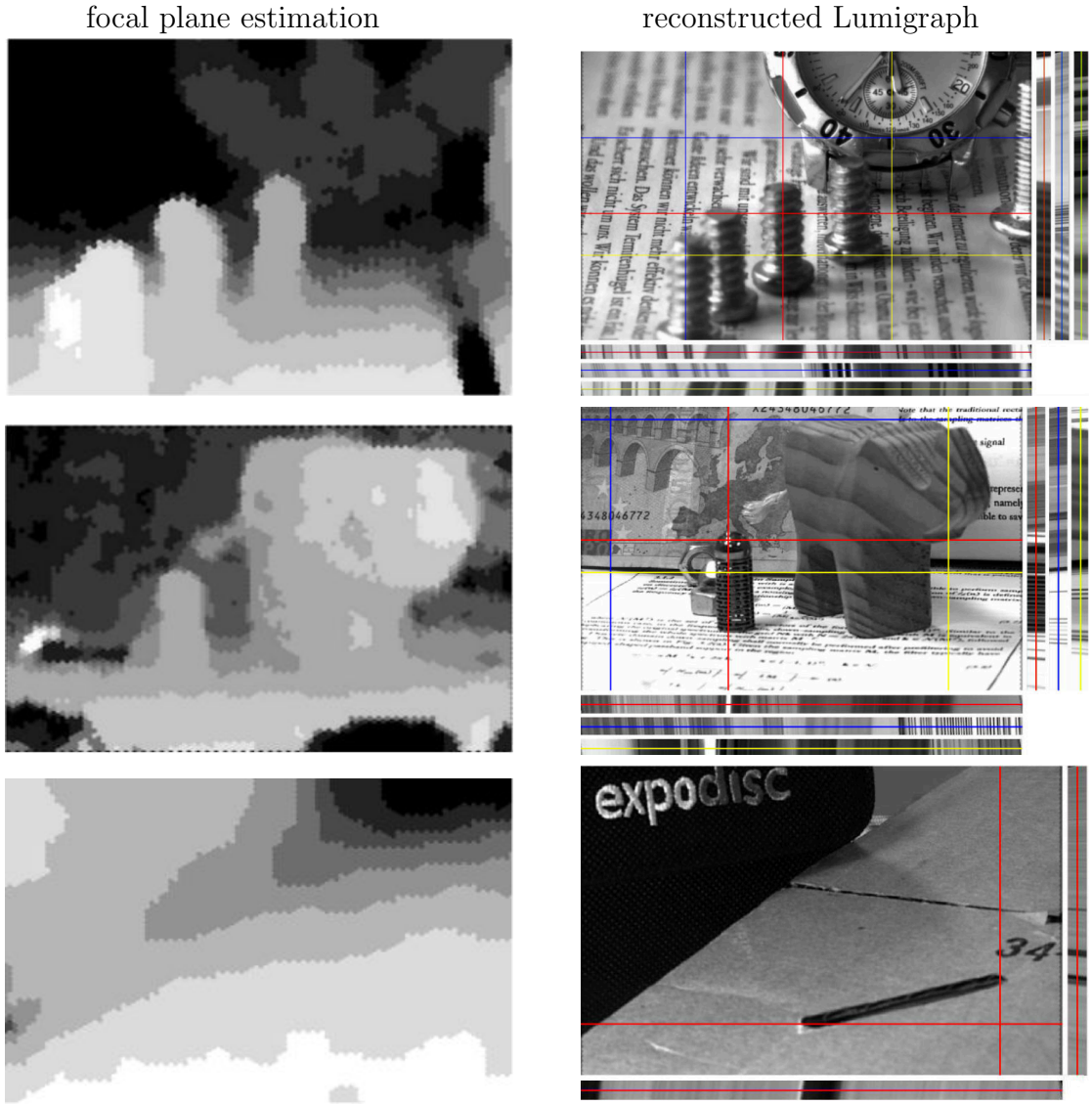


Figure 15: Estimation of the focal length. The left side shows a typically dense image plane map $z\,[i,j]$ (see equation 16), computed with our algorithm. On the right, the center views of the reconstructed all-in-focus *Lumigraph* as well as exemplary extracted epipolar plane images are depicted.

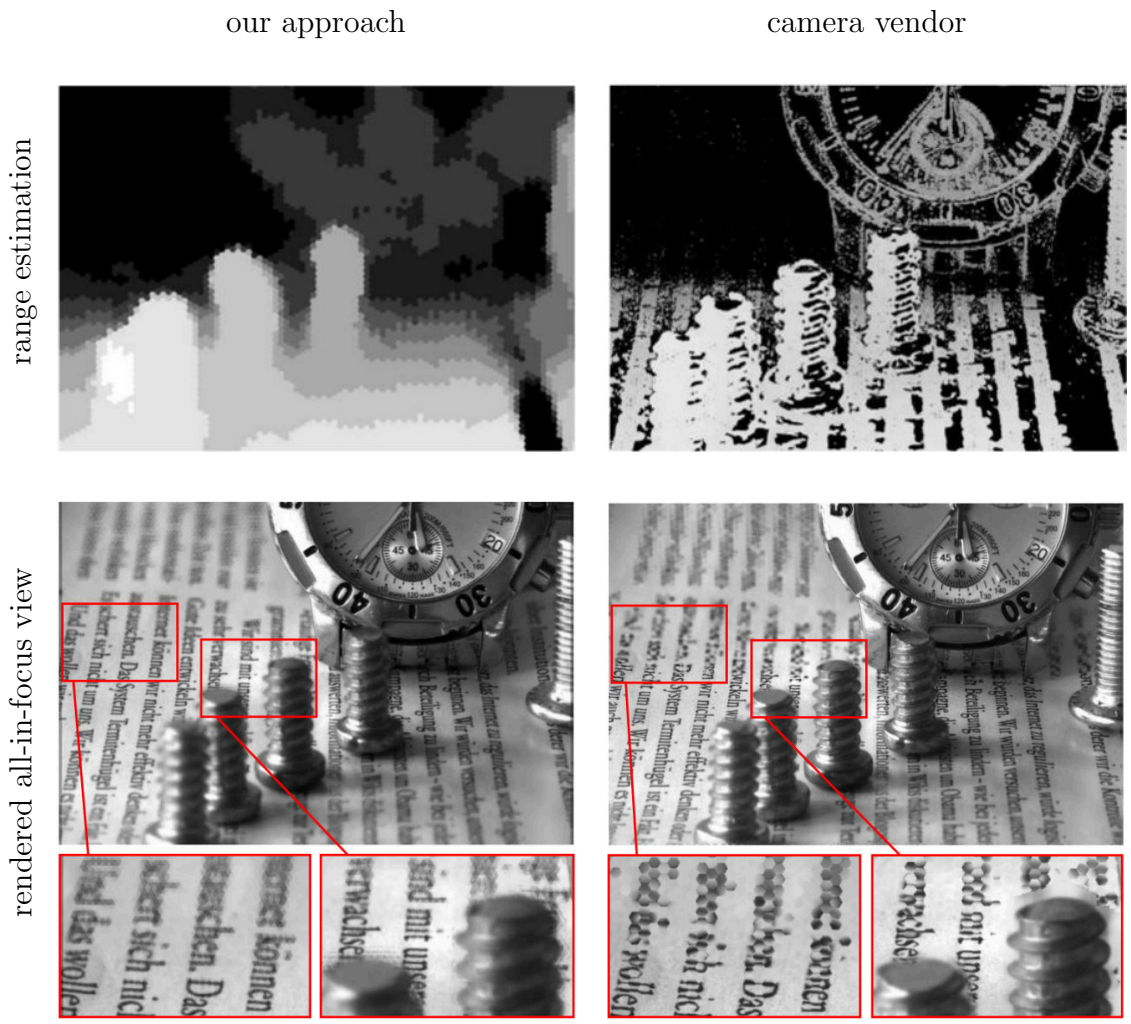


Figure 16: Top row: Focal plane reconstruction vs. the *stereo-based* depth reconstruction of the camera vendor [72]. Bottom row: the all-in-focus rendering of: (left) The proposed method and (right) the stereo-based method of the camera vendor.

4 Data Sets - The 4D Light Field Archive

The driving force for successful algorithm development is the availability of suitable benchmark datasets with ground truth data in order to compare results and initiate competition. Light field datasets and, in particular, the type of light fields used in this work – namely dense sampled 4D *Lumigraphs* (see section 2.2) – are not yet widely deployed. There are a few but none of the existing fulfill all of our needs and thus we decided to establish a new benchmark database.

The current public light field databases we are aware of are the following.

- **Stanford Light Field Archive**
  `http://lightfield.stanford.edu/lfs.html`
  The Stanford Archives provide more than 20 light fields sampled using a camera array [109], a gantry and a light field microscope [60], but none of the datasets includes ground truth disparities.

- **UCSD/MERL Light Field Repository**
  `http://vision.ucsd.edu/datasets/lfarchive/lfs.shtml`
  This light field repository [47] offers video as well as static light fields, but there is also no ground truth depth available, and the light fields are sampled in a one-dimensional domain of view points only.

- **Synthetic Light Field Archive**
  `http://web.media.mit.edu/~gordonw/SyntheticLightFields/index.php`
  The synthetic light field archive [66] provides many interesting artificial light fields including some nice challenges like transparencies, occlusions and reflections. Unfortunately, there is also no ground truth depth data available for benchmarking.

- **Middlebury Stereo Datasets**
  `http://vision.middlebury.edu/stereo/data/`
  The Middlebury Stereo Dataset [43, 82, 83, 84] includes a single 4D light field which provides ground truth data for the center view, as well as some additional 3D light fields including depth information for two out of seven views. The main issue with the Middlebury light fields are that they are designed with stereo matching in mind, thus the baselines are quite large and not representative for compact light field cameras and unsuitable for direct epipolar plane image analysis.

While there is a lot of variety and the data is of high quality, we observe that all of the available light field databases either lack ground truth disparity information or exhibit large camera baselines and disparities, which is not representative for compact light field cameras like i.e. *Plenoptic Cameras*. Furthermore, we believe that a large part of what distinguishes light fields from standard multi-view images is the ability to treat the view point space as a continuous domain. There is also emerging interest in light field segmentation [31, 50, 92, 106], so it would be highly useful to have ground truth segmentation data available to compare light field labeling schemes. The above data

sets lack this information as well.

To alleviate the above shortcomings, we present a new benchmark database which consists at the moment of 13 high quality densely sampled light fields. The database offers seven computer graphics generated data sets providing complete ground truth disparity for all views. Four of these data sets also come with ground truth segmentation information and pre-computed local labeling cost functions to compare global light field labeling schemes. Furthermore, there are six real world data sets captured using a single Nikon D800 camera mounted on a *gantry*. Using this device, we sampled objects which were pre-scanned with a structured light scanner to provide ground truth ranges for the center view. An interesting special data set contains a transparent surface with ground truth disparity for both the surface as well as the object behind it - we believe it is the first real-world data set of this kind with ground truth depth available.

4.1 The Light Field Archive

Our light field archive (`www.lightfield-analysis.net`) is split into two main categories, *Blender* and *Gantry*. The *Blender* category consists of seven scenes rendered using the open source software *Blender* [77] and our own light field plug-in, see figure 18 for an overview of the data sets. The *Gantry* category provides six real-world light fields captured with a commercially available standard camera mounted on a *gantry* device, see figure 19. More information about all the data sets can be found in the overview in figure 17.

Each data set is split into different files in the *HDF5-format* [95], exactly which of these are present depends on the available information. Common to all data sets is a main file called **lf.h5**, which contains the light field itself and the range data. In the following, we will explain its content as well as that of the different additional files, which can be specific to the category.

4.1.1 The Main File

The main file **lf.h5** for each scene consists of the actual light field image data as well as the ground truth depth, see figure 17. Each light field is 4D, and sampled on a regular grid. All images have the same size, and views are spaced equidistantly in horizontal and vertical directions, respectively. The general properties of the light field can be accessed in the following attributes:

dataset name	category	resolution	GTD	GTL
buddha	Blender	768x768x3	full	yes
horses	Blender	576x1024x3	full	yes
papillon	Blender	768x768x3	full	yes
stillLife	Blender	768x768x3	full	yes
buddha2	Blender	768x768x3	full	no
medieval	Blender	720x1024x3	full	no
monasRoom	Blender	768x768x3	full	no
couple	Gantry	898x898x3	cv	no
cube	Gantry	898x898x3	cv	no
maria	Gantry	926x926x3	cv	no
pyramide	Gantry	898x898x3	cv	no
statue	Gantry	898x898x3	cv	no
transparency	Gantry	926x926x3	2xcv	no

Figure 17: *Overview of the datasets in the benchmark.* **dataset name**: The name of the dataset. **category**: *Blender* (rendered synthetic dataset) or *Gantry* (real-world dataset sampled using a single moving camera). **resolution**: spatial resolution of the views, all light fields consist of 9x9 views. **GTD**: indicates completeness of ground truth depth data, either cv (only center view) or full (all views). A special case is the transparency dataset, which contains ground truth depth for both background and transparent surface. **GTL**: indicates if object segmentation data is available.

HDF5 attribute	description
yRes	height of the images in pixel
xRes	width of the images in pixel
vRes	# of images in vertical direction
hRes	# of images horizontal direction
channels	light field is rgb (3) or grayscale (1)
vSampling	rel. camera position grid vertical
hSampling	rel. camera position grid horizontal

The actual data is contained in two HDF5 data sets:

HDF5 dataset	size
LF	*vRes* x *hRes* x *xRes* x *yRes* x *channels*
GT_DEPTH	*vRes* x *hRes* x *xRes* x *yRes*

These store the separate images in RGB or gray-scale (range 0-255), as well as the associated depth maps, respectively.

Conversion between depth and disparity. To compare disparity results to the ground truth depth, the latter has to first be converted to disparity. Given a depth Z, the disparity or slope of the epipolar lines d in pixels per grid unit is

$$d = \frac{B * f}{Z} - \Delta x, \tag{19}$$

where B is the baseline or distance between two cameras, f the focal length in pixel and Δx the shift between two neighboring images relative to an arbitrary rectification plane (in case of light fields generated with Blender, this is the scene origin). The parameters in equation 19 are given by the following attributes in the main HDF file:

	attribute	**description**
B	*dH*	distance between to cameras
f	*focalLength*	focal length
Δx	*shift*	shift between neighboring images

The following sections describe differences and conventions about the depth scale for the two current categories.

4.1.2 Blender Category

The computer graphics generated scenes consist without exception of ground truth depth over the entire light field. This information is given as orthogonal distance of the 3D point to the image plane of the camera, measured in *Blender* units $[BE]$. The *Blender* main files have an additional attribute *camDistance* which is the base distance of the camera to the origin of the 3D scene, and used for the conversion to disparity values.

Conversion between Blender depth units and disparity. The above *HDF5* camera attributes in the main file for conversion from *Blender* depth units to disparity are calculated from *Blender* parameters via

$$\begin{aligned} dH &= b * xRes, \\ focalLength &= 1/\left(2 * tan\left(\frac{fov}{2}\right)\right), \\ shift &= \frac{1}{\left(2 * Z_0 * tan\left(\frac{fov}{2}\right)\right) * b}, \end{aligned} \tag{20}$$

where Z_0 is the distance between the *Blender* camera and the scene origin in $[BE]$, fov is the field of view in units radian and b the distance between two cameras in $[BE]$. Since all light fields are rendered or captured on a regular equidistant grid, it is sufficient to use only the horizontal distance between two cameras to define the baseline.

4.1.3 Segmentation Ground Truth

Some light fields have segmentation ground truth data available, see figure 17, and offer five additional *HDF5* files:

- **labels.h5**:
  This file contains the *HDF5* dataset *GT_LABELS* which is the segmentation ground truth for all views of the light field and the *HDF5* dataset *SCRIBBLES* which are user scribbles on a single view.

- **edge_weights.h5**:
  Contains an *HDF5* data set called *EDGE_WEIGHTS* which are probabilities for edges [106] for all views. These are not only useful for segmentation, but any algorithm which might require edge information, and can help with comparability since all of these can use the same reference edge weights.

- **feature_single_view_probabilities.h5**:
  The *HDF5* data set *Probabilities* contains the prediction of a random forest classifier trained on a single view of the light field without using any feature requiring light field information [106].

- **feature_depth_probabilities.h5**:
  The *HDF5* data set *Probabilities* contains the prediction of a random forest classifier trained on a single view of the light field using estimated disparity [100] as an additional feature [106].

- **feature_gt_depth_probabilities.h5**:
  The *HDF5* data set *Probabilities* contains the prediction of a random forest classifier trained on a single view of the light field using ground truth disparity as an additional feature [106].

4.1.4 Gantry category

In the *Gantry* category, each scene always provides a single main **lf.h5** file, which contains an additional *HDF5* data set *GT_DEPTH_MASK*. This is a binary mask indicating valid regions in the ground truth *GT_DEPTH*. Invalid regions in the ground truth disparity have mainly two causes. First, there might be objects in the scene for which no 3D data is available, and second, there are parts of the mesh not covered by the structured light scan and thus having unknown geometry. See section 4.2.2 for details.

A special case is the light field *transparency*, which has two depth channels for a transparent surface and an object behind it, respectively. Therefore, there also exist two mask *HDF5* data sets, see figure 20. We believe this is the first benchmark light field for multi-channel disparity estimation.

Here, the *HDF5* data sets are named:

- *GT_DEPTH_FOREGROUND*,
- *GT_DEPTH_BACKGROUND*,
- *GT_DEPTH_FOREGROUND_MASK*,
- *GT_DEPTH_BACKGROUND_MASK*.

4.2 Generation of the light fields

The process of light field sampling is very similar for both synthetic as well as real world scenes. The camera is moved on an equidistant grid parallel to its own sensor plane and an image is taken at each grid position. Although not strictly necessary, an odd number of grid positions is used for each movement direction as there then exists a well-defined center view which makes the processing simpler. An epipolar rectification on all images is performed to align individual views to the center one. The source for the internal and external camera matrices needed for this rectification depends on the capturing system used.

4.2.1 Blender category

For the synthetic scenes, the camera can be moved using a script for the *Blender* engine. As camera parameters can be set arbitrarily and the sensor and movement plane coincide perfectly, no explicit camera calibration is necessary. Instead, the values required for rectification can be derived directly from the internal *Blender* settings.

4.2.2 Gantry category

For real-world light fields, a *Nikon D800* digital camera is mounted on a stepper-motor driven gantry manufactured by *Physical Instruments*. A picture of the setup can be seen in figure 13. Accuracy and repositioning error of the *gantry* is well in the micrometer range. The capturing time for a complete light field depends on the number of images, about 15 seconds are required per image. As a consequence, this acquisition method is limited to static scenes. The internal camera matrix must be estimated beforehand by capturing images of a calibration pattern and invoking the camera calibration algorithms of the *OpenCV library* [17], (see next section for details). Experiments have shown that the positioning accuracy of the *gantry* actually surpasses the pattern based external calibration as long as the differences between the sensor and movement planes are kept minimal.

Ground Truth for the Gantry Light Fields. This section is the work of Stephan Meister [68]. Ground truth for the real world scenes was generated using standard pose estimation techniques. First, we acquired 3D polygon meshes for an object in the scene using a *Breuckmann SmartscanHE structured light scanner*. The meshes contain between 2.5 and 8 Million faces with a stated accuracy of down to 50 micron.

The object-to-camera pose was estimated by hand-picking 2D-to-3D feature points from the light field center view and the 3D mesh, and then calculating the external camera matrix using an iterative *Levenberg-Marquardt* approach from the *OpenCV* library [17]. This method is used for both the internal and external calibration. An example set of correspondence points for the scene *pyramide* can be observed in figure 21.

The re-projection error for all scenes was typically 0.5 ± 0.1 pixels. The depth is then defined as the distance between the sensor plane and the mesh surface visible in each pixel. The depth projections are computed by importing the mesh and measured camera parameters into Blender and performing a depth rendering pass. At depth discontinuities (edges) or due to the fact that the meshes' point density is higher than the lateral resolution of the camera, one pixel can contain multiple depth cues. In the former case, the pixel was masked out as an invalid edge pixel and, in the latter case, the depth of the polygon with the biggest area inside the pixel was selected. The error is generally negligible as the geometry of the objects is sufficiently smooth at these scales. Smaller regions where the mesh contained holes were also masked out and not considered for the final evaluations.

For an accuracy estimation of the acquired ground truth, we perform a simple error propagation on the projected point coordinates. Given an internal camera matrix C and an external matrix R, a 3D point $\vec{P} = (X, Y, Z, 1)$ is projected onto the sensor pixel $(u\ v)$ according to

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = C\ R\ \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

For simplicity, we assume that the camera and object coordinate systems coincide, save for an offset t_z along the optical axis. Given focal length f_x, principal point c_x and re-projection error Δu, this yields for a pixel on the $v = 0$ scan-line

$$t_z = Z - \frac{f_x X}{u - c_x},$$

resulting in a depth error Δt_z of

$$\Delta t_z = \frac{\partial t_z}{\partial u} \Delta u = \frac{f_x X}{(c_x - u)^2} \Delta u.$$

Calculations for pixels outside of the center-scan line are performed analogously. The error estimate above depends on the distance of the pixel from the camera's principal point. As the observed objects are rigid, we assume that the distance error Δt_z between camera and object corresponds to the minimum observed Δt_z among the selected 2D-3D correspondences. For all gantry scenes, this value is in the range of $1mm$ so we assume this to be the approximate accuracy of our ground truth.

Figure 18: Data sets in the category Blender. Top: Light fields with segmentation information available. From left to right: *buddha*, *papillon*, *stillLife*, *horses*. First row shows center view, second depth ground truth and third label ground truth. Bottom: Light fields without segmentation information. From left to right: *buddha2*, *monasRoom*, *medieval*. First row shows center view, the second the depth ground truth.

Figure 19: Data sets in the category Gantry. From left to right: center view, depth channel, mask which indicates regions with valid depth information. The ordering of the data sets is the same as in figure 17.

Figure 20: Data set *transparency*. Left: center view, middle top: depth of the background, middle bottom: depth of the foreground, right top: background mask for valid depth ground truth pixel, right bottom: foreground mask for valid depth ground truth pixel.

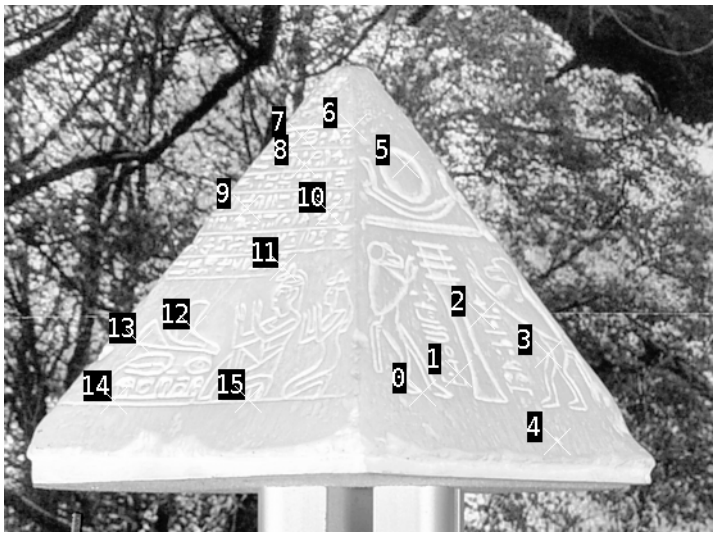


Figure 21: Selected 2D correspondences for pose estimation for the *pyramide* dataset. In theory, four points are sufficient to estimate the six degrees of freedom of an external camera calibration matrix, but more points increase the accuracy in case of outliers.

5 Orientation Analysis in Light Fields

5.1 Single Orientation Analysis

A main benefit of light fields compared to traditional images or stereo pairs is the expansion of the disparity space to a continuous space. This becomes apparent when considering epipolar plane images (section 2.3), which can be viewed as 2D slices of constant angular and spatial direction through the *Lumigraph* (section 2.2). Due to a dense sampling in angular direction, corresponding pixels are projected onto lines in EPIs, which can be detected more robustly and faster than point correspondences.

EPIs were introduced to the analysis of scene geometry by Bolles et al. [16]. They detect edges, peaks and troughs with a subsequent line fitting in the EPI to reconstruct 3D structure. Later, Baker used zero crossings of the *Laplacian* [6, 7]. Another approach is presented by Criminisi [27], who use an iterative extraction procedure for collections of EPI-lines of the same depth, which they call an EPI-tube. Lines belonging to the same tube are detected via shearing the EPI and analyzing photo-consistency in the vertical direction. They also propose a procedure to remove specular highlights from already extracted EPI-tubes.

There are also two less heuristic methods which work in an energy minimization framework. In Matousek et al. [67], a cost function is formulated to minimize a weighted path length between points in the first and the last row of an EPI, preferring constant intensity in a small neighborhood of each EPI-line. However, their method only works in the absence of occlusions.

Berent et al. [11] deal with the simultaneous segmentation of EPI-tubes by a region competition method using active contours, imposing geometric properties to enforce correct occlusion ordering.

In contrast to the above works, we propose a local gradient based orientation analysis of the EPIs and additionally can perform a labeling for all points in the EPI simultaneously by using a state-of-the-art continuous convex energy minimization framework. We enforce globally consistent visibility across views by restricting the spatial layout of the labeled regions.

Compared to methods of Bolles [16] and Criminisi [27] which extract EPI information sequentially, this is independent of the order of extraction and does not suffer from an associated propagation of errors. While a simultaneous extraction is also performed by Berent et al. [11], they perform local minimization only and require good initialization, as opposed to our convex relaxation approach. Furthermore, they use a level set approach, which makes it expensive and cumbersome to deal with a large number of regions.

In this section we propose a range estimation approach using a 4D light field parametrized

as *Lumigraph* (see section 2.2 and equation 9). The basic idea is as follows. We first compute local slope estimates on epipolar plane images for the two different slice directions (x, s)-slice Σ_{y^*,t^*}, (y, t)-slice $\Sigma_{x^*s^*}$ (section 2.3) using the structure tensor (section 5.1.1). This gives two local disparity estimates for each pixel in each view. These can be merged into a single disparity map in different ways: just locally choosing the estimate with the higher reliability, optionally smoothing the result (which is very fast), or solving a global optimization problem (which is slow). In the experiments, we will show that, fortunately, the fast approach leads to estimates which are slightly more accurate. The content in this section is published in Wanner et al. [100], [103] whereby the theory as well as fast GPU implementations [37] of the optimization techniques are the work of Bastian Goldlücke

5.1.1 The Structure Tensor

A common technique to estimate orientations is the structure tensor introduced by Bigun et al. [12]. Derivations below follow the chapter "The Structure Tensor" in Jähne [45].

If we assume a unit vector $\mathbf{n} \in \mathbb{R}^D$ as the preferred local orientation of the gray value changes of a function $g : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^D$, the following must be satisfied:

$$\left(\nabla g^T \mathbf{n}\right)^2 = |\nabla g|^2 cos^2 \left(\sphericalangle \left(\nabla g, \mathbf{n}\right)\right). \tag{21}$$

This will become a maximum if ∇g is parallel to $\mathbf{n}$ or if ∇g is anti-parallel to $\mathbf{n}$ and zero if ∇g is orthogonal to $\mathbf{n}$. Thus we need to maximize the following expression in a local environment

$$\int w(\mathbf{x} - \mathbf{x}') \left(\nabla g(\mathbf{x}')^T \mathbf{n}\right)^2 d^D x', \tag{22}$$

where w is a window function determining size and shape of the average region around $\mathbf{x}$. Equation (22) can be reformulated to

$$\mathbf{n}\boldsymbol{J}\mathbf{n} \to \text{maximum} \tag{23}$$

$$\boldsymbol{J} = \int w(\mathbf{x} - \mathbf{x}')(\nabla g(\mathbf{x}') \nabla g(\mathbf{x}')^T) d^D x'. \tag{24}$$

which results in a symmetric $D \times D$ tensor

$$J_{pq}(\mathbf{x}) = \int_{-\infty}^{\infty} w(\mathbf{x} - \mathbf{x}') \left(\frac{\partial g(\mathbf{x}')}{\partial x'_p} \frac{\partial g(\mathbf{x}')}{\partial x'_q}\right) d^D x'. \tag{25}$$

An Eigenvalue decomposition of $\boldsymbol{J}$, in case of $D = 2$, gives two Eigenvalues λ_1,λ_2. Without limiting the generality, we assume that $\lambda_1 > \lambda_2$. Due to the orthogonality of

condition	rank	meaning
$\lambda_1 = \lambda_2 = 0$	0	const. local environment
$\lambda_1 > 0, \lambda_2 = 0$	1	ideal local orientation
$\lambda_1 > 0, \lambda_2 > 0$	2	isotropic environment

Table 1: The table shows the meaning of different Eigenvalue conditions of the structure tensor for 2D images

the Eigenvectors $\mathbf{v}_1$,$\mathbf{v}_2$ it is obvious that $\mathbf{v}_1$ is parallel to $\mathbf{n}$ and $\mathbf{v}_2$ is anti-parallel to $\mathbf{n}$.

The relationship between the Eigenvalues give a quality measure of the local orientation pattern. Jähne [45] defines the coherence c, which varies between zero for isotropic structures and one for ideal orientations:

$$c = \frac{\sqrt{(J_{11} - J_{22})^2 + 4J_{12}^2}}{J_{11} + J_{22}} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}, \quad c \in [0, 1] \tag{26}$$

Implementation. In general the computation of the structure tensor consists of four steps. An initial (*Gaussian*) smoothing to reduce noise and high frequencies, the gradient computation, the computation of the structure tensor components, and a final (*Gaussian*) smoothing of these components. A widely used approach to compute the gradients is the so-called *Sobel*-operator [75].

$$S_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}. \tag{27}$$

Another approach to compute the gradients is the *Scharr*-operator [81].

$$S_x = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}, S_y = \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}. \tag{28}$$

Scharr optimized the filter coefficients to guarantee an optimal rotational symmetry, leading to much better orientation estimations compared to the *Sobel*-operator. In this work we use a variant of the structure tensor combining the initial smoothing step and the gradient computation using a *Gaussian* derivative filter as implemented in the *VIGRA Computer Vision Library* [49]. We discuss the reason for this choice in section 5.1.2.2 and figure 23.

The definition of *Gaussian* filter is as follows

$$G_{\sigma,n}(\mathbf{x}) = \frac{\partial^n}{\partial x_1^{n_x} \partial x_2^{n_y}} \frac{1}{2\pi\sigma^2} e^{-\frac{x_1^2 + x_2^2}{2\sigma^2}} \tag{29}$$

whereas $n = n_x + n_y$ is the order of derivative and $\sigma > 0$ the standard deviation of the Gaussian. The kernel radius r_K is computed through

$$r_K = 3\sigma + \frac{1}{2}n. \tag{30}$$

The result is rounded to the next higher integer. The Kernel size δ_K then is

$$\delta_K = 2r_K + 1. \tag{31}$$

The gradient of an image I is defined as

$$\nabla I = (S_{x,\sigma}, S_{y,\sigma}) = \begin{pmatrix} G_{\sigma,1}(I)|n_x = 1 \\ G_{\sigma,1}(I)|n_y = 1 \end{pmatrix}. \tag{32}$$

The algorithm to compute the structure Tensor $J_{\sigma,\rho}$ on an gray-scale image I is then as follows:

1. compute the gradients $S_{x,\rho}, S_{y,\rho}$
2. compute the structure tensor components

$$J_{\rho,\sigma}(I) = \begin{pmatrix} G_{\sigma,0}(S_{x,\rho}S_{x,\rho}) & G_{\sigma,0}(S_{x,\rho}S_{y,\rho}) \\ G_{\sigma,0}(S_{y,\rho}S_{x,\rho}) & G_{\sigma,0}(S_{y,\rho}S_{y,\rho}) \end{pmatrix} \tag{33}$$

5.1.2 Disparities On Epipolar Plane Images

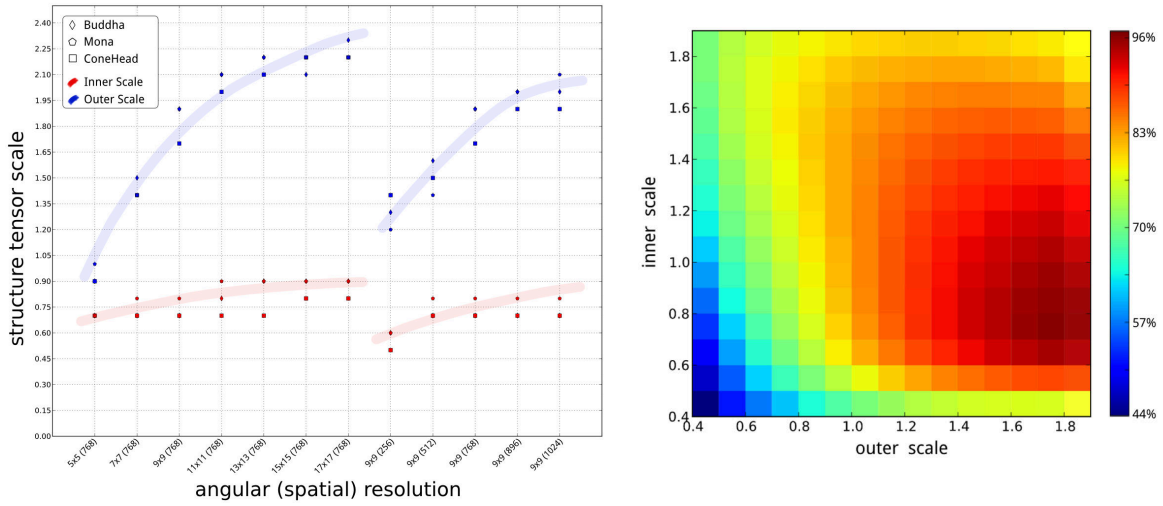
5.1.2.1 Local Disparity Estimation We first consider how we can estimate the local direction of a line at a point (x, s) in an epipolar plane image S_{y^*,t^*} (see section 2.3), where y^* and t^* are fixed. The case of vertical slices is analogous. The goal of this step is to compute a local disparity estimate $d_{y^*,t^*}(x, s)$ for each point of the slice domain, as well as a reliability estimate $r_{y^*,t^*}(x, s) \in [0, 1]$ (eq. 38), which is the coherence of the structure tensor (eq. 26) and gives a measure of how reliable the local disparity estimate is. Both local estimates will be used in subsequent sections to obtain a consistent disparity map in a global optimization framework.

In order to obtain the local disparity estimate, we need to estimate the direction of lines on the slice. This is done using the structure tensor J (see eq. 33) of the epipolar plane image $S = S_{y^*,t^*}$,

$$J_{\rho,\sigma}(S) = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix}. \tag{34}$$

The direction of the local level lines can then be computed via Bigun et al. [12]

$$\mathbf{n}_{y^*,t^*} = \begin{bmatrix} \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} \sin(\frac{1}{2}\arctan\left(\frac{J_{yy}-J_{xx}}{2J_{xy}}\right)) \\ \cos(\frac{1}{2}\arctan\left(\frac{J_{yy}-J_{xx}}{2J_{xy}}\right)) \end{bmatrix}, \tag{35}$$



(a) optimal structure tensor parameters

(b) grid search on dataset buddha

Figure 22: Using grid search, we find the ideal structure tensor parameters over a range of both angular and spatial resolutions (a). Blue colored data points show the optimal outer scale, red points the optimal inner scale. The thick streaks are added only for visual orientation. In (b) an example of a single grid search is depicted. Colour-coded is the amount of pixels with a relative error to the ground truth of less than 1%, which is the target value to be optimized for in (a).

from which we derive the local depth estimate via

$$Z = -f\frac{\Delta s}{\Delta x}. \tag{36}$$

Frequently, a more convenient unit is the disparity

$$d_{y^*,t^*} = \frac{f}{Z} = \frac{\Delta x}{\Delta s} = \tan\left(\frac{1}{2}\arctan\left(\frac{J_{yy} - J_{xx}}{2J_{xy}}\right)\right), \tag{37}$$

which describes the pixel shift of a scene point when moving between the views. We will usually use disparity instead of depth in the remainder of this work. According to equation 26 as the natural reliability measure we use the coherence of the structure tensor

$$r_{y^*,t^*} := \frac{\sqrt{(J_{yy} - J_{xx})^2 + 4J_{xy}^2}}{(J_{xx} + J_{yy})}. \tag{38}$$

Using the local disparity estimates d_{y^*,t^*}, d_{x^*,s^*} and reliability estimates r_{y^*,t^*}, r_{x^*,s^*} for all the EPIs in horizontal and vertical directions, respectively, one can now proceed to directly compute disparity maps in a global optimization framework, which is explained in section 5.1.3.2. However, it is possible to first enforce global visibility constraints separately on each of the EPIs, which we explain in section 5.1.2.3.

5.1.2.2 Limits of the Local Orientation Estimation In the following, we will perform a detailed evaluation of the orientation analysis by applying the structure tensor on synthetically generated epipolar planes. These EPIs are initialized with random stripe patterns of parallel lines. Random in this context means that we vary the stripe thickness and the assigned gray-scale to simulate the structure of real epipolar plane images. Below we list the parameters to control the generated EPI appearance in our experiments.

- h: height or number of pixels in y direction representing the number of cameras.
- d: the pixel shift or slope of the epipolar lines, equivalent to the disparity in real light fields. The EPIs are initialized with a disparity of zeros which can be changed by applying affine transformations with sub-pixel accuracy simulating a refocusing or change in depth.
- σ_n: the noise level. We add random Gaussian noise with a standard deviation of σ_n[px] to the images.
- w_{max}: maximum width of the epipolar lines, whereby width means the number of pixels of a line in the x-direction having assigned the same intensity value. This simulates low- or non-textured regions in the image domain.
- δ_c: the color variance. Epipolar lines have a random intensity value of $128 \pm \delta_c$ simulating low contrasts.

The general procedure of the experiments is:

1. generate an EPI with a random stripe pattern with respect to the parameters described above.
2. evaluate orientation on the EPI generated in step 1.
3. extract the estimated disparity from the center row $h/2$ (neglecting 10% of the pixels at the left and right borders to avoid border artifacts) and calculate the mean over the remaining pixels.
4. change the disparity d of the EPI by applying sub-pixel shifts on each row of the epipolar plane image
5. compute the disparity deviation δd between the evaluated d_m and the ground truth disparity d: $\delta d = d - d_m$.
6. repeat steps 2 to 4 over the desired parameter range evaluated in the experiment.
7. repeat steps 1 to 5 a number of N times and return the mean disparity deviation to achieve statistically reliable results over N randomly generated orientation patterns. A value of $N = 200$ showed to be suitable to stabilize the results. This step is included in all of our experiments without explicitly being mentioned every time.

Our goal with these experiments is to show the ideal behavior and theoretical abilities of the orientation estimation on synthetic epipolar plane images. The results do not allow to draw conclusions about the behavior on real epipolar planes, they do not cover effects like continuously changing disparity related to non planar objects, occlusions or *non-Lambertian* effects. However they give an insight into the raw orientation estimation ability of the structure tensor and its behavior under certain conditions.

Comparison of gradient filters. The first experiment motivates our choice to compute the gradients of the structure tensor using the *Gaussian* derivatives filter in equation 29. We compute the structure tensor in three different variants, using the *Sobel*-operator (eq. 27), the *Scharr*-operator (eq. 28) and the *Gaussian* derivative operator (eq. 29) and compare their behavior on synthetic EPIs. We use epipolar planes with $h = 15px$, $\sigma_n = 0$ and $\delta_c = 128$. On the one hand, we are interested in the orientation estimation accuracy of all variants, but also in the robustness against untextured regions in the EPIs. As a reminder, we obtain an EPI when fixing a row/column index in the image domain and stack them over a collection of images of different viewpoints. This means the texture of the objects along this rows/columns is mapped into the epipolar space as lines whose slope corresponds to the distance of the object to the camera (compare section 2.3). As a result, the thickness of an epipolar line depends on the intensity variance of texture mapped onto the epipolar space. We simulate this in our experiment by generating random EPIs with epipolar lines of random widths with a maximum width of w_{max}. To evaluate the accuracy in orientation estimation we compute the structure tensor on an EPI with a fixed w_{max} over an orientation range from $d = [-1, 1]$ by applying affine transformations to the EPI to create the different slopes with sub-pixel accuracy. The accuracy is then computed as mentioned in step 3 of the general procedure above as the mean over all orientations. We evaluate this mean error over the whole orientation range for values of $w_{max} = [2, 19]$ and plot the result in figure 23 (a) with the mean orientation estimation error over the maximum epipolar line width. The result is that the *Scharr*-operator leads to more accurate orientation estimations but due to the extensible kernel size, the overall performance of the *Gaussian* derivative filter is more robust against increasing epipolar line widths or, in other words, against the presence of decreasing frequencies in the EPIs.

Optimal structure tensor scales. In the next experiment we use epipolar planes with $h = 15px$, $w_{max} = 4$, $\sigma_n = 0$, $d = [-1, 1]$ and $\delta_c = 128$. We vary the disparity d from $-1px$ to $1px$ in $0.1px$ steps and compute for each disparity the orientation on a parameter grid of the inner scale ρ and outer scale σ of the structure tensor. We varied ρ from 0.4 to 0.9 in steps of 0.01 and σ from 0.6 to 2.5 in 0.01 steps as well. The result is depicted in figure 23. Outer and inner scale behave nearly constant. The stronger variations in the outer scale signal are primarily due to a lesser sensitivity of the outer scale in a wider range – thus some randomness in the exact position of the absolute minimum occurs (compare also figure 25). The slight indentation at disparity zero for both signals can be explained by the fact that orientation estimation works

perfect for vertical lines. Due to this experiment we define in later experiments an inner scale $\rho = 0.75$ and an outer scale $\sigma = 1.0$ as optimal scales for this epipolar plane configuration.

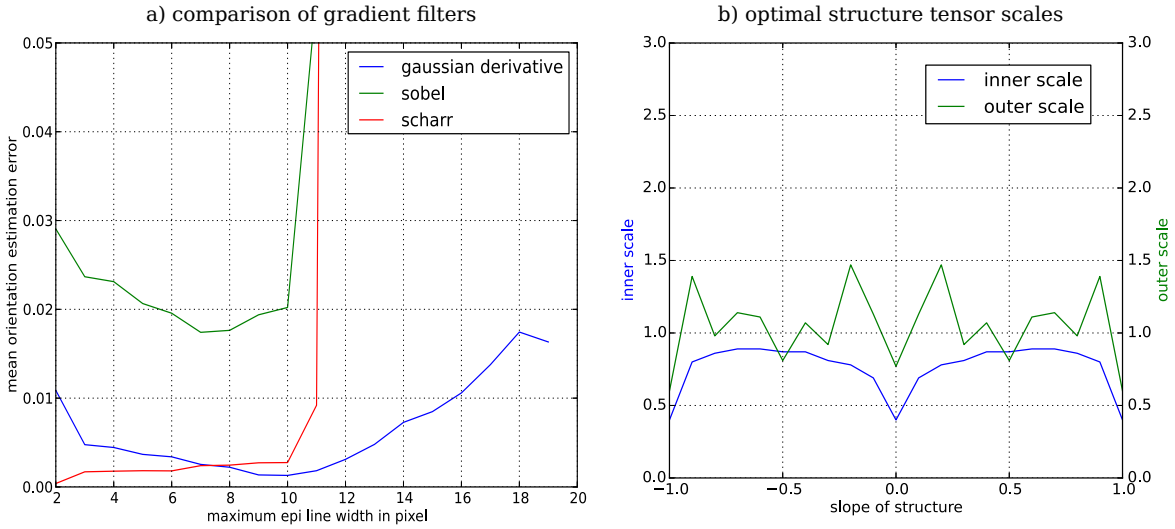


Figure 23: a) Comparison of gradient filters. On synthetic epipolar pane images we evaluate the structure tensor using three different approaches to compute the gradients (*Sobel,Scharr,Gaussian* derivative). We create synthetic EPIs with random epipolar lines of maximum widths w_{max}, where width means the number of pixels of a line in x direction having assigned the same intensity value. We vary w_{max}, drawn on the x-axis, and compute for each w_{max} the orientations over a slope range $d = [-1, 1]$. The average of the estimation errors for each gradient filter is drawn on the y-axis. We see that the *Scharr-Operator* has the best orientation estimation abilities, but with increasing untextured regions the Gaussian derivative filter shows a better overall performance due to the fact that its kernel size is not restricted to 3×3. b) Here, we evaluate the optimal scale parameter of the structure tensor. In this experiment we generate synthetic EPIs and compute for each slope in the range of $d = [-1, 1]$ the inner and outer scale using a grid search and plot the resulting scale parameter with the lowest estimation error for each slope. They behave more or less constant over the range of slopes, the slight indentation at disparities 0 and ± 1 can be explained by the fact that orientation estimation works ideal for vertical and horizontal lines even with small kernel sizes.

Inner and outer scale limits. Using the optimal scale parameter of the structure tensor from our second experiment, we will now look what happens when we fix one scale and vary the other to see the operative range of the corresponding second parameter. Again we use epipolar planes with $h = 15px$, $w_{max} = 4$, $\sigma_n = 0$, $d = [-1, 1]$ and $\delta_c = 128$. Results are depicted in figures 24 and 25. The first show results for a constant σ varying ρ, the second is the opposite. We observe that the inner scale ρ has much narrower tolerances than σ.

Minimal number of cameras. In the next experiment we change the parameter h of an epipolar plane which is equivalent to the number of cameras or sampling steps used

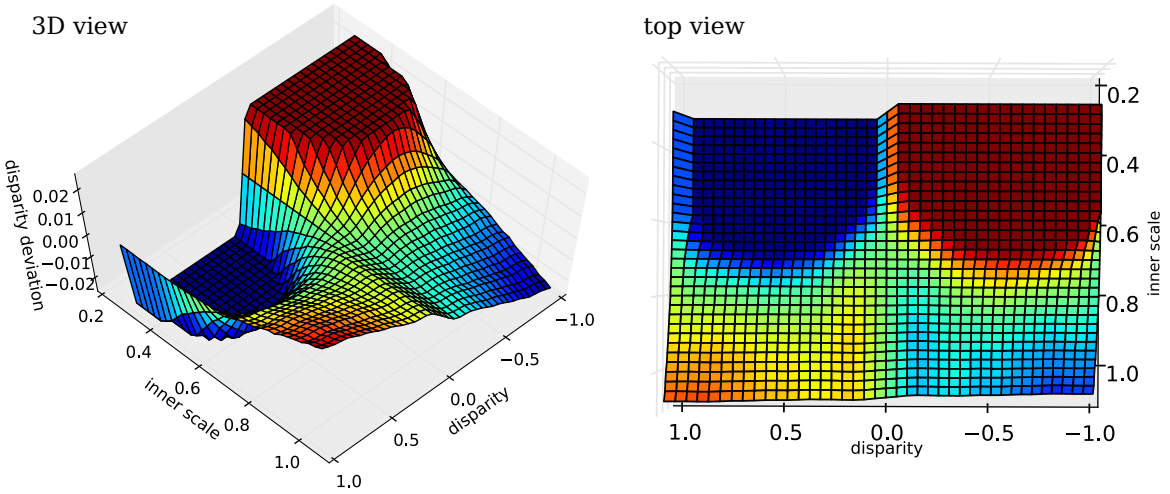


Figure 24: Inner *Gaussian* kernel variation when fixing the outer scale σ to 1.0. Left side shows a 3D view of the disparity deviation $d - d_m$ on the z axis computed over the disparity d and the inner scale ρ. The right side is the left plot viewed from above. It is obvious that the region of minimal disparity deviation $\rho_{opt} \approx [0.7, 0.8]$ is quite narrow.

to acquire the light field. The other parameters are $w_{max} = 4$, $\sigma_n = 0$, $d = [-1.5, 1.5]$ and $\delta_c = 128$. Result are depicted in figure 26 and show that a number of 7 cameras seems optimal for the method. This can be explained with error diffusion caused by border effects. To calculate the structure tensor we need to apply three convolutions. If we use the minimal kernel size of 3×3 each convolution diffuses an error – caused by the image borders – one pixel towards the center row. This adds up to $2 \cdot 3 + 1 = 7$ pixel if we want a center row pixel which is unaffected by border errors. In this experiment we adapted the kernel sizes for structure tensor evaluation to use the EPI height h in an optimal fashion to see if bigger kernel size leads to more and more increasing estimation results. But we see in figure 26 that the estimation accuracy above $h = 11px$ does not increase anymore. Therefore, we propose that at least 7 cameras are necessary and more than 11 superfluous. These statement is only fully valid if one is only interested in an optimal estimation for the center view of a light field, which is equivalent in this experiments to only taking the center row into account. If an optimal depth estimation for more than the center view is desired, an increasing number of cameras can be useful.

Influence of noise on accuracy and coherence. A 1D plot for $h = 7px$, inner and outer kernels of $\sigma = 0.75$ and $\rho = 1.0$ is depicted in figure 27. The left side shows the evaluation on a noise-free EPI and the right side an EPI with a noise level $\sigma_n = 11$. The plots show the coherence and the disparity deviation with its standard deviation. We observe that the orientation analysis seems to work perfectly for disparities ± 1 and 0 and is worse for disparities ± 0.5. However, the overall accuracy is in the range 0.01 px. The error increases quickly if the slope of the EPI lines goes beyond 45°. This is quite

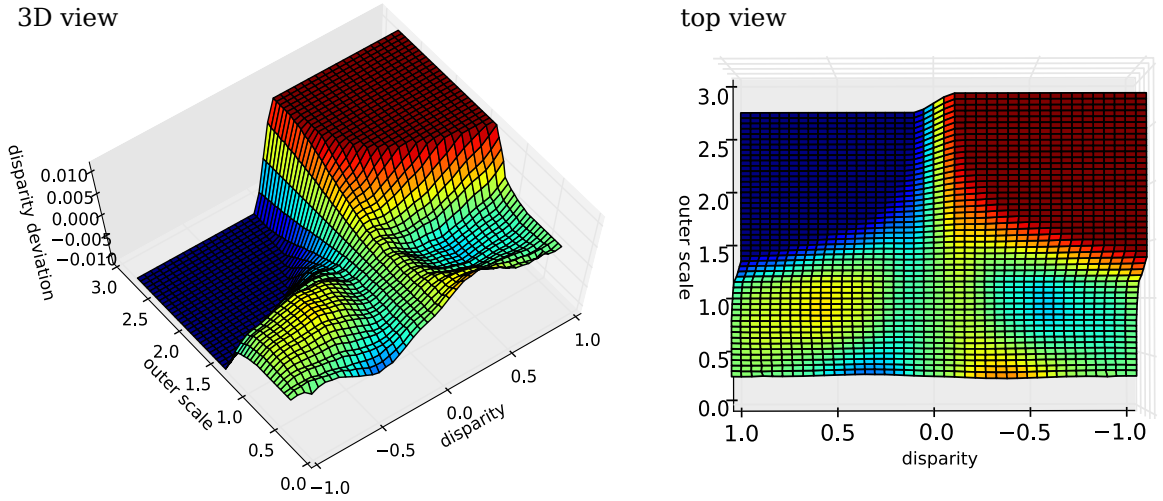


Figure 25: Outer *Gaussian* kernel variation when fixing the inner scale ρ to 0.75. Left side shows a 3D view of the disparity deviation $d - d_m$ on the z axis computed over the disparity d and the outer scale σ. The right side is the left plot viewed from above. It is obvious that the region corresponding to a minimal disparity deviation $\sigma_{opt} \approx [0.5, 1.3]$ is much wider than for the inner scale.

clear when realizing that the incline of a line on epipolar planes is caused by horizontal shifts of the image rows instead of a rotation. This of course leads to a disruption of the line above 45°. Adding noise (figure 27 right) leads to an increasing uncertainty but not to be affecting the overall accuracy that much.

Noise and contrast variation. In two more experiments, depicted in figure 28, we further check the sensitivity to noise and to contrast changes. The results are that the structure tensor is quite robust against decreasing contrast. Also, noise up to a certain amount does not affect the overall accuracy that much but increases the uncertainty of the estimation leading to noisy results.

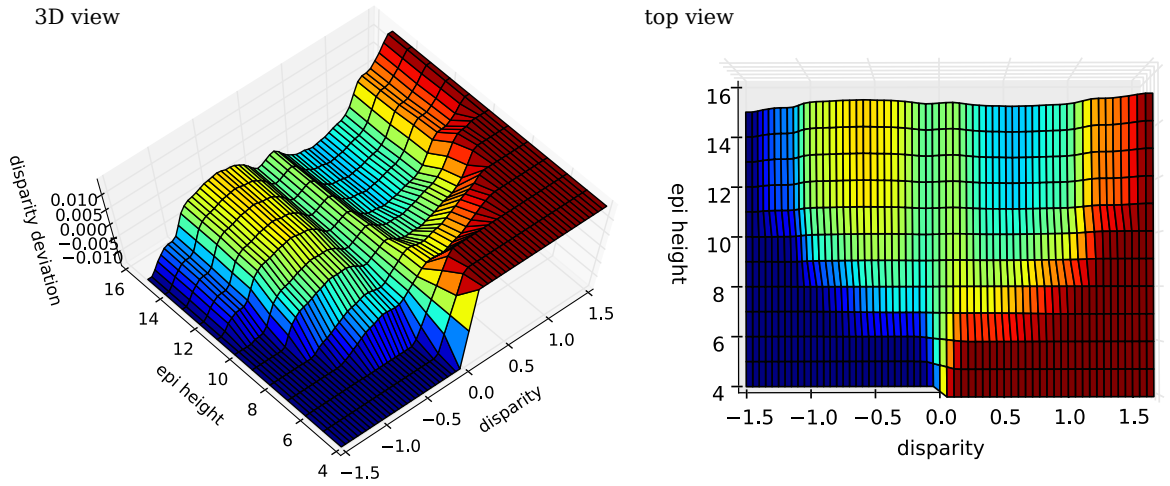


Figure 26: Variation of the EPI height or the number of cameras. We varied the disparity d of an epipolar plane image from -1.5 to 1.5 as well as h from 4px up to 14px. The z-axis is the deviation of the measured disparity from the ground truth disparity $d - d_m$. The measured disparity d_m is calculated using the method described in section 5.1.2.1 whereby the kernel size and standard deviation of the outer *Gaussian* kernel was adapted to the actual height h for each EPI to make use of the increasing EPI line length. As a result we see that $7px$ seems to be the first EPI height covering the entire range of $\pm 1px$ with acceptable accuracy. We also see that above $h = 11px$ the accuracy does not further increase.

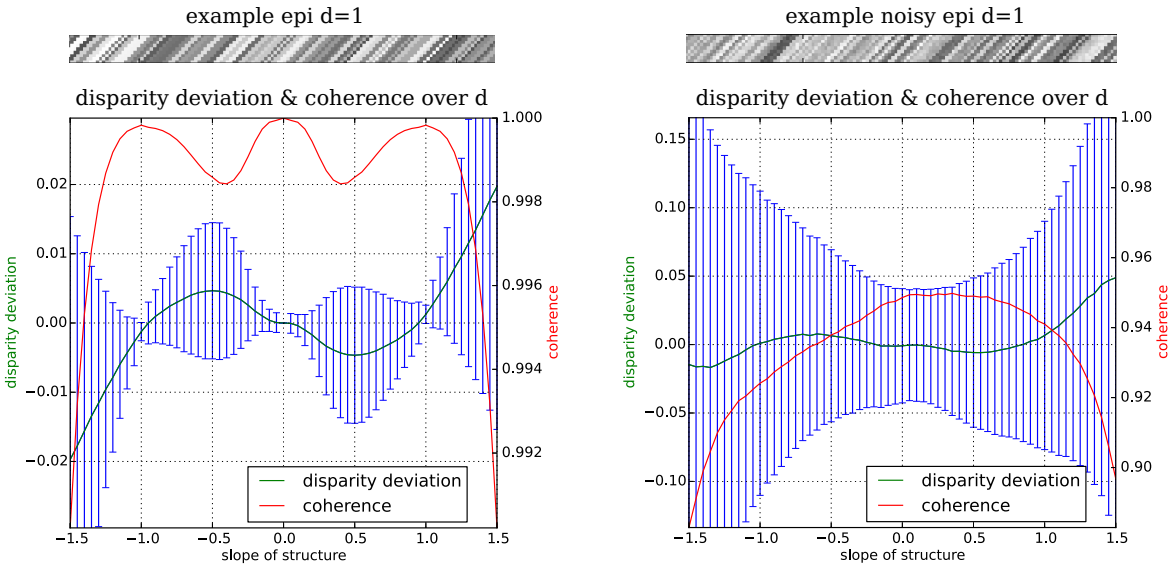


Figure 27: Comparison of orientation analysis on noise free and noisy epipolar plane images. Left side shows an EPI of height $h = 9px$. The smoothing parameters are inner scale $\rho = 0.75$ and outer scale of $\sigma = 1.0$. Same parameters on the right side but with an additive Gaussian noise of $\sigma_n = 31$. Plotted are the disparity deviation $d - d_m$ with standard deviation and the coherence or reliability r. It is obvious that noise does not affect the mean accuracy that much but the certainty is affected through a much lower coherence.

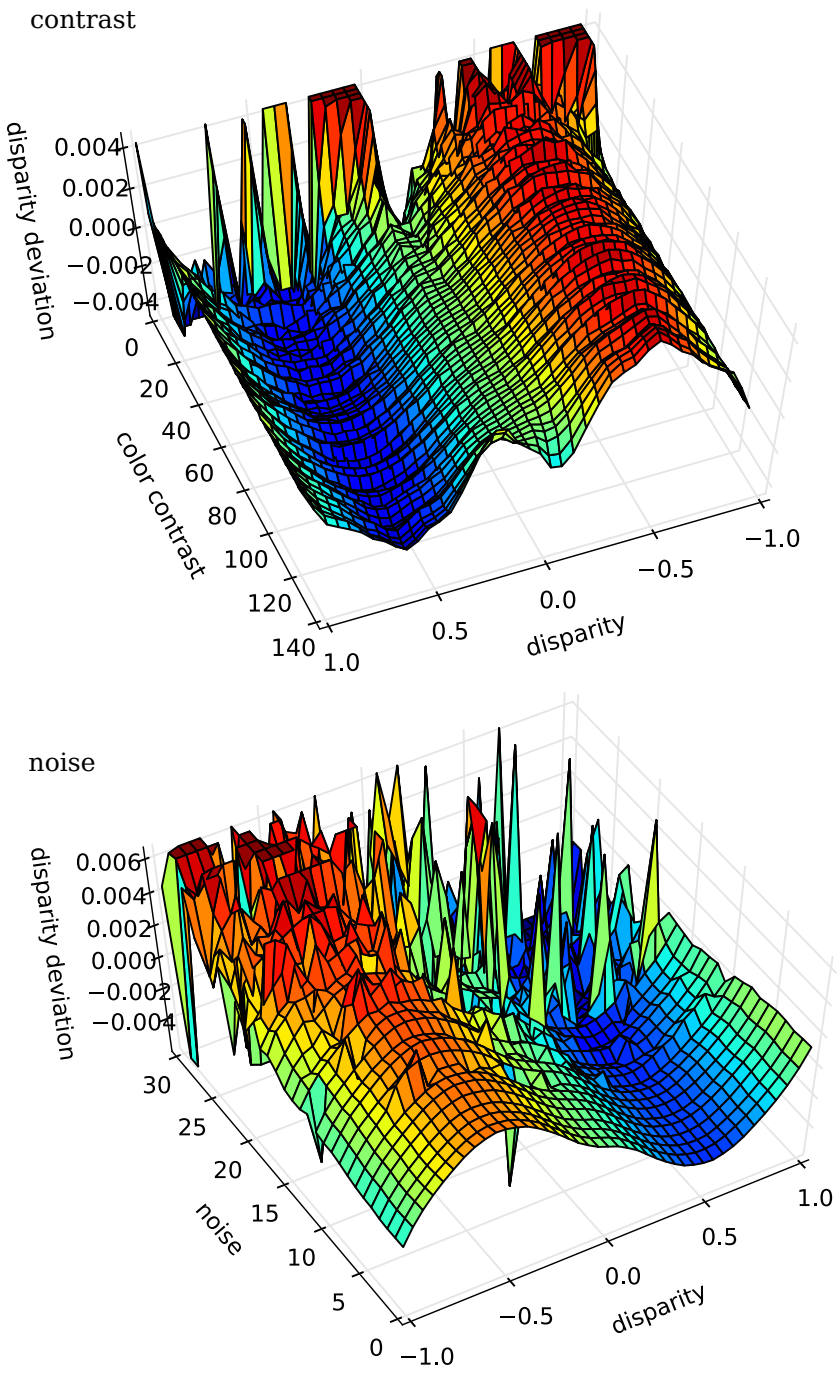


Figure 28: Top: Variation of the image contrast. We generate random EPI lines with random integer intensities $(128 - \delta_c, 128 + \delta_c)$ where $\delta_c \in [1, 128]$. We varied the color contrast δ_c and the disparity to compute the disparity deviation $d - d_m$. The results of the orientation estimation are contrast independent up to very little contrasts. Only at the lowest contrast of $\pm 2px$ do we see significant outliers. Bottom: Noise variation. We see an evaluation of the disparity deviation under increasing additive Gaussian noise $\sigma_n \in [0, 31]$.

5.1.2.3 Consistent Disparity Labeling The computation of the local disparity estimates using the structure tensor only takes into account the immediate local structure of the light field. In truth, the disparity values within a slice need to satisfy global visibility constraints across all cameras for the labeling to be consistent. In particular, a line which is labeled with a certain depth cannot be interrupted by a transition to a label corresponding to a greater depth, since this would violate occlusion ordering, figure 29. In the conference paper [100], a joint work of Dr Goldlücke and the author, we have

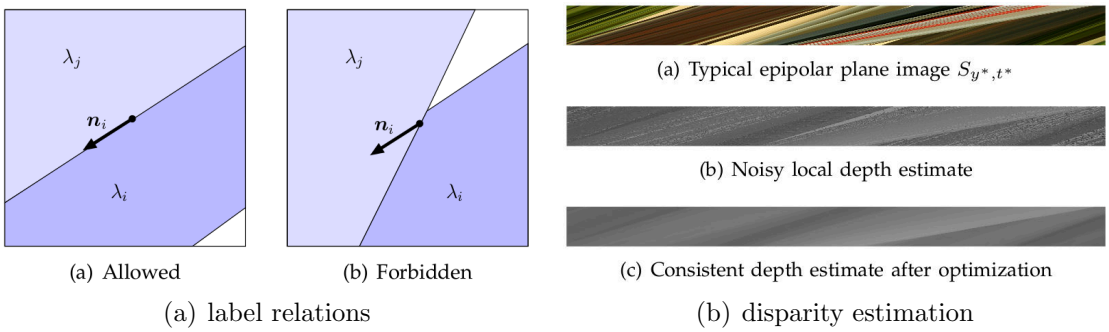


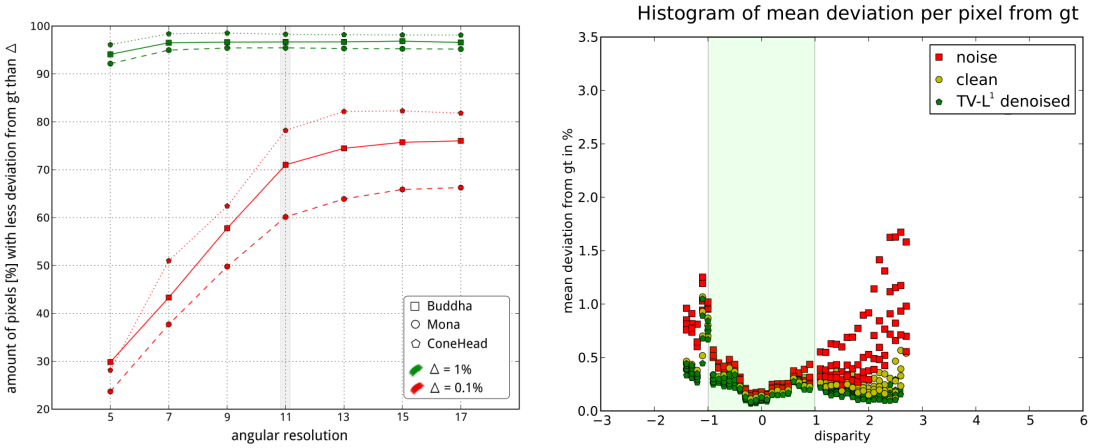
Figure 29: (a) Global labeling constraints on an EPI: if depth λ_i is less than λ_j and corresponds to direction $\mathbf{n_i}$, then the transition from λ_i to λ_j is only allowed in a direction orthogonal to $\mathbf{n_i}$ to not violate occluding order. (b) With the consistent labeling scheme one can enforce global visibility constraints in order to improve the depth estimates for each epipolar plane image.

shown that by using a variational labeling framework based on ordering constraints [93], one can obtain globally consistent estimates for each slice which take into account all views simultaneously. While this is a computationally very expensive procedure, it yields convincing results, see figure 29. In particular, consistent labeling greatly improves robustness to non-Lambertian surfaces, since they typically lead only to a small subset of outliers along an EPI-line. However, at the moment this is only a proof of concept, since it is far too slow to be usable in any practical applications. For this reason, we do not pursue this method further in this work, and instead evaluate only the interactive technique, using results from the local structure tensor computation directly.

5.1.3 Disparities On Individual Views

After obtaining EPI disparity estimates d_{y^*,t^*} and d_{x^*,s^*} from the horizontal and vertical slices, respectively, we integrate those estimates into a consistent single disparity map $u : \Omega \to \mathbb{R}$ for each view (s^*, t^*). This is the objective of the following section.

5.1.3.1 Fast Denoising Scheme Obviously, the fastest way to obtain a sensible disparity map for the view is to just point-wise choose the disparity estimate with the higher reliability r_{x^*,s^*} or r_{y^*,t^*}, respectively. We can see that it is still quite noisy, furthermore, edges are not yet localized very well, since computing the structure tensor



(a) Accuracy depending on angular resolution (b) Mean error depending on disparity for dataset buddha

Figure 30: Analysis of the error behaviour from two different points of view. In a), we plot the percentage of pixels which deviate from the ground truth (gt) by less than a given threshold over the angular resolution. Very high accuracy (i.e. more than 50% of pixels deviate by less than 0.1%) requires an angular resolution of the light field of at least 9×9 views. In b), we show the relative deviation from ground truth over the disparity value in pixels per angular step. Results were plotted for local depth estimations calculated from the original (clean) light field, local depth estimated from the same light field with additional Poisson noise (noisy) as well as the same result after TV-L^1 denoising, respectively. While the ideal operational range of the algorithm are disparities within ± 1 pixel per angular step, denoising significantly increases overall accuracy outside of this range.

entails an initial smoothing of the input data. For this reason, a fast method to obtain quality disparity maps is to employ a TV-L^1 smoothing scheme, where we encourage discontinuities of u to lie on edges of the original input image by weighting the local smoothness with a measure of the edge strength. We use

$$g(x, y) = 1 - r_{s^*,t^*}(x, y), \tag{39}$$

where r_{s^*,t^*} is the coherence measure for the structure tensor of the view image, defined similarly as in (38). Higher coherence means a stronger image edge, which thus increases the probability of a depth discontinuity.

We then minimize the weighted TV-L^1 smoothing energy

$$E(u) = \int_\Omega g\,|Du| + \frac{1}{2\lambda}\,|u - f|\;\mathrm{d}(x, y), \tag{40}$$

where f is the noisy disparity estimate and $\lambda > 0$ a suitable smoothing parameter. The minimization is implemented in the open-source library `cocolib` [37] by Dr. Goldlücke and performs in real-time.

5.1.3.2 Global Optimization Scheme From a modeling perspective, a more sophisticated way to integrate the vertical and horizontal slice estimates is to employ a globally optimal labeling scheme in the domain Ω, where we minimize a functional of the form

$$E(u) = \int_{\Omega} g\,|Du| + \rho(u, x, y)\,\mathrm{d}(x, y). \tag{41}$$

In the data term, we want to encourage the solution to be close to either d_{x^*,s^*} or d_{y^*,t^*}, while suppressing impulse noise. Also, the two estimates d_{x^*,s^*} and d_{y^*,t^*} shall be weighted according to their reliability r_{x^*,s^*} and r_{y^*,t^*}. We achieve this by setting

$$\begin{aligned}\rho(u, x, y) := \min(&r_{y^*,t^*}(x, s^*)\,|u - d_{y^*,t^*}(x, s^*)|\,,\\ &r_{x^*,s^*}(y, t^*)\,|u - d_{x^*,s^*}(y, t^*)|).\end{aligned} \tag{42}$$

We compute globally optimal solutions to the functional (41) using the technique of functional lifting described in [74], which is also implemented in `cocolib` [37]. While being more sophisticated modeling-wise, the global approach requires minutes per view instead of being real-time, and a discretization of the disparity range into labels, which might even lead to a loss instead of gain in accuracy.

5.1.4 Performance Analysis for Interactive Labeling

In this section, we perform detailed experiments with the local disparity estimation algorithm to analyze both quality and speed of this method. The aim is to investigate how well our disparity estimation paradigm performs when the focus lies on interactive applications, as well as find out more about the requirements regarding light field sampling and the necessary parameters.

Optimal Parameter Selection. In a first experiment, we establish guidelines to select optimal inner and outer scale parameters of the structure tensor. As a quality measurement, we use the percentage of depth values below a relative error

$$\epsilon = |u(x, y) - r(x, y)|/r(x, y) \tag{43}$$

where u is the depth map for the view and r the corresponding ground truth. Optimal parameters are then found with a simple grid search strategy, where we test a number of different parameter combinations. Results are depicted in figure 22, and determine the optimal parameter for each light field resolution and data set. Following evaluations are all done with these optimal parameters. In general, it can be noted that an inner scale parameter of 0.08 is always reasonable, while the outer scale should be chosen larger with larger spatial and angular resolution to increase the overall sampling area. Here, it could be noted that applying median filtering to the results will reduce the outer-scale parameter behavior in figure 22 which causes a better edge preserving in the results but this is of course linked with higher computational cost. Here we did the experiments

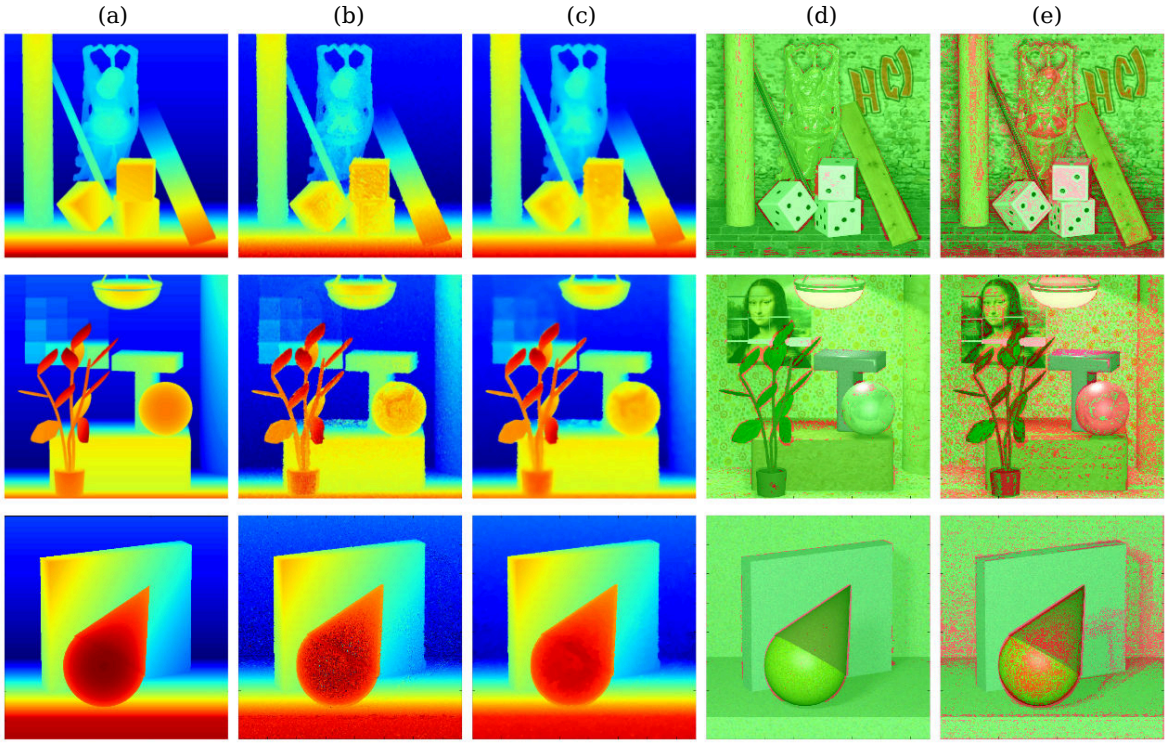


Figure 31: Results of disparity estimation on the datasets Buddha (top), Mona (center) and Conehead (bottom). (a) shows ground truth data, (b) the local structure tensor disparity estimate described in section 5.1.2.1 and (c) the result after TV-L^1 denoising according to section 5.1.3. In (d) and (e), one can observe the amount and distribution of error, where green labels mark pixels deviating by less than the given threshold from ground truth, red labels pixels which deviate by more. Most of the larger errors are concentrated around image edges.

without any post-processing to show the raw ability of the local method.

Minimum Sampling Density. In a second step, we investigate what sampling density we need for an optimal performance of the algorithm on light fields instead of synthetically created EPIs (compare section 5.1.2.2 and figure 26). To achieve this, we evaluated three simulated light fields over the full angular resolution range with the optimal parameter selection found in figure 22. The results are illustrated in figure 30, and show that for very high accuracy, i.e. less than 0.1% deviation from ground truth, we require about nine views in each angular direction of the light field.

Moreover, the performance degrades drastically when the disparities become larger than around ± 1 pixels, which makes sense from a sampling perspective since the derivatives in the structure tensor are computed on a 3×3 stencil. Together with the characteristics of the camera system used (baseline, focal length, resolution), this places constraints on the depth range where we can obtain estimates with our method. For the Raytrix

Plenoptic Camera we use in the later experiments, for example, it turns out that we can reconstruct scenes which are roughly contained within a cube-shaped volume, whose size and distance is determined by the main lens we choose.

Noisy Input. Another interesting fact is observable on the right hand side of figure 30, where we test the robustness against noise (compare also figure 27). Within a disparity range of ± 1, the algorithm is very robust, while the results quickly degrade for larger disparity values when impulse noise is added to the input images. However, when we apply TV-L^1 denoising, which requires insignificant extra computational cost, we can see that the deviation from ground truth is on average reduced below the error resulting from a noise-free input. Unfortunately, denoising always comes at a price: since it naturally incurs some averaging, while accuracy is globally increased, some sub-pixel details can be lost. In figure 31 we observe the distribution of the errors, and can see that almost all large-scale error is concentrated around depth discontinuities.

Disparity Range Limitation. The histogram plot on the right in figure 30 depicts the effect of rapidly increasing orientation estimation errors if the disparities exceed a range of ± 1. The reason is that the slope of an epipolar line depends on shifts in the image domain relative to the center view (see figure 32). The pixels of an epipolar line with a slope $> |\pm 1|$ are torn apart and thus cannot be matched as a line anymore using convolution operations. Reconstructing scenes with disparity ranges above 2 pixels

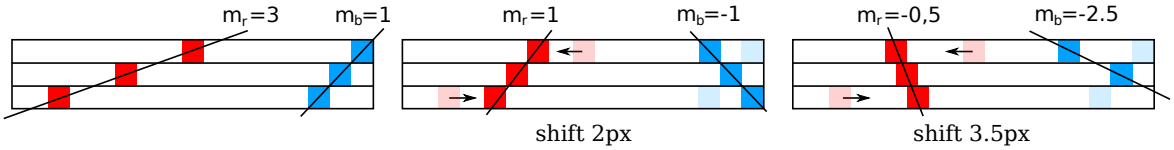


Figure 32: Visualization of a refocusing operation on the EPI domain. The left image sketches an EPI with a red and a blue epipolar line initially having slopes of $m_r = 3$ and $m_b = 1$ respectively. By shifting the rows of the EPI opposing with respect to the center view the slope of each line changes like depicted in the middle and right image.

makes an iterative processing necessary. One simply repeats the following steps over the entire disparity range:

- refocus the light field.
- compute orientations.
- store valid disparities between ± 1.
- add total pixel shift to the disparities.

Merging the corresponding results from each iteration step can be done for example by choosing the disparity with the highest coherence (see eq. 38).

5.1.5 Comparison to Multi-View Stereo

We compute a simple local stereo matching cost for a single view as follows. Let $V = \{(s_1, t_1), ..., (s_N, t_N)\}$ be the set of N view points with corresponding images $I_1, ..., I_N$, with (s_c, t_c) being the location of the current view I_c for which the cost function is being computed. We then choose a set Λ of 64 disparity labels within an appropriate range. For our test we choose equidistant labels within the ground truth range for optimal results. The local cost $\rho_{\mathrm{AV}}(x, l)$ for label $l \in \Lambda$ at location $x \in I_c$ computed on *all* neighboring views is then given by

$$\rho_{\mathrm{AV}}(x, l) := \sum_{(s_n, t_n) \in V} \min(\epsilon, \|I_n(x + l v_n) - I_c(x)\|), \tag{44}$$

where $v_n := (s_n - s_c, t_n - t_c)$ is the view point displacement and $\epsilon > 0$ is a cap on the error to suppress outliers. To test the influence of the number of views, we also compute a cost function on a *crosshair* of view points along the s- and t-axis from the view (s_c, t_c), which is given by

$$\rho_{\mathrm{CH}}(x, l) := \sum_{\substack{(s_n, t_n) \in V \\ s_n = s_c \text{ or } t_n = t_c}} \|I_n(x + l v_n) - I_c(x)\| . \tag{45}$$

In effect, this cost function thus uses exactly the same number of views as required for the local structure tensor of the center view. The results of these two purely local methods can be found under **ST_AV_L** for all views, and **ST_CH_L** for all views or just a *crosshair*, respectively.
Results of both multi-view data terms are denoised with a simple TV-L^2 scheme, algorithms **ST_AV_S** and **ST_CH_S**. Finally, they were also integrated into a global energy functional

$$E(u) = \int_\Omega \rho(x, u(x)) \, \mathrm{d}x + \lambda \int_\Omega |Du| \tag{46}$$

for a labeling function $u : \Omega \to \Lambda$ on the image domain Ω, which is solved to global optimality using the method in [74]. The global optimization results can be found under algorithms **ST_AV_G** and **ST_CH_G**. We compare to our approach. First, we start with the purely local method **EPI_L**, which estimates orientation using the Eigensystem analysis of the structure tensor discussed in section 5.1.2. The second method, **EPI_S**, just performs a TV-L^2 denoising of this result, while **EPI_G** employs the globally optimal labeling scheme of section 5.1.3.2. Finally, the method **EPI_C** performs a constrained denoising on each epipolar plane image, which takes into account occlusion ordering constraints [39]. All results are depicted in figure 33.

5.1.6 Experiments and Discussion

The table in figure 33 and the figures 34, 35 show detailed visual and quantitative disparity estimation results on our benchmark datasets. Algorithm parameters for all methods were tuned for an optimal structural similarity (SSIM) measure. Strong

arguments why this measure should be preferred to the MSE are given in [99], but we also have computed a variety of other quantities for comparison (however, the detailed results vary when parameters are optimized for different quantities).

First, one can observe that our local estimate always is more accurate than any of the multi-view stereo data terms, while using all of the views gives slightly better results for multi-view than using only the crosshair. Second, our results after applying the TV-L^1 denoising scheme (which takes altogether less than two seconds for all views) are more accurate than all other results, even those obtained with global optimization schemes (which takes minutes per view). A likely reason why our results do not become better with global optimization is that the latter requires a quantization in to a discrete set of disparity labels, which of course leads to an accuracy loss. Notably, after either smoothing or global optimization, both multi-view stereo data terms achieve the same accuracy, see figure 33 - it does not matter that the crosshair data term makes use of less views, likely since information is propagated across the view in the second step. This also justifies our use of only two epipolar plane images for the local estimate.

Our method also is the fastest, achieving near-interactive performance for computing disparity maps for all of the views simultaneously. Note that by construction, the disparity maps for all views are always computed simultaneously. Performance could further be increased by restricting the computation on each EPI to a small stripe if only the result of a specific view is required.

Obviously – when analyzing epipolar plane images – our approach does not use the full 4D light field information around a ray to obtain the local estimates - we just work on two different 2D cuts through this space. The main reason is performance, in order to be able to achieve close to interactive speeds, which is necessary for most practical applications, the amount of data which is used locally must be kept to a minimum. Moreover, in experiments with a multi-view stereo method, it turns out that using all of the views for the local estimate, as opposed to only the views in the two epipolar plane images, does not lead to overall more accurate estimates. While it is true that the local data term becomes slightly better, the result after optimization is the same. A likely reason is that the optimization or smoothing step propagates the information across the view.

Orientation Analysis

lightfield	EPI_L	EPI_S	EPI_C	EPI_G
buddha	0.81	0.57	0.55	0.62
buddha2	1.22	0.87	0.87	0.89
horses	3.60	2.12	2.21	2.67
medieval	1.69	1.15	1.10	1.24
monasRoom	1.15	0.90	0.82	0.93
papillon	3.95	2.26	2.52	2.48
stillLife	3.94	3.06	2.61	3.37
couple	0.40	0.18	0.16	0.19
cube	1.27	0.85	0.82	0.87
maria	0.19	0.10	0.10	0.11
pyramide	0.56	0.38	0.38	0.39
statue	0.88	0.33	0.29	0.35
average	1.64	1.07	1.04	1.18

Multi-View Stereo

lightfield	ST_AV_L	ST_AV_S	ST_AV_G	ST_CH_L	ST_CH_S	ST_CH_G
buddha	1.20	0.78	0.90	1.01	0.67	0.80
buddha2	2.26	1.05	0.68	3.08	1.31	0.75
horses	5.29	1.85	1.00	6.14	2.12	1.06
medieval	7.22	0.91	0.76	12.14	1.08	0.79
monasRoom	2.25	1.05	0.79	2.28	1.02	0.81
papillon	4.84	2.92	3.65	4.85	2.57	3.10
stillLife	5.08	4.23	4.04	4.48	3.36	3.22
couple	0.60	0.24	0.30	1.10	0.24	0.30
cube	1.28	0.51	0.56	2.25	0.51	0.55
maria	0.34	0.11	0.11	0.51	0.11	0.11
pyramide	0.72	0.42	0.42	1.30	0.43	0.42
statue	1.56	0.21	0.21	3.39	0.29	0.21
average	2.72	1.19	1.12	3.54	1.14	1.01

Figure 33: Detailed evaluation of all disparity estimation algorithms described in section 5.1.5 on all of the data sets in our benchmark. The values in the tables show the mean squared error in pixels times 100, i.e. a value of "0.81" means that the mean squared error in pixels is "0.0081".

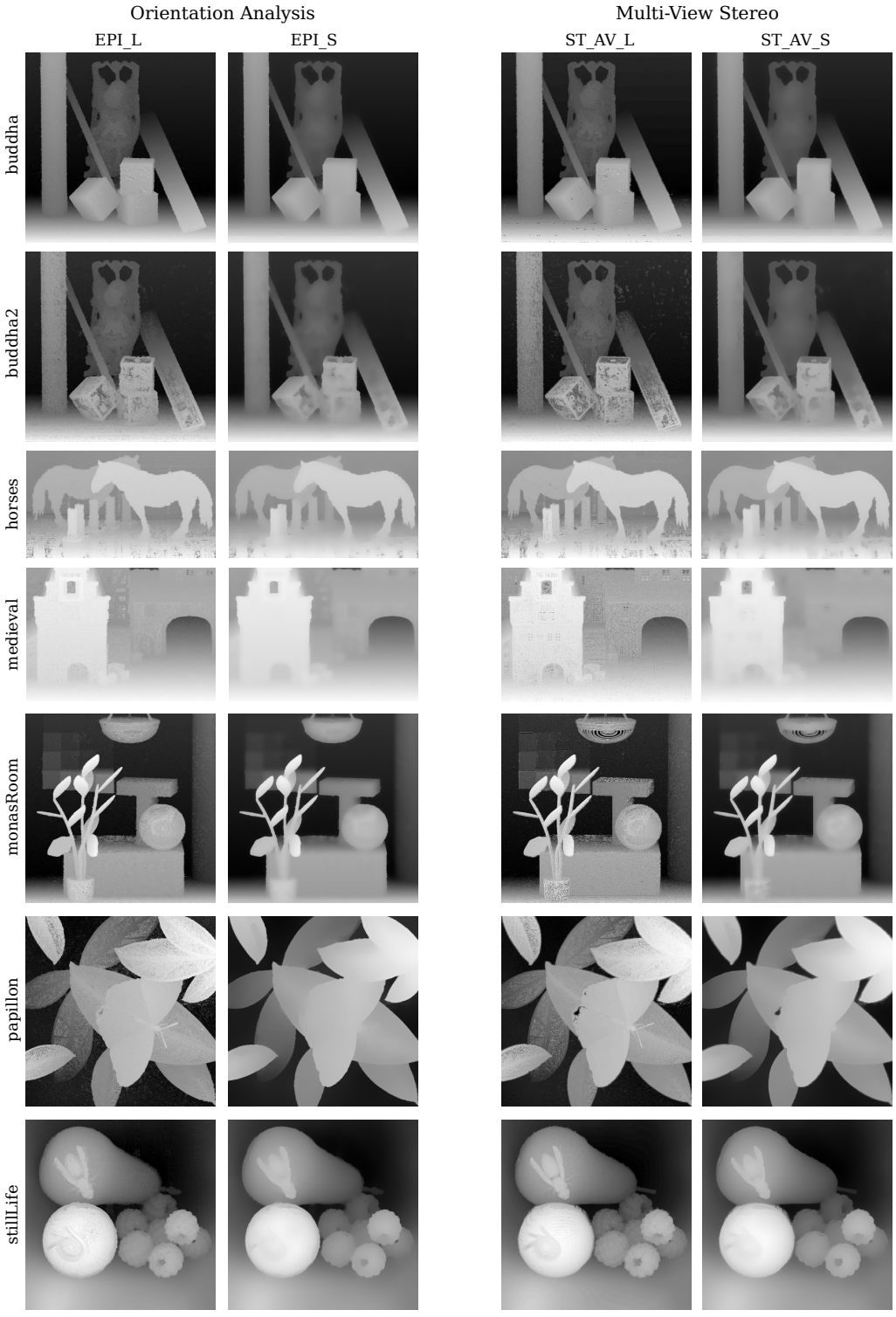


Figure 34: Comparison of the orientation analysis and multi-view stereo (see section 5.1.5) using the synthetic data of the benchmark database (see section 4). First two columns depict the local estimated disparity using the structure tensor (EPI_L) described in section 5.1.2.1 and the results after applying a TV-denoising (EPI_S). Third and fourth columns depict results from the multi-view stereo algorithm (ST_AV_L) described in section 5.1.5 and a TV-denoised version (ST_AV_S) as well.

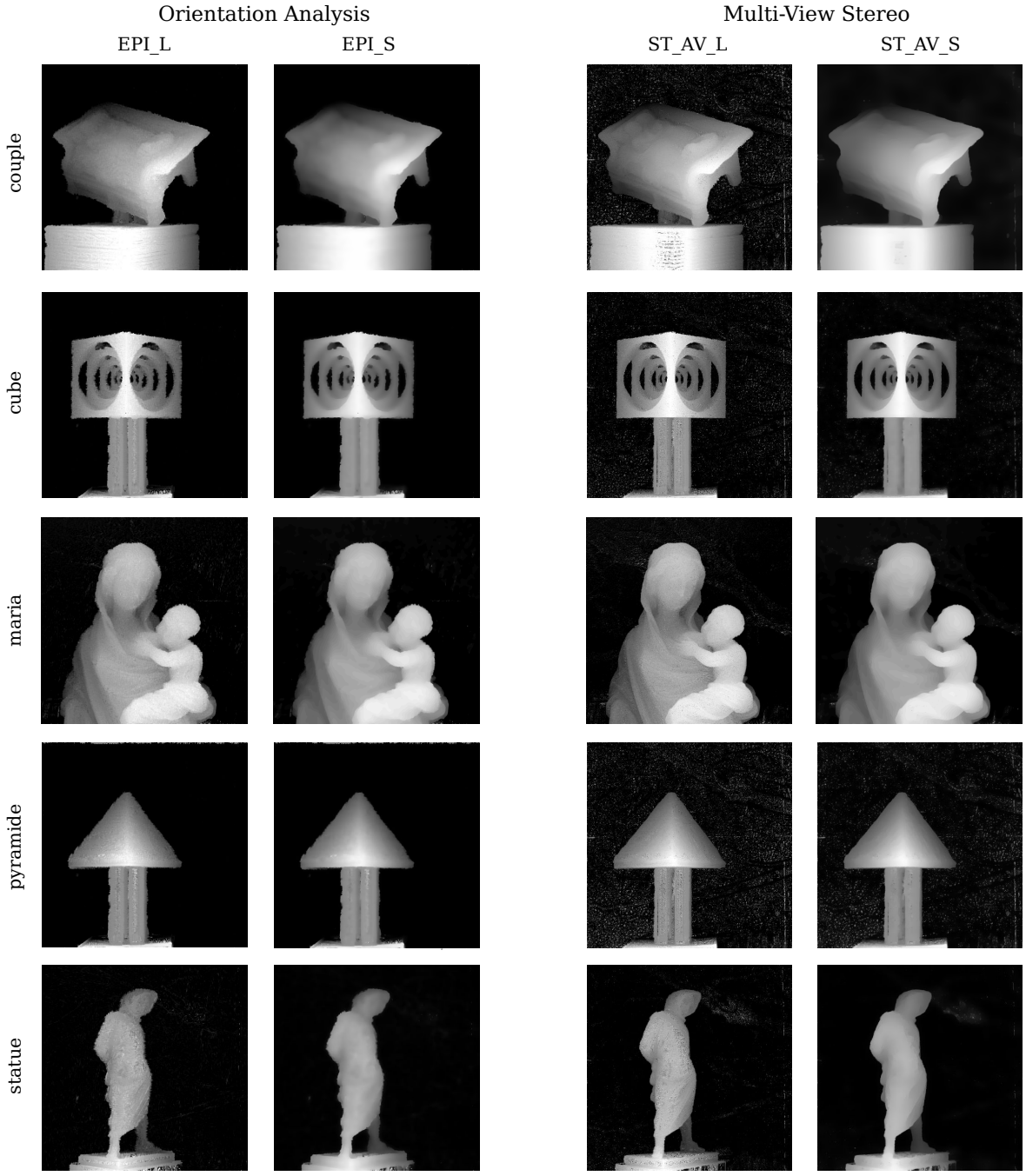


Figure 35: Comparison of the orientation analysis and multi-view stereo (see section 5.1.5) using the real world data of the benchmark database (see section 4). First two columns depict the local estimated disparity using the structure tensor (EPI_L) described in section 5.1.2.1 and the results after applying a TV-denoising (EPI_S). Third and fourth columns depict results from the multi-view stereo algorithm (ST_AV_L) described in section 5.1.5 and a TV-denoised version (ST_AV_S) as well.

5.2 Double Orientation Analysis

While there has been progress in the field of non-Lambertian reconstruction under controlled lighting conditions [4, 29, 40, 79], it remains quite hard to generalize the standard matching models to more general reflectance functions if only a set of images under unknown illumination is available. Previous attempts employ a rank constraint on the radiance tensor [46] to derive a discrepancy measure for non-Lambertian scenes. While this improves upon the standard Lambertian matching models and allows to reconstruct surface reflection parameters, the results still somewhat lack in robustness.

An interesting alternative approach is Helmholtz stereopsis from Zickler et al. [112], which makes use of the symmetry of reflectance or Helmholtz reciprocity principle in order to eliminate the view dependency of specular reflections in restricted imaging setups. By alternating light source and camera at two different locations, one can obtain a stereo pair where specularities are exactly identical and thus classical matching techniques can be employed for non-Lambertian scenes. Other works try to remove reflection data from images using prior assumptions or user input [55, 56].

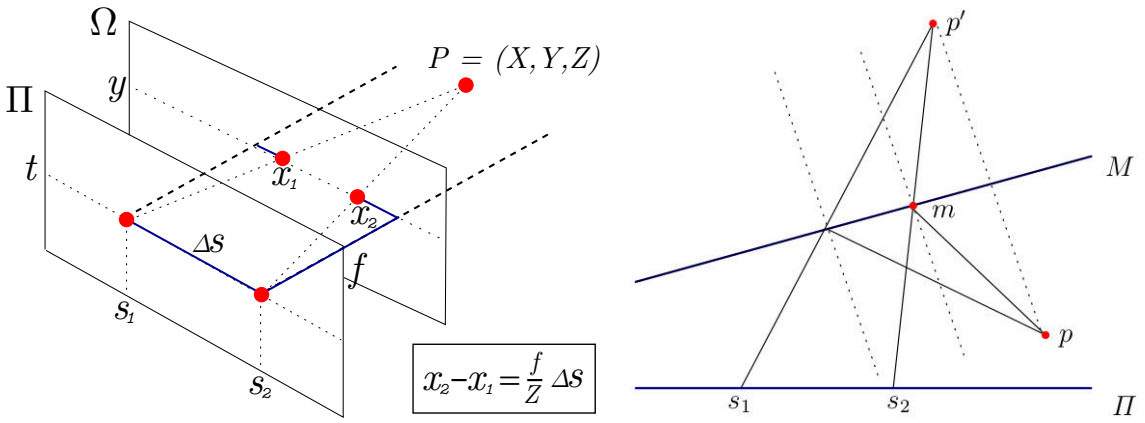
The works which are most closely related to ours are Sinha et al. [96] and Tsin et al. [90]. They also separate a reflecting surface from the reflection in an epipolar volume data structure. At their heart, these works still rely on classical correspondence matching, since they optimize for two overlaid matching models in a nested plane sweep algorithm using graph cuts or semi-global matching, respectively.

In contrast, in our proposed method we do not try to optimize for correspondence. Instead, we build upon early ideas in camera motion analysis [16] and investigate directional patterns in epipolar space. In our case, reflections and transparencies manifest as overlaid structures, which we investigate with higher order structure tensors [1] as a consequent generalization of section 5.1.

As a result, we obtain a direct continuous method which requires no discretization into depth labels, and which is highly parallelizable and quite fast: a center view disparity map for both layers can be obtained in less than two seconds for a reasonably sized light field, which is around a hundred times faster than even the shortest run-times reported in [90]. The content in this section is published in Wanner et al. [102], whereby the theory of the optimization techniques as well as fast CUDA [71] implementations of the algorithms published in *cocolib* [37] are the work of Dr. Bastian Goldlücke.

5.2.1 EPI Structure for Lambertian Surfaces

Before discussing the mapping of reflections into the epipolar space let us quickly recap the model for single orientation, which is equivalent to the assumption of Lambertian material properties.



(a) Light field parametrization

(b) Mirror plane geometry

Figure 36: (a) Each camera location (s, t) in the view point plane Π yields a different pinhole view of the scene. The two thick dashed black lines are orthogonal to both planes, and their intersection with the plane Ω marks the origins of the (x, y)-coordinate systems for the views (s_1, t) and (s_2, t), respectively. (b) Geometry of reflection on a planar mirror. All cameras view the reflections of a scene point p at a planar mirror M as the image of a virtual point p' which lies behind the mirror plane. We assume the intensity measured by the sensor has two contributions, an intensity or color $c(m)$ – the contribution of the reflector m – and a color $c(p)$ – the contribution of the mirrored object p.

Let $P \in \mathbb{R}^3$ be a scene point. It is easy to show that the projection of P on each epipolar plane image is a straight line with slope $\frac{f}{Z}$, where Z is the depth of P, i.e. distance of P to the plane Π, and f the focal length, i.e. distance between the planes Π and Ω (compare sections 2.2, 2.3 and figure 36 a). The quantity $\frac{f}{Z}$ (equation 11) is called the disparity of P. In particular, the above means that if P is a point on an opaque Lambertian surface, then for all points on the epipolar plane image where the point P is visible, the light field L (equation 9) must have the same constant intensity. This is the reason for the single pattern of solid lines which we can observe in the epipolar plane images of a Lambertian scene. In section 5.1, this well-known observation was the foundation for a novel approach to depth estimation, which leveraged the structure tensors of the epipolar plane images in order to estimate the local orientation and thus the disparity of the observed point visible in the corresponding ray. While in conjunction with visibility constraints this leads to a certain robustness against specular reflections, the image formation model implicitly underlying this method is still the Lambertian one, thus the method cannot deal correctly with reflecting surfaces. Furthermore, it is not possible to infer information for both the surface And a possible reflection. The following sections will propose a more general model to remedy this.

5.2.2 EPI Structure for Planar Reflectors

We now introduce an idealized appearance model for the epipolar plane images in the presence of a planar mirror - a translucent surface is an obvious specialization where

a real object takes the place of the virtual one behind the mirror. It is kept simple in order to arrive at a computationally tractable model, but we will see that it captures the characteristics of reflective and translucent surfaces reasonably well to be able to cope with real-world data. A similar appearance model was successfully employed in [90].

Let $M \subset \mathbb{R}^3$ be the surface of a planar mirror. We fix coordinates (y^*, t^*) and consider the corresponding epipolar plane image L_{y^*,t^*}. The idea of the appearance model is to define the observed color for a ray at location (x, s) which intersects the mirror at $m \in M$. Our simplified assumption is that the observed color is a linear combination of two contributions. The first is the base color $c(m)$ of the mirror, which describes the appearance of the mirror without the presence of any reflection. The second is the color $c(p)$ of the reflection, where p is the first scene point where the reflected ray intersects the scene geometry, see Figure 36(a). We do not consider higher order reflections, and assume the surface at p to be Lambertian. We also assume the reflectivity $\alpha > 0$ is a constant independent of viewing direction and location. The epipolar plane image itself will then be a linear combination

$$L_{y^*,t^*} = L^M_{y^*,t^*} + \alpha L^V_{y^*,t^*} \tag{47}$$

of a pattern $L^M_{y^*,t^*}$ from the mirror surface itself as well as a pattern $L^V_{y^*,t^*}$ from the virtual scene behind the mirror. In each point (x, s) as above, both constituent patterns have a dominant direction corresponding to the disparities of m and p. The next section shows how to extract these two dominant directions.

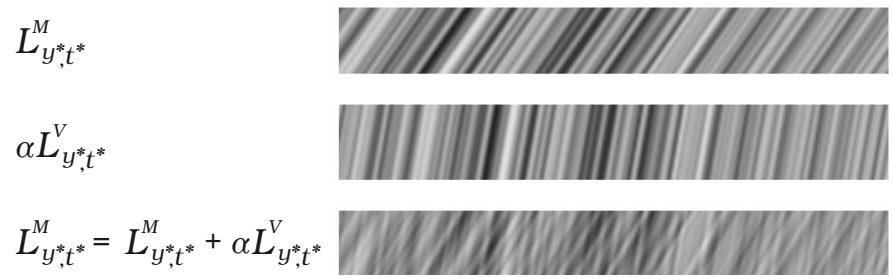


Figure 37: Illustration of overlayed signals $L^M_{y^*,t^*}$, $L^V_{y^*,t^*}$ and L_{y^*,t^*}

5.2.3 Analysis of Multiorientation Patterns

We briefly summarize the theory for the analysis of superimposed patterns described in Aach et al. [1]. A region $R \subset \Omega$ of an image $f : \Omega \to \mathbb{R}$ has orientation $\mathbf{v} \in \mathbb{R}^2$ if and only if

$$f(x) = f(x + \alpha \mathbf{v}) \;\; \forall\, x, x + \alpha \mathbf{v} \in R. \tag{48}$$

Analysis shows that the orientation $\mathbf{v}$ is given by the Eigenvector corresponding to the smaller Eigenvalue of the structure tensor [12] of f. However, the model fails if the

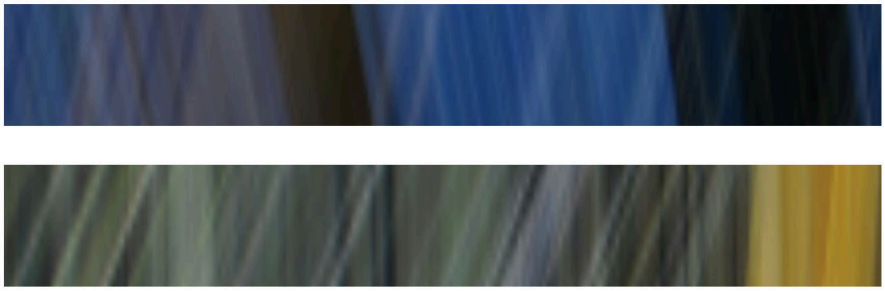


Figure 38: Exemplary epipolar plane images showing double orientation patterns from reflections.

image f is a superposition of two oriented images, $f = f_1 + f_2$, where f_1 has orientation $\mathbf{u}$ and f_2 has orientation $\mathbf{v}$. In this case, the two orientations $\mathbf{u}, \mathbf{v}$ need to satisfy the conditions

$$\mathbf{u}^T \nabla f_1 = 0 \text{ and } \mathbf{v}^T \nabla f_2 = 0 \tag{49}$$

individually on R. Analogous to the single orientation case, the two orientations in a region R can be found by performing an Eigensystem analysis of the second order structure tensor, see Aach et al. [1],

$$\mathcal{T} = \int_R \sigma \begin{bmatrix} f_{xx}^2 & f_{xx}f_{xy} & f_{xx}f_{yy} \\ f_{xx}f_{xy} & f_{xy}^2 & f_{xy}f_{yy} \\ f_{xx}f_{yy} & f_{xy}f_{yy} & f_{yy}^2 \end{bmatrix} \mathrm{d}(x, y), \tag{50}$$

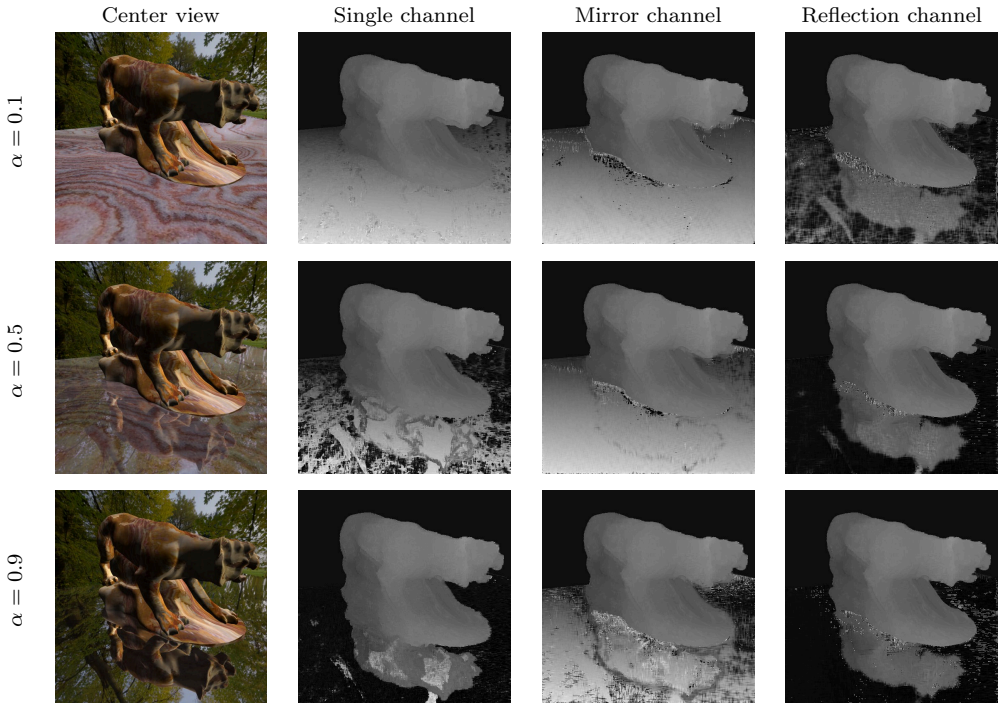
where σ is a (usually Gaussian) weighting kernel on R which essentially determines the size of the sampling window. Since $\mathcal{T}$ is symmetric, we can compute Eigenvalues and Eigenvectors in a straight-forward manner using the explicit formulas in [91]. Analogous to the Eigenvalue decomposition of the 2D structure tensor, the Eigenvector $\mathbf{a} \in \mathbb{R}^3$ corresponding to the smallest Eigenvalue of $\mathcal{T}$, the so-called MOP vector, encodes the orientations. Indeed, the two disparities are equal to the Eigenvalues λ_+, λ_- of the 2×2 matrix

$$\begin{bmatrix} a_2/a_1 & -a_3/a_1 \\ 1 & 0 \end{bmatrix}, \tag{51}$$

from which one can compute the orientations $\mathbf{u} = [\lambda_+ \; 1]^T$ and $\mathbf{v} = [\lambda_- \; 1]^T$.

5.2.4 Merging into Single Disparity Maps

From the steps sketched above, we obtain three different disparity estimates for both the horizontal as well as vertical epipolar images: one from the single orientation model, and two from the double orientation model. It is clear that the closer estimate in the double orientation model will always correspond to the primary surface, regardless of whether it is a mirror or translucent object. Unfortunately, we do not know yet of a reliable mathematical measure which tells us whether the two-layer model is valid or not. We therefore impose a simple heuristic: if at a given point, the disparity values



Reflection coefficient	Point-wise result				Result after TV-L^2 denoising			
	Single orientation		Double orientation		Single orientation		Double orientation	
	mirror	reflection	mirror	reflection	mirror	reflection	mirror	reflection
$\alpha = 0.1$	0.0034	0.7409	0.0078	0.1191	0.0025	0.7392	0.0036	0.09924
$\alpha = 0.3$	0.0236	0.5994	0.0061	0.0349	0.0086	0.6273	0.0032	0.02371
$\alpha = 0.5$	0.0869	0.3735	0.0066	0.0236	0.0252	0.5111	0.0036	0.01377
$\alpha = 0.7$	0.1807	0.1547	0.0101	0.0239	0.1434	0.1821	0.0060	0.01053
$\alpha = 0.9$	0.2579	0.0365	0.0389	0.0473	0.2557	0.0312	0.0249	0.00980

Figure 39: Influence of reflectivity on accuracy. The table shows mean squared disparity error in pixels of the single and double orientation model for both the mirror plane as well as the reflection. While the single orientation model shifts from reconstruction of mirror to reflection with growing reflectivity α, the double orientation model can still reconstruct both when even a human observer has difficulties separating them. The images show the point-wise results.

of horizontal and vertical EPIs agree up to a small error for both the primary and secondary orientation, we flag the double orientation model as valid, and choose its contribution in the disparity maps. Otherwise, we choose the estimate from the single orientation model.

5.2.5 Results

We compare our method primarily to the single orientation method (Wanner et al. [100] and section 5.1) based on the first order structure tensor, which is similar in spirit and an initial step in our algorithm in any case. However, it is clear that any multi-view stereo method will have similar problems as the single orientation method if the underlying model is also the Lambertian world.

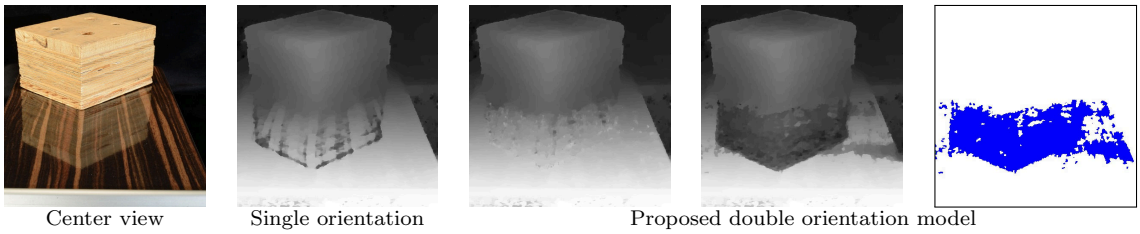


Figure 40: In the absence of a structured background, the reflecting surface can of course only reliably be detected where a reflection of a foreground object is visible. The blue region indicates where the double orientation model returns valid results.

5.2.5.1 Synthetic Data Sets Figure 39 shows reconstruction accuracy on a synthetic light field with varying amounts of reflectivity α. The scene was ray-traced in a way which exactly fits the image formation model. As expected, the disparity reconstructed with the single orientation model is close to the disparity of the mirror surface if α is small, and close to the disparity of the reflection if α is large. In between, the result is a mixture between the two, depending on whose texture is stronger. In contrast, the double orientation model can reliably reconstruct both reflection as well as mirror surface for the full range of reflectivities α, even when it is already difficult for a human to still observe both. While the point-wise results are already very accurate, they are still quite noisy and can be greatly improved by adding a small amount of TV-L^2 denoising [20]. We deliberately do not employ more sophisticated global optimization in this step to showcase only the raw output from the model and what is possible at interactive performance levels. For all of the light fields shown, at image resolutions upwards of 512×512 with 9×9 views, the point-wise disparity computation for the whole center view takes less than 1.5 seconds on an nVidia GTX 680 GPU.

5.2.5.2 Real-World Data Sets In Figures 41, 40, and 42, we show reconstruction results for light fields recorded with our gantry, see Figure 36(b). Each one has 9×9 views at resolutions between 0.5 and 1 mega-pixels. For both reflective and transparent surfaces, a reconstruction of a single disparity based on the Lambertian assumption produces major artifacts and is unusable in the region of the surface. In contrast, the proposed method always produces a very reliable estimate for the primary surface, as well as a reasonably accurate one for the reflected or transmitted objects, respectively. For the results in the figures, we employed a global optimization scheme [74, 100] to reach maximum possible quality, which takes about 3 minutes per disparity map. The same scheme and parameters were used for both methods and all data sets. To show what is possible in near real-time, we also provide the raw point-wise results in the additional material.
The results show that certain apparent limitations of the model are not practically

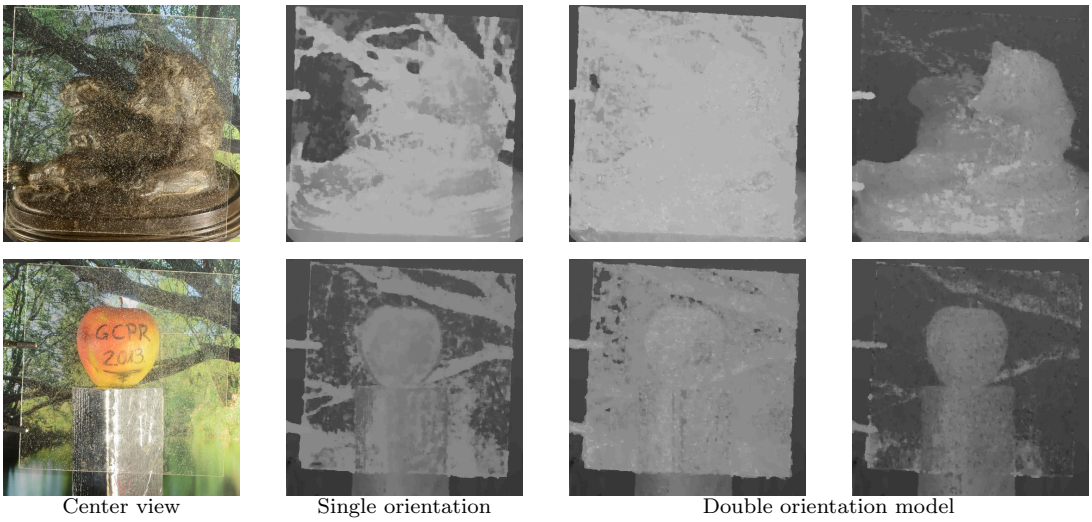


Figure 41: *Reconstructing a transparent surface.* The single orientation model cannot distinguish the two signals from the dirty glass surface and the objects behind it. In contrast, multi-orientation analysis correctly separates both layers.

relevant. In particular, reflectivity α is certainly not constant everywhere due to influences of e.g. the Fresnel term, but since all estimates are strictly local and the angular range small, the variations do not seem to impact the final result by much. A stronger limitation, however, is the planarity of the reflecting or transparent surface. We predict that it can be considerably weakened, since the main assumption of the existence of an object "behind" the primary surface (which is of course only virtual in case of a mirror) also holds for more general geometries. However, exploring this direction is left for future work.

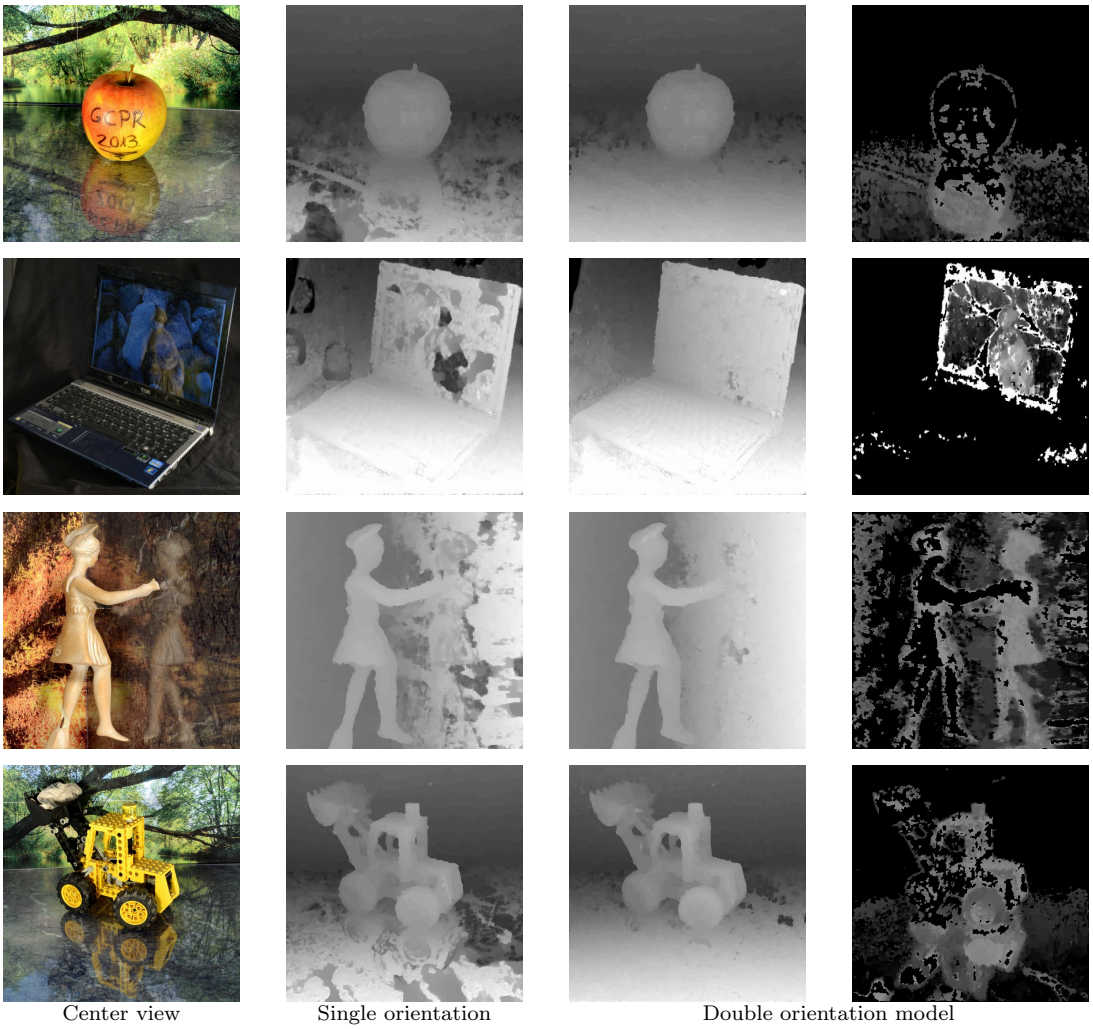


Figure 42: *Reconstructing a mirror.* Like multi-view stereo algorithms, the single orientation model cannot distinguish the two signals from mirror plane and reflection and reconstructs erroneous disparity for the mirror plane. In contrast, the proposed double orientation analysis correctly separates the data for the mirror plane from the reflection. The reflection channel is masked out where the double orientation model does not return valid results as specified in section 5.2.4, and the results for this channel have been increased in brightness and contrast for better visibility (raw results and many more data sets can be observed in the additional material).

6 Inverse Problems on Ray Space

In this section, we discuss some applications showing that when using light fields and their inherently available depth information, much better results can be achieved compared to classical approaches. The first application is an adaptation of super-resolution techniques to light fields which additionally is – as a side effect – a proof for the high accuracy of the orientation analysis because a necessary condition for the proposed super-resolution algorithm are depth maps of sub-pixel accuracy.
Another application is object segmentation where we will see that by adapting classical methods to light fields we can improve segmentation accuracy compared to segmentation using single images.

6.1 Spatial and Viewpoint Superresolution

Here, we propose a variational model for the synthesis of super-resolved novel views. The theoretical background of the variational methods used in this section is the work of Dr. Bastian Goldlücke. Fast GPU implementations of the algorithms can be found in his open source library *cocolib* [37]. The content is already published in Wanner et al. [101] and Wanner et al. [103].

Since the model is continuous, we will be able to derive Euler-Lagrange equations which correctly take into account foreshortening effects of the views caused by variations in the scene geometry. This makes the model essentially parameter-free. The framework is in the spirit of [38], which computes super-resolved textures for a 3D model from multiple views, and shares the same favorable properties. However, it has substantial differences, since we do not require a complete 3D geometry reconstruction and costly computation of a texture atlas. Instead, we only make use of disparity maps on the input images, and model the super-resolved novel view directly.

The following mathematical framework is formulated for views with arbitrary projections. However, an implementation in this generality would be quite difficult to achieve. We therefore specialize to the scenario of a 4D light field in the subsequent section, and leave a generalization of the implementation for future work.

For the remainder of the section, assume we have images $v_i : \Omega_i \to \mathbb{R}$ of a scene available, which are obtained by projections $\pi_i : \mathbb{R}^3 \to \Omega_i$. Each pixel of each image stores the integrated intensities from a collection of rays from the scene. This sub-sampling process is modeled by a blur kernel b for functions on Ω_i, and essentially characterizes the point spread function for the corresponding sensor element. It can be measured for a specific imaging system [8]. In general, the kernel may depend on the view and even on the specific location in the images. We omit the dependency here for simplicity of notation.

The goal is to synthesize a view $u : \Gamma \to \mathbb{R}$ of the light field from a novel view point, represented by a camera projection $\pi : \mathbb{R}^3 \to \Gamma$, where Γ is the image plane of the

novel view. The basic idea of super-resolution is to define a physical model for how the sub-sampled images v_i can be explained using high-resolution information in u, and then solve the resulting system of equations for u. This inverse problem is ill-posed, and is thus reformulated as an energy minimization problem with a suitable prior or regularizer on u.

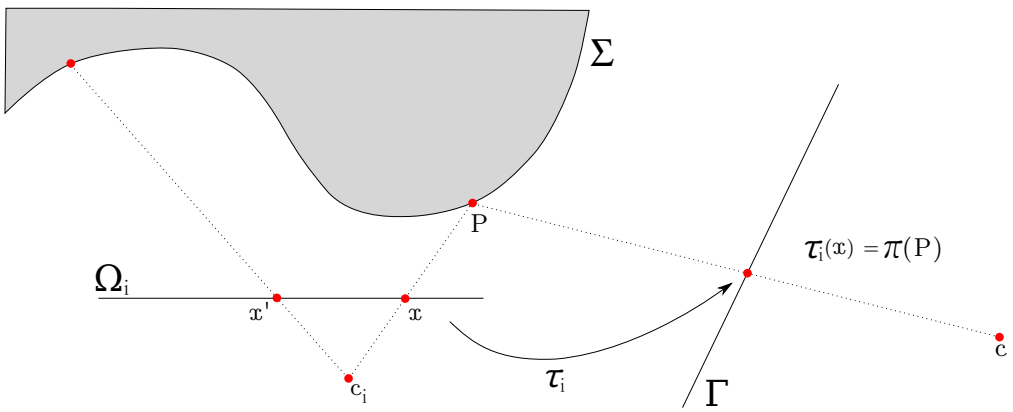


Figure 43: Transfer map τ_i from an input image plane Ω_i to the image plane Γ of the novel view point. The scene surface Σ can be inferred from the depth map on Ω_i. Note that not all points $x \in \Omega_i$ are visible in Γ due to occlusion, which is described by the binary mask m_i on Ω_i . Above, $m_i(x) = 1$ while $m_i(x') = 0$.

6.1.1 Image Formation and Model Energy

In order to formulate the transfer of information from u to v_i correctly, we require geometry information [19]. Thus, we assume we know (previously estimated) depth maps d_i (see section 5) for the input views. A point $x \in \Omega_i$ is then in one-to-one correspondence with a point P which lies on the scene surface $\Sigma \subset \mathbb{R}^3$. The color of the scene point can be recovered from u via $u \circ \pi(P)$, provided that x is not occluded by other scene points, see figure 43.

The process explained above induces a backwards warp map $\tau_i : \Omega_i \to \Gamma$ which tells us where to look on Γ for the color of a point, as well as a binary occlusion mask $m_i : \Omega_i \to \{0, 1\}$ which takes the value 1 if and only if a point in Ω_i is also visible in Γ. Both maps only depend on the scene surface geometry as seen from v_i, i.e. the depth map d_i. The different terms and mappings appearing above and in the following are visualized for an example light field in figure 44.

Having computed the warp map, one can formulate a model of how the values of v_i within the mask can be computed, given a high-resolution image u. Using the down-sampling kernel, we obtain $v_i = b * (u \circ \tau_i)$ on the subset of Ω_i where $m_i = 1$, which consists of all points in v_i which are also visible in u. Since this equality will not be satisfied exactly due to noise or inaccuracies in the depth map, we instead propose to minimize

the energy

$$E(u) = \sigma^2 \int_\Gamma |Du| + \sum_{i=1}^{n} \underbrace{\frac{1}{2} \int_{\Omega_i} m_i (b * (u \circ \tau_i) - v_i)^2 \, \mathrm{d}x}_{=:E^i_{\text{data}}(u)} . \tag{52}$$

which is the MAP [30] (maximum a posteriori estimation) estimate under the assumption

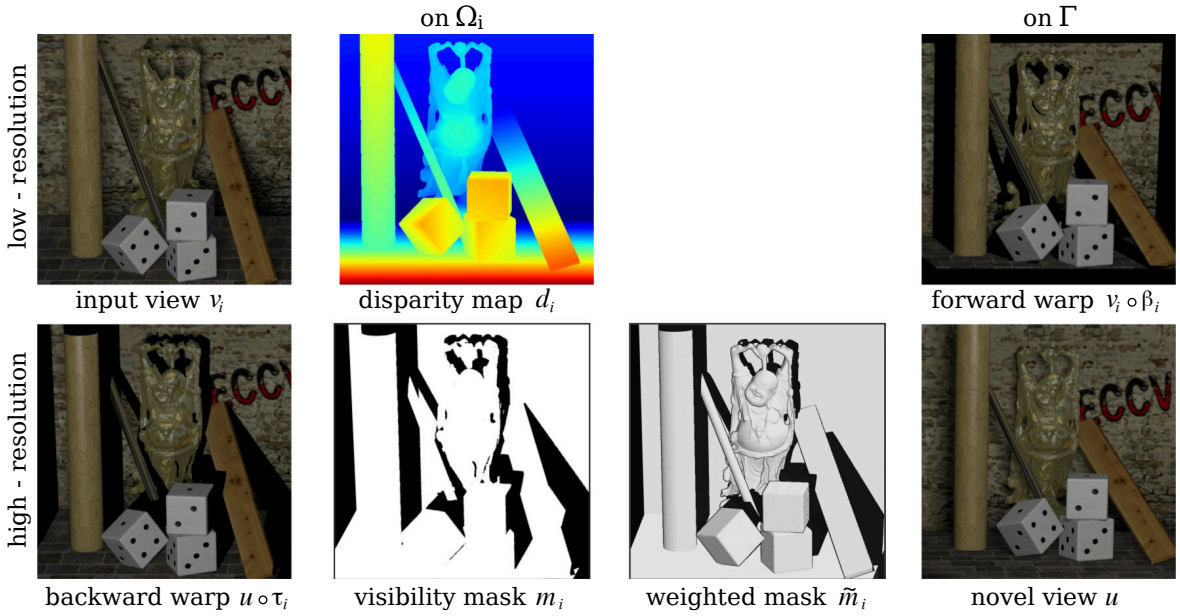


Figure 44: Illustration of the terms in the super-resolution energy. The figure shows the ground truth depth map for a single input view and the resulting mappings for forward- and backward warps as well as the visibility mask m_i. White pixels in the mask denote points in Ω_i which are visible in Γ as well.

of Gaussian noise with standard deviation σ on the input images. It resembles a classical super-resolution model [8], which is made slightly more complex by the inclusion of the warp maps and masks.

In the energy, formulated in equation 52, the total variation acts as a regularizer or objective prior on u. Its main tasks are to eliminate outliers and enforce a reasonable in-painting of regions for which no information is available, i.e. regions which are not visible in any of the input views. It could be replaced by a more sophisticated prior for natural images, however, the total variation [78] leads to a convex model which can be very efficiently minimized. Furthermore, the regularization weight λ, which is the only free parameter of the model, is usually set very low in order to not destroy any details in the reconstruction. We have it at 0.0001 in all experiments, which makes the exact choice of regularizer not very significant.

6.1.2 Functional Derivative

The functional derivative for the inverse problem above is required in order to find solutions. It is well-known in principle, but one needs to take into account complications caused by the different domains of the integrals. Note that τ_i is one-to-one when restricted to the visible region $V_i := \{m_i = 1\}$, thus we can compute an inverse *forward warp map* $\beta_i := (\tau_i|_{V_i})^{-1}$, which we can use to transform the data term integral back to the domain Γ, see figure 44. We obtain for the derivative of a single term of the sum in equation 52

$$dE^i_{\text{data}}(u) = |\det D\beta_i| \ \left(m_i \bar{b} * (b * (u \circ \tau_i) - v_i)\right) \circ \beta_i. \tag{53}$$

The determinant is introduced by the variable substitution of the integral during the transformation. A more detailed derivation for a structurally equivalent case can be found in [38].

The term $|\det D\beta_i|$ in equation 53 introduces a point-wise weight for the contribution of each image to the gradient descent. However, β_i depends on the depth map on Γ, which needs to be inferred and is not readily available. Furthermore, for efficiency it needs to be pre-computed, and storage would require another high-resolution floating point matrix per view. Memory is a bottleneck in our method, and we need to avoid this. For this reason, it is much more efficient to transform the weight to Ω_i and multiply it with m_i to create a single weighted mask. Note that

$$|\det D\beta_i| = \left|\det D\tau_i^{-1}\right| = |\det D\tau_i|^{-1} \circ \beta_i. \tag{54}$$

Thus, we obtain a simplified expression for the functional derivative,

$$dE^i_{\text{data}}(u) = \left(\tilde{m}_i \ \bar{b} * (b * (u \circ \tau_i) - v_i)\right) \circ \beta_i \tag{55}$$

with $\tilde{m}_i := m_i \, |\det(D\tau_i)|^{-1}$. An example weighted mask is visualized in figure 44. In total, only the weighted mask $\tilde{m}_i$ needs to be pre-computed and stored for each view. In the scenario we present in the next section, the warp maps will be simple and can be computed on the fly from just the disparity map.

6.1.3 Specialization to 4D Light Fields

The model introduced until now is hard to implement efficiently in fully general form. Thus we focus on the setting of a 4D light field, where we can make a number of significant simplifications. The main reason is that the warp maps between the views are given by parallel translations in the direction of the view point change. The amount of translation is proportional to the disparity of a pixel, which is in one-to-one correspondence with the depth, as explained in sections 2.2, 5.1.2. How the disparity maps are obtained does not matter, but in this work, naturally, they will be computed using the technique described in section 5.

6.1.4 View Synthesis in the Light Field Plane

The warp maps required for view synthesis become particularly simple when the target image plane Γ lies in the common image plane Ω of the light field, and π resembles the corresponding light field projection through a focal point $c \in \Pi$. In this case, τ_i is simply given by a translation proportional to the disparity,

$$\tau_i(x) = x + d_i(x)(c - c_i), \tag{56}$$

see figure 45. Thus, one can compute the weight in equation 55 to be

$$|\det D\tau_i|^{-1} = |1 + \nabla d_i \cdot (c - c_i)|^{-1} \tag{57}$$

There are a few observations to make about this weight. Disparity gradients which are not aligned with the view translation $\Delta c = c - c_i$ do not influence it, which makes sense since it does not change the angle under which the patch is viewed. Disparity gradients which are aligned with Δc and tend to infinity lead to a zero weight, which also makes sense since they lead to a large distortion of the patch in the input view and thus unreliable information.

A very interesting result is the location of maximum weight. The weights become larger when $\Delta c \cdot \nabla d_i$ approaches -1. An interpretation can be found in figure 45. If $\Delta c \cdot \nabla d_i$ gets closer to -1, then more information from Ω_i is being condensed onto Γ, which means that it becomes more reliable and should be assigned more weight. The extreme case is a line segment with a disparity gradient such that $\Delta c \cdot \nabla d_i = -1$, which is projected onto a single point in Γ. In this situation, the weight becomes singular. This does not pose a problem: From a theoretical point of view, the set of singular points is a null set according to the theorem of Sard [80], and thus not seen by the integral. From a practical point of view, all singular points lead to occlusion and the mask m_i is zero anyway. Note that formula 57 is non-intuitive, but the correct one to use when geometry is taken into account. We have not seen anything similar being used in previous work. Instead, weighting factors for view synthesis are often imposed according to measures based on distance to the interpolated rays or matching similarity scores, which are certainly working, but also somewhat heuristic strategies [35, 51, 59, 76].

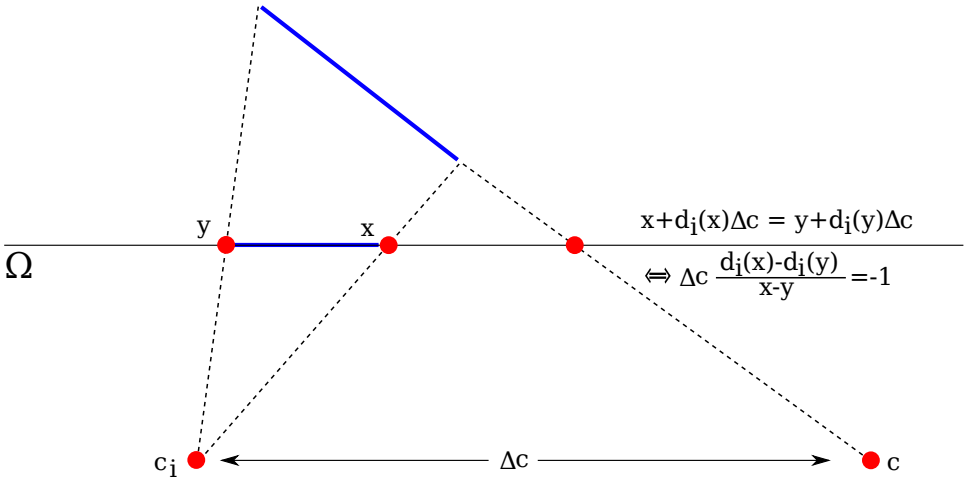


Figure 45: The slope of the solid blue line depends on the disparity gradient in the view v_i. If $\Delta c \cdot \nabla d_i = -1$, then the line is projected onto a single point in the novel view u.

6.1.5 Results

For the optimization of the (convex) energy in equation 52, we transform the gradient to the space of the target view via equation 55, discretize, and employ the fast iterative shrinkage and thresholding algorithm (FISTA) found in [9].

In order to demonstrate the validity and robustness of our algorithm, we perform extensive tests on our synthetic light fields, where we have ground truth available, as well as on real-world data sets from a plenoptic camera. As a by-product, this establishes again that disparity maps obtained by our proposed method in section 5 have subpixel accuracy, since this is a necessary requirement for super-resolution to work.

View Interpolation and Superresolution In a first set of experiments, we show the quality of view interpolation and super-resolution, both with ground truth as well as estimated disparity. In table 47, we synthesize the center view of a light field with our algorithm using the remaining views as input, and compare the result to the actual view. For the down-sampling kernel b, we use a simple box filter of size equal to the down-sampling factor, so that it fits exactly on a pixel of the input views. We compute results both with ground truth disparities to show the maximum theoretical performance of the algorithm, as well as for the usual real-world case that disparity needs to be estimated. This estimation is performed using the local method described in section 5.1.2.1, so requires less than five seconds for all of the views. Synthesizing a single super-resolved view requires about 15 seconds on an nVidia GTX 580 GPU.
In order to test the quality of super-resolution, we compute the 3×3 super-resolved center view and compare with ground truth. For reference, we also compare the result of bilinear interpolation (IP) as well as TV-zooming [20] of the center view synthesized

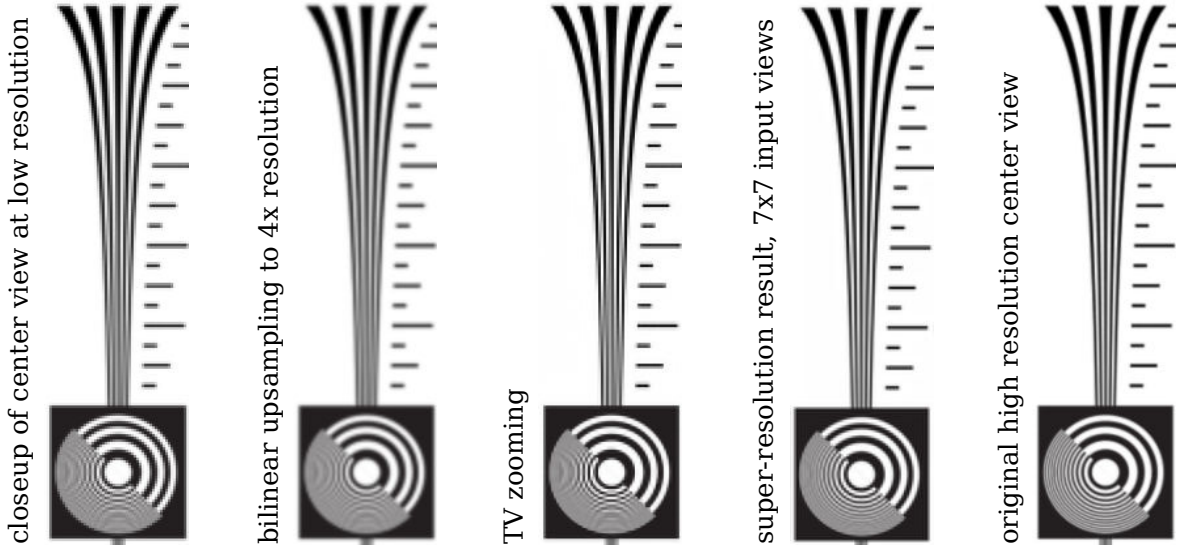


Figure 46: Comparison of the different up-sampling schemes on the light field of a resolution chart. Input resolution is 512×512, which is $4\times$ up-sampled. From left to right: original low resolution input, bilinear up-sampling, TV zooming [20], our result, the original 1024×1024 center view for comparison. All images shown are closeups.

in the first experiment. While the reconstruction with ground truth disparities is very precise, we can see that in the case of estimated disparity, the result strongly improves with larger angular resolution due to better disparity estimates (compare figure 30). Super-resolution is superior to both competing methods. This also emphasizes the sub-pixel accuracy of the disparity maps, since without accurate matching, super-resolution would not be possible. Figures 48 and 46 show closeup comparison images of the input light fields and up-sampled novel views obtained with different strategies. At this zoom level, it is possible to observe increased sharpness and details in the super-resolved results. Figure 46 indicates that the proposed scheme also produces the least amount of artifacts.

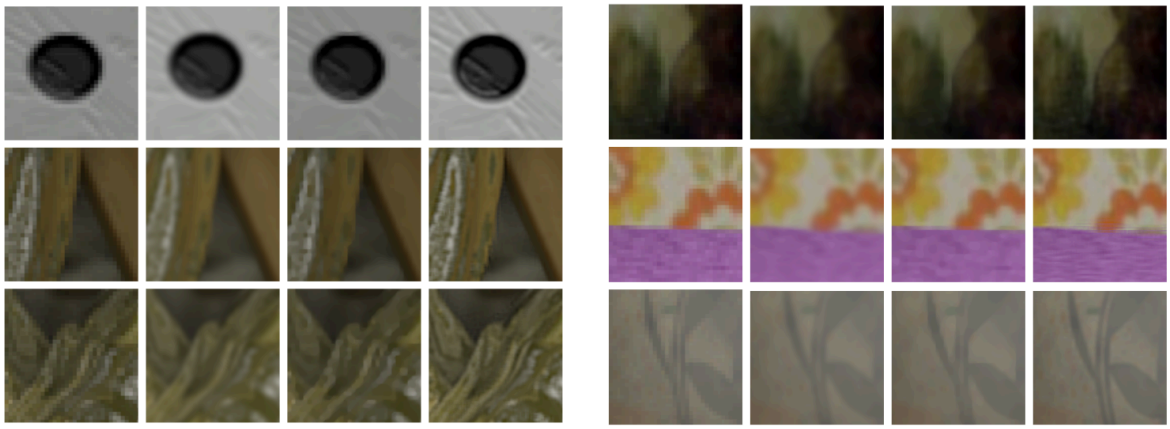
Figures 51 and 50 show the results of the same set of experiments for two real-world scenes captured with the Raytrix plenoptic camera. The plenoptic camera data was transformed to the standard representation as an array of 9×9 views using the method in section 3. Since no ground truth for the scene is available, the input views were down-sampled to lower resolution before performing super-resolution and compared against the original view. We can see that the proposed algorithm allows to accurately reconstruct both sub-pixel disparity as well as a high-quality super-resolved intermediate view.

	Conehead				*Buddha*				*Mona*				
Views	$1x1$	$3x3$	TV	IP	$1x1$	$3x3$	TV	IP	$1x1$	$3x3$	TV	IP	
5×5	31.6	29.3	27.4	26.5	32.2	28.9	27.5	26.5	30.1	28.3	27.4	26.4	GT
9×9	31.6	29.4	27.5	26.5	32.2	29.1	27.5	26.5	30.0	28.3	27.4	26.3	
17×17	31.2	30.4	27.3	26.0	31.8	30.2	28.8	27.2	30.2	28.9	27.8	26.5	
5×5	31.1	29.3	27.1	25.8	28.0	28.9	25.8	24.3	26.4	28.3	25.7	23.8	ED
9×9	31.4	29.4	27.6	26.2	30.7	29.1	28.9	27.7	28.9	28.3	26.8	25.1	
17×17	31.5	30.9	25.9	24.3	31.4	29.5	27.9	26.8	29.5	28.3	27.1	25.8	

Figure 47: Reconstruction error for the data sets obtained with a ray-tracer. The table shows the PSNR of the center view without super-resolution, at super-resolution magnification 3×3, and for bilinear interpolation (IP) and TV-Zooming (TV) [20] to 3×3 resolution as a comparison. The set of experiments is run with both ground truth (GT) and estimated disparities (ED). The estimation error for the disparity map can be found in figure 30. Input image resolution is 384×384.

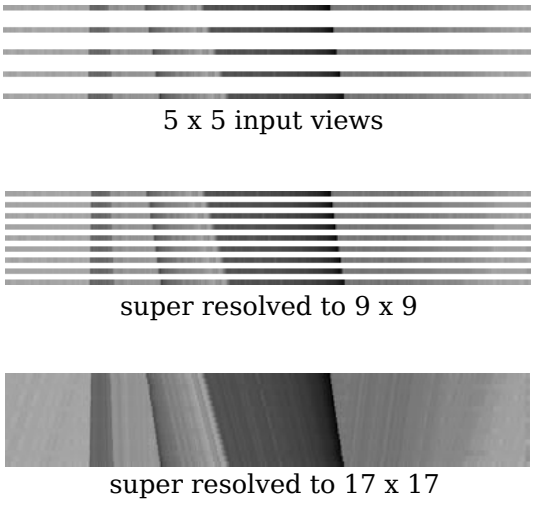
Disparity Refinement As we have seen in figure 49, the disparity estimate is more accurate when the angular sampling of the light field is more dense. An idea is therefore to increase angular resolution and improve the disparity estimate by synthesizing intermediate views.

We first synthesize novel views to increase angular resolution by a factor of 2 and 4. Figure 49 shows resulting epipolar plane images, which can be seen to be of high quality with accurate occlusion boundaries. Nevertheless, it is highly interesting that the quality of the disparity map increases significantly when recomputed with the super-resolved light field, figure 49. This is a striking result, since one would expect that the intermediate views reflect the error in the original disparity maps. However, they actually provide more accuracy than a single disparity map, since they represent a consensus of all input views. Unfortunately, due to the high computational cost, this is not a really viable strategy in practice.



(a) Buddha (b) Mona

Figure 48: Closeups of the up-sampling results for the light fields generated with a ray tracer. From left to right: low-resolution center view (not used for reconstruction), high resolution center view obtained by bilinear interpolation of a low-resolution reconstruction from 24 other views, TV-Zooming [20], super-resolved reconstruction. The super-resolved result shows increased sharpness and details.



Views	*Cone*	*Buddha*	*Mona*
input 5×5	4.534	0.883	2.125
SR 9×9	1.084	0.559	1.058
input 9×9	0.066	0.080	0.192
SR 17×17	0.044	0.066	0.105

Disparity MSE

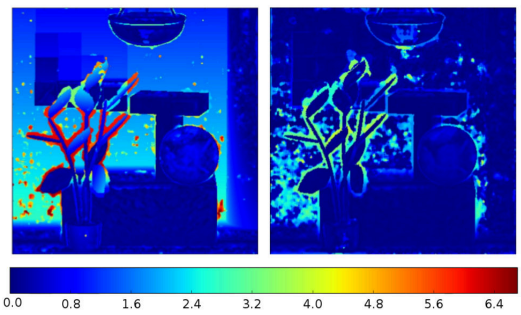
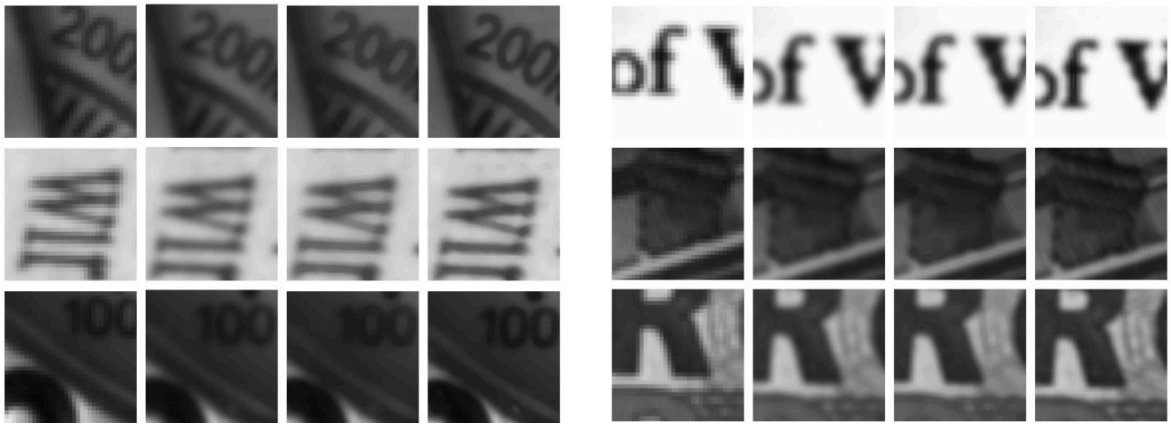


Figure 49: Left: Up-sampling of epipolar plane images (EPIs). From Top to bottom the five layers of an epipolar plane image of the input data set with 5×5 views, the super resolved 7×7 and the super resolved 17×17 views are depicted. We generate intermediate views using our method to achieve angular super-resolution. One can observe the high quality and accurate occlusion boundaries of the resulting view interpolation. **Right:** Indeed, they are accurate enough such that using the up-sampled EPIs leads to a further improvement in depth estimation accuracy. Here the mean square errors for all angular resolutions as well as the color coded error distribution of the depth error before and after super-resolution are shown.



(a) Demo (b) Motor

Figure 50: Super-resolution view synthesis using light fields from a plenoptic camera. Scenes were recorded with a Raytrix camera at a resolution of 962×628 and super-resolved by a factor of 3×3. The light field contains 9×9 views. From left to right: low-resolution center view (not used for reconstruction), high resolution center view obtained by bilinear interpolation of a low-resolution reconstruction from 24 other views, TV-Zooming [20], super-resolved reconstruction. One can find additional detail, for example the diagonal stripes in the Euro note, which were not visible before.

Method	*Demo*	*Motor*
1×1	36.91	35.36
3×3	30.82	31.72
TV	25.21	24.99
IP	23.89	22.84

Reconstruction PSNR

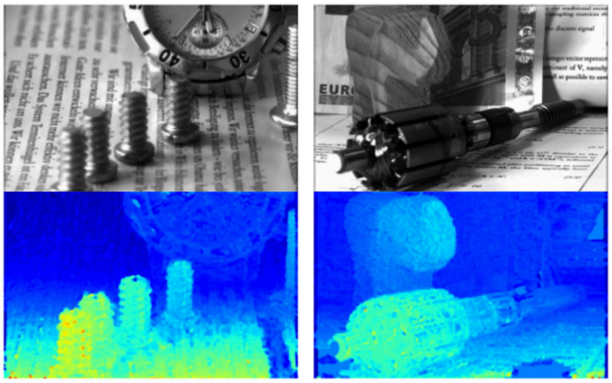


Figure 51: Reconstruction error for light fields captured with the Raytrix plenoptic camera. The table shows PSNR for the reconstructed input view at original resolution as well as 3×3 super-resolution and 3×3 interpolation (IP) and TV-Zooming (TV) [20] for comparison.

6.2 Rayspace Segmentation

Here we present the first variational framework for multi-label segmentation on the ray space of 4D light fields. For traditional segmentation of single images, features need to be extracted from the 2D projection of a three-dimensional scene. The associated loss of geometry information can cause severe problems, for example if different objects have a very similar visual appearance. In this section, we show that using a light field instead of an image not only enables to train classifiers which can overcome many of these problems, but also provides an optimal data structure for label optimization by implicitly providing scene geometry information. Thus it is possible to consistently optimize label assignment over all views simultaneously.

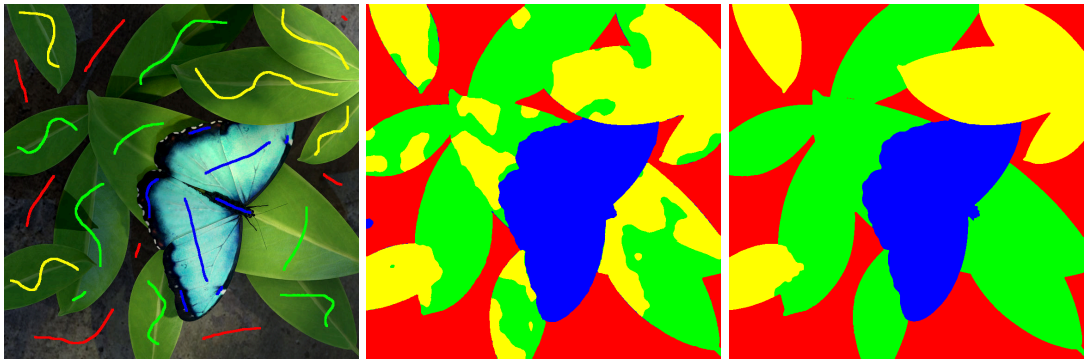
Recent developments in light field acquisition systems [14, 64, 69, 72] strengthen the prediction that we might soon enter an age of light field photography [57]. Since compared to a single image, light fields increase the content captured of a scene by directional information, they require an adaptation of established algorithms in image processing and computer vision as well as the development of completely novel techniques. Here, we develop methods for training classifiers on features of a light field, and for consistently optimizing label assignments to rays in a global variational framework. The ray space of the light field is considered four-dimensional, parametrized by the two points of intersection of a ray with two parallel planes, so that the light field can be considered as a collection of planar views, see figures 6 and 5.

Due to this planar sampling, 3D points are projected onto lines in cross-sections of the light field called epipolar-plane images (section 2.3). In recent works, it was shown that robust disparity reconstruction is possible by analyzing this line structure [11, 16, 27, 100] (see also section 5). In contrast to traditional stereo matching, no correspondence search is required, and floating-point precision disparity data can be reconstructed at a very small cost.

From the point of view of segmentation, this means that in light fields, we have access to more than the color of a pixel and information about the neighboring image texture. Additionally, we can assume that disparity is readily available as a feature. Disparity turns out to be highly effective for increasing the prediction quality of a classifier. As long as the inter-class variety of imaged objects is high and the intra-class variation is low, state of the art classifiers can easily discriminate different objects. However, separating for example background and foreground leafs (example in figure 52) poses a more difficult task.

In general, there is no easy way to alleviate issues like this using only single images. However, for a classifier which also has geometry based features available, similar looking objects are readily distinguishable if their geometric features are separable.

In the following, we will show that light fields are ideally suited for image segmentation. One reason is that geometry is an inherent characteristic of a light field, and thus we



Input user scribbles | single view labeling | ray space labeling

Figure 52: Multi-label segmentation with light field features and disparity-consistent regularization across ray space leads to results which are superior to single-view labeling.

can use disparity as a very helpful additional feature. While this has already been realized in related work on e.g. multi-view co-segmentation [50] or segmentation with depth or motion cues, which are in many aspects similar to disparity [31, 92], light fields also provide an ideal structure for a variational framework which readily allows consistent labelling across all views, and thus increases the accuracy of label assignments dramatically.

6.2.1 Regularization on Ray Space

In segmentation problems, when one wants to label rays according to e.g. the visible object class, the unknown function on ray space ultimately reflects a property of scene points. In consequence, all the rays which view the same scene point have to be assigned the same function value. Equivalent to this is to demand that the function must be consistent with the structure on the epipolar plane images. In particular, except at depth discontinuities, the value of such a function is not allowed to change in the direction of the epipolar lines, which are induced by the disparity field.

The above considerations give rise to a regularizer $J_{\lambda\mu}(\boldsymbol{U})$ for vector-valued functions $\boldsymbol{U} : \mathcal{R} \to \mathbb{R}^n$ on ray space. It can be written as the sum of contributions for the regularizers on all epipolar plane images as well as all the views,

$$\begin{aligned} J_{\lambda\mu}(\boldsymbol{U}) &= \mu J_{xs}(\boldsymbol{U}) + \mu J_{yt}(\boldsymbol{U}) + \lambda J_{st}(\boldsymbol{U}) \\ \text{with } J_{xs}(\boldsymbol{U}) &= \int J_\rho(\boldsymbol{U}_{x^*,s^*})\; d(x^*, s^*), \\ J_{yt}(\boldsymbol{U}) &= \int J_\rho(\boldsymbol{U}_{y^*,t^*})\; d(y^*, t^*), \\ \text{and } J_{st}(\boldsymbol{U}) &= \int J_V(\boldsymbol{U}_{s^*,t^*})\; d(s^*, t^*), \end{aligned} \tag{58}$$

where the anisotropic regularizers J_ρ act on 2D epipolar plane images, and are defined

such that they encourage smoothing in the direction of the epipolar lines. This way, they enforce consistency of the function $\boldsymbol{U}$ with the epipolar plane image structure. For a detailed definition, we refer to our related work [39]. The spatial regularizer J_V encodes the label transition costs, as we will explore in more detail in the next section. Finally, the constants $\lambda > 0$ and $\mu > 0$ are user-defined and adjust the amount of regularization on the separate views and epipolar plane images, respectively.

6.2.2 Optimal Label Assignment on Ray Space

In this section, we introduce a new variational labeling framework on ray spaces. Its design is based on the representation of labels with indicator functions [22, 53, 111], which leads to a convex optimization problem. We can use the efficient optimization framework presented in [39] to obtain a globally optimal solution to the convex problem, however, as usual we need to project back to indicator functions and only end up within a (usually small) posterior bound of the optimum.

The Variational Multi-Label Problem. Let Γ be the (discrete) set of labels, then to each label $\gamma \in \Gamma$ we assign a binary function $u_\gamma : \mathcal{R} \to \{0, 1\}$ which takes the value 1 if and only if a ray is assigned the label γ. Since the assignment must be unique, the set of indicator functions must satisfy the simplex constraint

$$\sum_{\gamma \in \Gamma} u_\gamma = 1. \tag{59}$$

Arbitrary spatially varying label cost functions c_γ can be defined, which penalize the assignment of γ to a ray $R \in \mathcal{R}$ with the cost $c_\gamma(R) \geq 0$.

Let $\boldsymbol{U}$ be the vector of all indicator functions. To regularize $\boldsymbol{U}$, we choose $J_{\lambda\mu}$ defined in equation 58. This implies that the labelling is encouraged to be consistent with the epipolar plane structure of the light field to be labelled. The spatial regularizer J_V needs to enforce the label transition costs. For the remainder of this work, we choose a simple weighted Potts penalizer [110]

$$J_V(\boldsymbol{U}_{s^*,t^*}) := \frac{1}{2} \sum_{\gamma \in \Gamma} \int_\Omega g \, |(Du_\gamma)_{s^*t^*}| \; d(x, y), \tag{60}$$

where g is a spatially varying transition cost. Since the total variation of a binary function equals the length of the interface between the zero and one level set due to the co-area formula [32], the factor $1/2$ leads to the desired penalization.

While we use the weighted Potts model in this work, the overall framework is by no means limited to it. Rather, we can use any of the more sophisticated regularizers proposed in the literature [22, 53], for example truncated linear penalization, Euclidean label distances, Huber TV or the Mumford-Shah regularizer. An overview as well as

further specializations tailored to vector-valued label spaces can be found in [94].

The space of binary functions over which one needs to optimize is not convex, since convex combinations of binary functions are usually not binary. We resort to a convex relaxation, which with the above conventions can now be written as

$$\underset{\boldsymbol{U}\in\mathcal{C}}{\operatorname{argmin}} \left\{ J_{\lambda\mu}(\boldsymbol{U}) + \sum_{\gamma\in\Gamma} \int_{\mathcal{R}} c_\gamma u_\gamma \; d(x,y,s,t) \right\}, \tag{61}$$

where $\mathcal{C}$ is the convex set of functions $\boldsymbol{U} = (u_\gamma : \mathcal{R} \to [0,1])_{\gamma\in\Gamma}$ which satisfy the simplex constraint equation 59. After optimization, the solution of equation 61 needs to be projected back onto the space of binary functions. This means that we usually do not achieve the global optimum of equation 61, but can only compute a posterior bound for how far we are from the optimal solution. An exception is the two-label case, where we indeed achieve global optimality via thresholding, since the anisotropic total variation also satisfies a co-area formula [111].

Optimization. Note that according to equation 58, the full regularizer $J_{\lambda\mu}$ which is defined on 4D ray space decomposes into a sum of 2D regularizers on the epipolar plane images and individual views, respectively. While solving a single saddle point problem for the full regularizer would require too much memory, it is feasible to iteratively compute independent descent steps for the data term and regularizer components.

The overall algorithm is detailed in [39]. Aside from the data term, the main difference here is the simplex constraint set for the primal variable $\boldsymbol{U}$. We enforce it with Lagrange multipliers in the proximity operators of the regularizer components, which can be easily integrated into the primal-dual algorithm [21]. An overview of the algorithm adapted to problem in equation 61 can be found in figure 53.

On our system equipped with an nVidia GTX 580 GPU, optimization takes about 1.5 seconds per label in Γ and per million rays in $\mathcal{R}$, i.e. about 5 minutes for our rendered data sets if the result for all views is desired. If only the result for one single view (i.e. the center one) is required, computation can be restricted to view points located in a cross with that specific view at the center. The result will usually be very close to the optimization over the complete ray space. While this compromise forfeits some information in the data it leads to significant speeds ups, for our rendered data sets to about 30 seconds.

6.2.3 Local Class Probabilities

We calculate the unary potentials c_γ in equation 61 from the negative log-likelihoods of the local class probabilities,

$$c_\gamma(R) = -\log p\big(\gamma|\mathbf{v}(R)\big), \tag{62}$$

To solve the multi-label problem in equation 61 on ray space, we initialize the unknown vector-valued function $\boldsymbol{U}$ such that the indicator function for the optimal point-wise label is set to one, and zero otherwise. Then we iterate

- data term descent: $U_\lambda \leftarrow U_\lambda - \tau c_\lambda$ for all $\lambda \in \Lambda$,
- EPI regularizer descent:

$$\boldsymbol{U}_{x^*s^*} \leftarrow \text{prox}_{\tau\mu J_\rho}(\boldsymbol{U}_{x^*s^*}) \text{ for all } (x^*, s^*),$$
$$\boldsymbol{U}_{y^*t^*} \leftarrow \text{prox}_{\tau\mu J_\rho}(\boldsymbol{U}_{y^*t^*}) \text{ for all } (y^*, t^*),$$

- spatial regularizer descent:

$$\boldsymbol{U}_{s^*t^*} \leftarrow \text{prox}_{\tau\lambda J_V}(\boldsymbol{U}_{s^*t^*}) \text{ for all } (s^*, t^*).$$

The proximation operators prox_J compute subgradient descent steps for the respective 2D regularizer, and enforce the simplex constraint in equation 59 for $\boldsymbol{U}$. The possible step size τ depends on the data term scale, in our experiments $\tau = 0.1$ lead to reliable convergence within about 20 iterations.

Figure 53: Algorithm for the general multi-label problem in equation 61.

so that by solving equation 61, we obtain the maximum a-posteriori (MAP) solution [30] for the label assignment. The local class probabilities $p\big(\gamma|\mathbf{v}(R)\big) \in [0, 1]$ for the label γ, conditioned on a local feature vector $\mathbf{v}(R) \in \mathbb{R}^{|F|}$ for each ray $R \in \mathcal{R}$, are obtained by training a classifier on a user-provided partial labelling of the center view. As features, we use a combination of color, Laplace operator of the view, intensity standard deviation in a neighbourhood, Eigenvalues of the Hessian and the disparity computed on several scales. While our framework allows the use of arbitrary classifiers, we specialize in this thesis to a *Random Forest* [18]. These are becoming increasingly popular in image processing due to their wide applicability [26] and the robustness with regard to their hyper-parameters. Random Forests make use of *bagging* to reduce variance and avoid over-fitting. A decision forest is built from a number n of trees, which are each trained from a random subset of the available training samples. In addition to bagging, extra randomness is injected into the trees by testing only a subset of $m < |F|$ different features for their optimal split in each split node. The above internal random forest parameters were fixed to $m = \sqrt{|F|}$ and $n = 71$ in our experiments.

Each individual tree is now built by partitioning the set of training samples recursively into smaller subsets, until the subsets become either class-pure or smaller than a given minimal split node size. The partitioning of the samples is achieved by performing a line search over all possible splits along a number of different feature axes for the optimal *Gini-impurity* of the resulting partitions, and repeating this process for the child partitions recursively. In each node, the chosen feature and the split value of that

Features used	Classifier		
	IMG	IMG-D	IMG-GT
RGB value	✓	✓	✓
Intensity standard deviation (in local neighbourhood)	✓	✓	✓
Eigenvalues of Hessian	✓	✓	✓
Laplace operator	✓		
Estimated disparity		✓	
Ground truth disparity			✓

Figure 54: *Combination of features used for the experiments in this paper. The individual scales of the features were determined via a grid search to find optimal parameters for each dataset individually.*

feature are stored. After building a single tree, the class distribution of the samples in each leaf node is stored and used at prediction time to obtain the conditional class probability of samples that arrive at that particular leaf node. The leaf node with which a prediction-sample is associated is determined by comparing the nodes' split value for the split feature with the feature vector entry of a sample. Depending on whether the sample value is smaller (larger) than the node value, the sample is passed to the left (right) child of the split node, until a leaf node is reached.

Finally, the ensemble of decision tree classifiers is used to calculate the local class probability of unlabeled pixels by averaging their votes. In our experiments, we achieved total run-times for training and prediction between one and 5 minutes, depending on the size of the light field and the number of labels. However, we did not yet parallelize the local predictions, which is easily possible and would make computation much more efficient.

6.2.4 Experiments

In this section, we present the results of our multi-label segmentation framework on a variety of different data sets. To explore the full potential of our approach, we use computer graphics generated light fields rendered with the open source software Blender [77], which provides complete ground truth for depth values and labels. In addition, we show that the approach yields very good results on real world data obtained with a plenoptic camera and a gantry, respectively. A subset of the views in the real-world data sets were manually labeled in order to provide ground truth to quantify the results.

There are two main benefits of labeling in light fields. First, we demonstrate the usefulness of disparity as an additional feature for training a classifier, and second, we show the improvements from the consistent variational multi-label optimization on ray space.

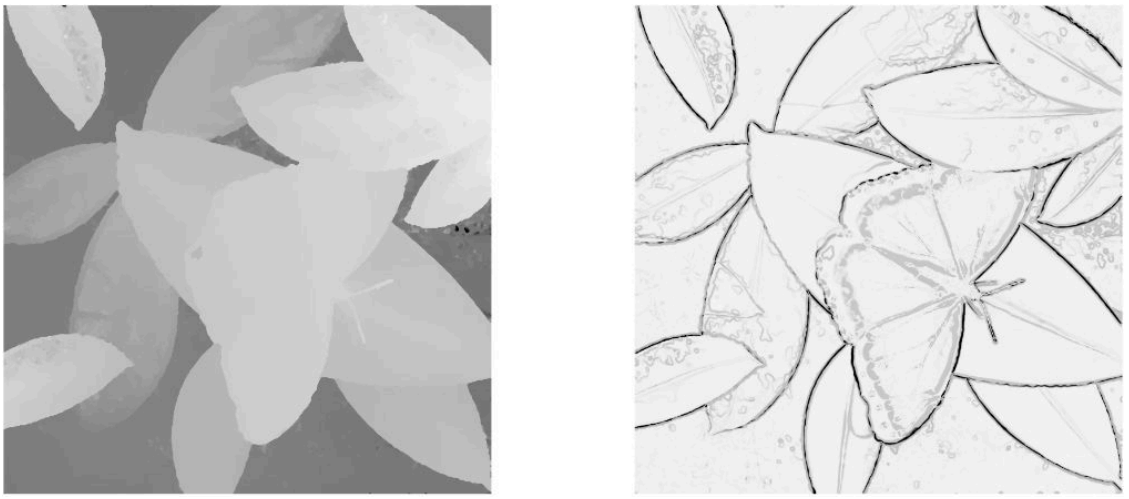


Figure 55: Depth estimated using the method in section 5 and spatial regularizer weight computed according to equation 63 for the light field view shown in figure 52.

Disparity as a Feature. The first step of the proposed segmentation framework does not differ from single image segmentation using a random forest. The user selects an arbitrary view from the light field, adds scribbles for the different labels, and chooses suitable features as well as the scales on which the features should be calculated. The classifier is then trained on this single view and, in a second step, used to compute local class probabilities for all views of the entire light field.

In advance, we have tested variations of common features for interactive image segmentation on our data sets to find a suitable combination of features which yields good results on single images. The optimal training parameters were determined using a grid search over the minimum split node size as well as the feature combinations and their scales for each data set individually. The number of different scales we used for each feature was fixed to four. This way, we can guarantee optimal results of the random forest classifier for all data sets and feature combinations, which ensures a meaningful assessment of the effects of our new ray space features.

Throughout the remainder of this section, we use the three different sets of features detailed in figure 54. The classifier *IMG* uses only classical single-view features, while *IMG-D* and *IMG-GT* employ in addition estimated and ground truth disparity, respectively, the latter of course only if available. Estimated disparity maps were obtained using our method in section 5 and are overall of very good quality, see figure 55. The achieved accuracy and the boundary recall for purely point-wise classification using the three classifiers above are listed in the table in figure 56. Sample segmentations for our data sets can be viewed in figure 58. It is obvious that the features extracted from the light field improve the quality of a local classifier significantly for difficult problem instances.

Data set	Classifier IMG acc	IMG br	IMG-D acc	IMG-D br	IMG-GT acc	IMG-GT br
synthetic data sets						
Buddha	93.5	6.4	96.7	39.6	98.6	43.1
Garden	95.1	54.8	96.7	51.1	96.9	53.3
Papillon 1	98.6	59.3	98.3	57.4	99.0	78.9
Papillon 2	90.8	16.7	96.5	33.1	99.1	73.0
Horses 1	93.2	13.4	94.3	34.9	98.3	48.7
Horses 2	94.6	15.9	95.3	36.8	98.5	50.9
StillLife 1	98.6	36.3	98.7	41.2	98.9	45.3
StillLife 2	97.8	25.4	98.3	36.1	98.5	39.1
real-world data sets						
UCSD [113]	95.8	8.9	97.0	11.2	-	-
Plenoptic 1 [72]	93.7	3.5	94.5	4.4	-	-
Plenoptic 2 [72]	91.0	6.6	96.1	8.5	-	-

Figure 56: *Comparison of local labeling accuracy (acc) and boundary recall (br) for all datasets.* The table shows percentages of correctly labeled pixels and boundary pixels, respectively, for pointwise optimal results of the three classifiers trained on the features detailed in figure 54. Disparity for IMG-D is estimated using the method described in section 5.1.2.1. Ground truth disparity is used for IMG-GT to determine the maximum possible quality of the proposed method. It is obvious that in scenes like *Buddha*, *Papillon 2*, *Horses 2* or *StillLife 2*, where the user tries to separate objects with similiar or even identical appearance, the rayspace based feature leads to a large benefit in the segmentation results.

Global Optimization. In the second set of experiments, we employ our ray space optimization framework on the results from the local classifier. The unary potentials in (61) are initialized with the log-probabilities (62) from the local class probabilities, while the spatial regularization weight g is set to

$$g = \max\{0, 1 - \left(|\nabla I|_2 - \mathcal{H}(I)\right) |\nabla \rho|_2\}, \tag{63}$$

where I denotes the respective single view image, $\mathcal{H}$ the Harris corner detector [42], and ρ the disparity field. This way, we combine the response from three different types of edge detectors. Experiments showed that the sum of the two different edge signals for the gray value image I leads to more robust boundary weights. For all of the data sets, training classifiers with light field features and optimizing over ray space leads to significantly improved results compared to single view multi-labeling (see figures 57 and 58). The effectiveness of light field segmentation is revealed in particular on data sets which have highly ambiguous texture and color between classes. In the light field *Buddha*, for example, it becomes possible to segment a column from a background wall having the same texture. In the scene *Papillon 2*, we demonstrate that it is possible to separate foreground from background leaves. Similarly, in *StillLife 2* we are able to correctly segment foreground from background raspberries. The data set *Horses 2* also represents a typical case for problems only solvable using the proposed approach. Here, we perform a labeling of identical objects in the scene with different label classes.

Optimization	Single view (SV)						Ray space (RS)						Overall improvement	
Classifier	IMG		IMG+D		IMG+GT		IMG		IMG+D		IMG+GT		RS+IMG+D	RS+IMG+GT
	acc	imp	acc	imp	acc	imp	acc	imp	acc	imp	acc	imp	vs. SV+IMG	vs. SV+IMG
synthetic data sets														
Buddha	96.3	43.4	97.5	22.1	99.1	31.2	96.3	43.8	98.8	63.8	99.1	35.5	68.2	76.0
Garden	96.4	25.7	97.9	36.6	98.1	39.2	96.9	37.0	98.0	37.5	98.2	41.4	43.4	49.2
Papillon 1	99.1	34.6	99.1	45.5	99.3	31.7	99.3	46.3	99.2	50.9	99.7	65.3	7.9	60.1
Papillon 2	92.3	22.4	98.1	46.0	99.3	29.8	93.0	24.6	98.9	68.2	99.5	44.7	84.7	92.8
Horses 1	94.7	22.3	95.7	23.7	99.2	52.6	94.8	23.0	97.7	59.4	99.2	55.6	56.2	85.6
Horses 2	96.1	28.4	96.3	21.4	99.0	31.3	96.2	29.5	98.3	64.1	99.1	36.7	56.7	76.2
StillLife 1	99.1	38.2	99.3	50	99.4	45.6	99.2	43.8	99.4	54.5	99.6	64.0	31.5	53.4
StillLife 2	98.8	47.1	98.8	31.0	99.1	41.1	99.0	55.6	98.9	38.5	99.2	45.9	10.1	33.6
real-world data sets														
UCSD	97.6	44.3	99.1	70	–	–	97.8	48.6	99.3	76.3	–	–	69.9	–
Plenoptic 1	96.4	43.5	97.0	43.6	–	–	96.8	49.5	96.9	43.6	–	–	12.9	–
Plenoptic 2	94.1	34.4	96.1	33.2	–	–	94.5	39.4	96.1	33.9	–	–	34.6	–
Average	96.8	32.2	97.9	36.3	99.2	38.7	96.9	37.1	98.7	55.9	99.4	52.0	41.2	67.1

Figure 57: *Relative improvements by global optimization.* All numbers are in percent. The quantities in the columns *acc* indicate the percentage of correctly labeled pixels. The columns *imp* denote the relative improvement of the optimized compared to the respective raw result from the local classifier in figure 56. To be more specific, if acc_p is the previous and acc_n the new accuracy, then the column *imp* contains the number $(acc_n - acc_p)/(1 - acc_p)$, i.e. the percentage of previously erroneous pixels which were corrected by optimization. Optimal smoothing parameters λ, μ were determined using a grid search over the parameter space. We also compare our ray space optimization framework (RS) to single view optimization (SV), which can be achieved by setting the amount of EPI regularization μ to zero. Note that for every single classifier and data set, RS optimization achieves a better result than SV optimization. The last two columns indicate the relative accuracy of ray space optimization and the indicated ray space classifier versus single view optimization and single view features, computed the same way as the *imp* columns. In particular, they demonstrate the overall improvement which is realized with the proposed method.

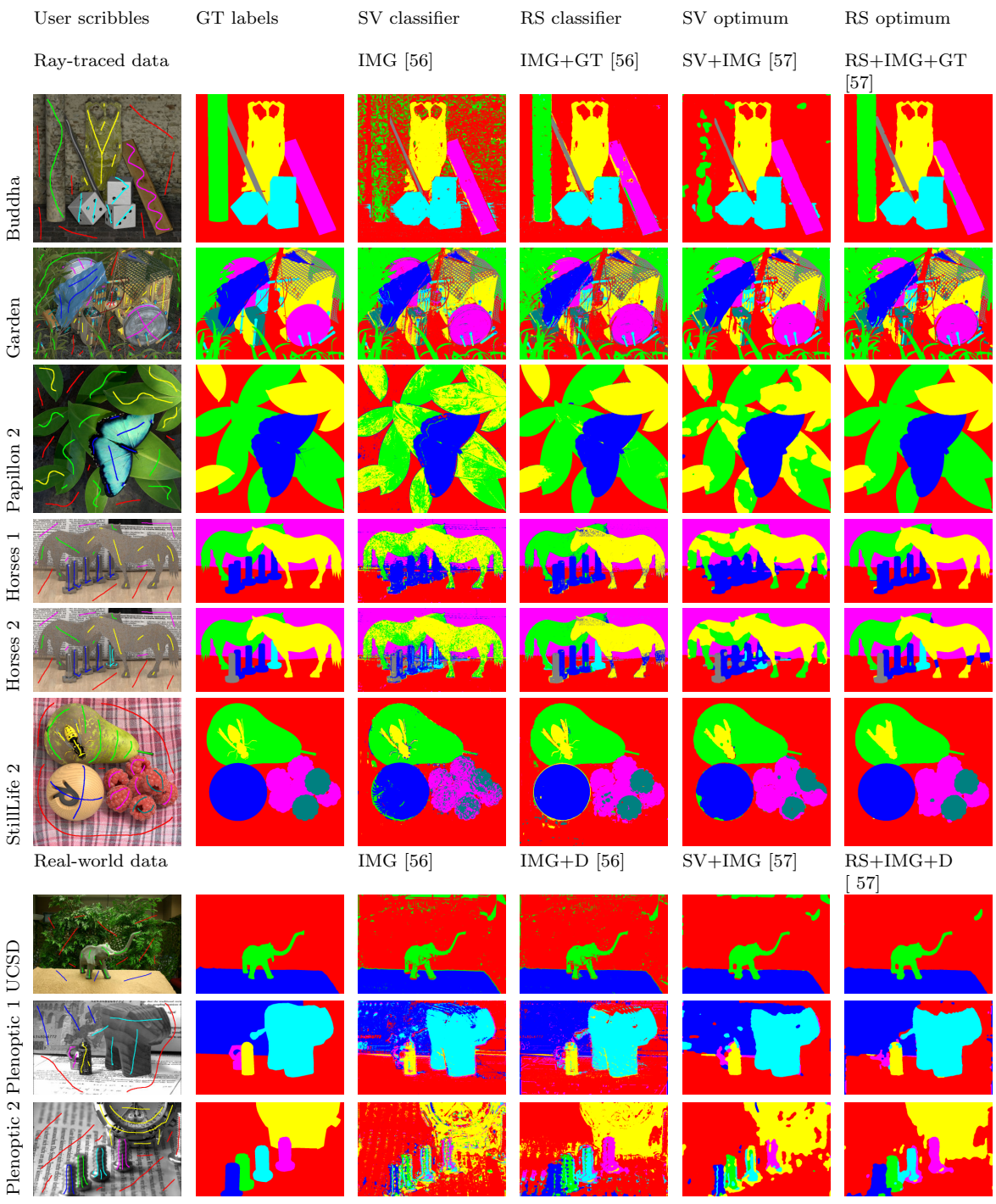


Figure 58: Segmentation results for a number of ray-traced and real-world light fields. *GT* stands for ground truth, *SV* for single view and *RS* for ray space. The numbers in squared brackets refer to the corresponding figures. The first two columns on the left show the center view with user scribbles and ground truth labels. The two middle columns compare classifier results for the local single view and light field features denoted on top. Since the focus of this paper is segmentation rather than depth reconstruction, here we show results for ground truth depth where available to compare to the optimal possible results from light field data. Finally, the two rightmost columns compare the final results after single view and ray space optimization, respectively. In particular for difficult cases, the proposed method is significantly superior.

7 Conclusion

In this work novel methods for the analysis of 4D light fields was presented. We showed that a specific parametrization, the so-called *Lumigraph*, is well suited for an orientation based analysis. The *Lumigraph* can be described as a dense collection of pinhole views captured on a planar regular grid of camera positions. This causes a linear mapping of 3D points onto lines in the so-called epipolar plane images. We discussed different devices and techniques to capture light fields as well as the effort necessary to represent the resulting raw data as a *Lumigraph* or as epipolar plane images respectively.

In chapter 3, we saw that raw data from a *Focused Plenoptic Camera* does not provide an immediate access to epipolar plane images. A method was proposed to render all possible *all-in-focus* views from the raw data, which is the desired *Lumigraph* parametrization. To avoid a pixel-wise depth estimation within the micro-lens images, we minimized the gradients at neighboring micro-image patches to render views without plenoptic artifacts.

In chapter 4 we discussed the acquisition techniques relevant for this work in detail. To generate light fields of best quality we used a high-end consumer camera in combination with a precise xy-stepper motor. This so-called *gantry* is ideal to capture very dense light fields down to baselines of $1mm$. The disadvantage is that only static scenes can be captured. Together with light fields generated using computer graphics, providing full ground truth data, a benchmark database containing over a dozen simulated and real world light fields was published during this work (`www.lightfield-analysis.net`).

In chapter 5, we proposed fast and robust methods, based on an orientation analysis of epipolar plane images, to compute depth range data. The single orientation analysis introduced makes use of the structure tensor to analyze epipolar plane images. The structure tensor analyzes first order derivatives to locally estimate structure and orientation in an image. If the appearance of a 3D point does not depend on the view point, it is mapped onto a line in an epipolar plane image. However, this approach is restricted to the *Lambertian* assumption. If reflections or transparencies are present, overlayed line patterns arise in the epipolar space the structure tensor cannot handle. An extension to multi-orientation patterns, making use of a higher order structure tensor, was proposed. We showed that this multi-orientation analysis leads to much more robust depth estimation where reflections or semi-transparent materials are present.

In Chapter 6, we discussed two applications of the orientation based depth estimation. We proposed an angular and spatial super-resolution algorithm based on an energy minimization framework as well as a framework for optimal label assignment on ray space for object segmentation. Both methods show the potential of light fields for image processing and computer vision tasks. The super-resolution framework can be seen as a proof for the high quality of the depth maps computed using the orientation analysis, since this method needs disparity estimations of sub-pixel accuracy to work properly. In the case of object segmentation the benefits are quite obvious. Due to the inherently

available depth information within light fields, object detection is getting much more robust when applied on light fields. We used a standard "random-forest" classifier in this work to predict object labels. Compared to predictions on 2D images we were able to distinguish objects of different classes but similar appearance.

8 Outlook

Possible extensions of the work presented could be the following:

The proposed orientation analysis in this work is still separated in single orientation and double orientation models and in particular the double orientation model needs the outcome of the single orientation to interpret the resulting channels. However, this needs to be unified in a more problem specific manner. The single orientation model is already included in the second order structure tensor and a more advanced evaluation of all tensor channels at once would lead to more robust results.

The orientation analysis as described in this work handles 4D light fields as separated horizontal and vertical 3D light fields merging the outcome in a final step by pixel-wise choosing the disparity more reliable as the final result. From a computational efficiency point of view this makes sense, since an evaluation of the 4D data as a whole is quite expensive, but on the other hand the method described in this work does not make use of all available information.

Next steps planned for future research are an evaluation of the depth estimation accuracy on real scenes and extensions of the orientation analysis to light fields varying over time. Further developments in both are planned, investigating light fields of dynamic scenes as well as light fields of static scenes under varying illumination conditions.

List of Publications

Generating EPI Representations of 4D Light Fields with a Single Lens Focused Plenoptic Camera
S. Wanner, J. Fehr, B. Jähne
7th International Symposium on Visual Computing, Las Vegas, Sept. 24-26, 2011

Globally Consistent Depth Labeling of 4D Light Fields
S. Wanner, B. Goldlücke
CVPR'12, Providence, Rhode Island, June 16-21, 2012

Spatial and Angular Variational Super-resolution of 4D Light Fields
S. Wanner, B. Goldlücke
ECCV'12, Florence, Italy, October 7-13, 2012

Variational Light Field Analysis for Disparity Estimation and Super-Resolution
S. Wanner, B. Goldlücke
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013

Globally Consistent Multi-Label Assignment on the Ray Space of 4D Light Fields
S. Wanner, C. Straehle, B. Goldlücke
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013

The Variational Structure of Disparity and Regularization of 4D Light Fields
B. Goldlücke, S. Wanner
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013

Datasets and Benchmarks for Densely Sampled 4D Light Fields
S. Wanner, S. Meister, B. Goldlücke
Vision, Modelling and Visualization (VMV), 2013

Reconstructing Reflective and Transparent Surfaces from Epipolar Plane Images
S. Wanner and B. Goldlücke
German Conference on Pattern Recognition (GCPR), 2013

Bibliography

[1] Aach, T., Mota, C., Stuke, I., Muehlich, M., and Barth, E. (2006). Analysis of superimposed oriented patterns. *IEEE Transactions on Image Processing*, **15**(12), 3690–3700.

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

[3] Adelson, E. H. and Wang, J. Y. (1992). Single lens stereo with a plenoptic camera. *IEEE transactions on pattern analysis and machine intelligence*, **14**(2), 99–106.

[4] Alldrin, N., Zickler, T., and Kriegman, D. (2008). Photometric Stereo With Non-Parametric and Spatially-Varying Reflectance. In *Proc. International Conference on Computer Vision and Pattern Recognition*.

[5] Ashdown, I. (1993). Near-field photometry: A new approach. *JOURNAL-ILLUMINATING ENGINEERING SOCIETY*, **22**, 163–163.

[6] Baker, H. H. (1989). Building surfaces of evolution: The weaving wall. *International Journal of Computer Vision*, **3**(1), 51–71.

[7] Baker, H. H. and Bolles, R. C. (1989). Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, **3**(1), 33–49.

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

[9] Beck, A. and Teboulle, M. (2009). Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences*, **2**, 183–202.

[10] Benton, S. A. (1983). Survey of holographic stereograms. In *26th Annual Technical Symposium*, pages 15–19. International Society for Optics and Photonics.

[11] Berent, J. and Dragotti, P. (2006). Segmentation of epipolar-plane image volumes with occlusion and disocclusion competition. In *IEEE 8th Workshop on Multimedia Signal Processing*, pages 182–185.

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

[13] Bishop, T. and Favaro, P. (2011). Full-resolution depth map estimation from an aliased plenoptic light field. *Computer Vision–ACCV 2010*, pages 186–200.

[14] Bishop, T. and Favaro, P. (2012). The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(5), 972–986.

[15] Bishop, T. E. and Favaro, P. (2009). Plenoptic depth estimation from multiple aliased views. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1622–1629. IEEE.

[16] Bolles, R., Baker, H., and Marimont, D. (1987). Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, **1**(1), 7–55.

[17] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

[18] Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

[19] Chai, J.-X., Tong, X., Chany, S.-C., and Shum, H.-Y. (2000). Plenoptic sampling. *Proc. SIGGRAPH*, pages 307–318.

[20] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, **20**(1-2), 89–97.

[21] Chambolle, A. and Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.*, **40**(1), 120–145.

[22] Chambolle, A., Cremers, D., and Pock, T. (2008). A Convex Approach for Computing Minimal Partitions. Technical Report TR-2008-05, Dept. of Computer Science, University of Bonn.

[23] Chang, C.-F., Bishop, G., and Lastra, A. (1999). Ldi tree: A hierarchical representation for image-based rendering. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 291–298. ACM Press/Addison-Wesley Publishing Co.

[24] Chen, S. E. and Williams, L. (1993). View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288. ACM.

[25] Chunev, G., Lumsdaine, A., and Georgiev, T. (2011). Plenoptic rendering with interactive performance using gpus. In *SPIE Electronic Imaging*.

[26] Criminisi, A. (2011). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, **7**(2-3), 81–227.

[27] Criminisi, A., Kang, S., Swaminathan, R., Szeliski, R., and Anandan, P. (2005). Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer vision and image understanding*, **97**(1), 51–85.

[28] Dansereau, D. G., Pizarro, O., and Williams, S. B. (2013). Decoding, calibration and rectification for lenselet-based plenoptic cameras.

[29] Davis, J., Yang, R., and Wang, L. (2005). BRDF Invariant Stereo using Light Transport Constancy. In *Proc. International Conference on Computer Vision*.

[30] DeGroot, M. H. (2005). *Optimal statistical decisions*, volume 82. Wiley-Interscience.

[31] Esedoglu, S. and March, R. (2003). Segmentation with Depth but Without Detecting Junctions. *Journal of Mathematical Imaging and Vision*, **18**(1), 7–15.

[32] Federer, H. (1969). *Geometric measure theory*. Springer-Verlag New York Inc., New York.

[33] Georgiev, T. and Lumsdaine, A. (2010a). Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, **19**(2), 021106–021106.

[34] Georgiev, T. and Lumsdaine, A. (2010b). Reducing plenoptic camera artifacts. In *Computer Graphics Forum*. Wiley Online Library.

[35] Geys, I., Koninckx, T. P., and Gool, L. V. (2004). Fast interpolated cameras by combining a GPU based plane sweep with a max-flow regularisation algorithm. In *3DPVT*, pages 534–541.

[36] Gokturk, S. B., Yalcin, H., and Bamji, C. (2004). A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 35–35. IEEE.

[37] Goldluecke, B. (2013). *cocolib* - a library for continuous convex optimization. http://cocolib.net.

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

[39] Goldluecke, B. and Wanner, S. (2013). The Variational Structure of Disparity and Regularization of 4D Light Fields. In *Proc. International Conference on Computer Vision and Pattern Recognition*.

[40] Goldman, D., Curless, B., Hertzmann, A., and Seitz, S. (2010). Shape and Spatially-Varying BRDFs from Photometric Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(6), 1060–1071.

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

[42] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK.

[43] Hirschmuller, H. and Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[44] Ives, F. (1903). Patent us 725,567.

[45] Jaehne, B. (2005). *Digitale Bildverarbeitung*. Springer DE.

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

[47] Joshi, N., Matusik, W., and Avidan, S. (2006). Natural video matting using camera arrays. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 779–786. ACM.

[48] Kang, S. B. (1997). *A survey of image-based rendering techniques*. Digital, Cambridge Research Laboratory.

[49] Koethe, U. (2012). The vigra computer vision library version 1.9.0 http://hci.iwr.uni-heidelberg.de/vigra/.

[50] Kowdle, A., Sinha, S., and Szeliski, R. (2012). Multiple View Object Cosegmentation using Appearance and Stereo Cues. In *Proc. European Conference on Computer Vision*.

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

[52] Lazaros, N., Sirakoulis, G. C., and Gasteratos, A. (2008). Review of stereo vision algorithms: from software to hardware. *International Journal of Optomechatronics*, **2**(4), 435–462.

[53] Lellmann, J., Becker, F., and Schnörr, C. (2009). Convex Optimization for Multi-Class Image Labeling with a Novel Family of Total Variation Based Regularizers. In *IEEE International Conference on Computer Vision (ICCV)*.

[54] Lengyel, J. (1998). The convergence of graphics and vision. *Computer*, **31**(7), 46–53.

[55] Levin, A. and Weiss, Y. (2007). User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[56] Levin, A., Zomet, A., and Weiss, Y. (2004). Separating Reflections from a Single Image Using Local Features. In *Proc. International Conference on Computer Vision and Pattern Recognition*.

[57] Levoy, M. (2006). Light fields and computational imaging. *Computer*, **39**(8), 46–55.

[58] Levoy, M. and Hanrahan, P. (1996a). Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM.

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

[60] Levoy, M., Ng, R., Adams, A., Footer, M., and Horowitz, M. (2006). Light field microscopy. *ACM Transactions on Graphics (TOG)*, **25**(3), 924–934.

[61] Lippmann, G. (1908). Epreuves reversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, **7**(1), 821–825.

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

[63] Lumsdaine, A. and Georgiev, T. (2009a). The focused plenoptic camera. In *In Proc. IEEE ICCP*, pages 1–8.

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

[65] Mark, W. R., McMillan, L., and Bishop, G. (1997). Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 7–ff. ACM.

[66] Marwah, K., Wetzstein, G., Bando, Y., and Raskar, R. (2013). Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections. *ACM Trans. Graph. (Proc. SIGGRAPH)*, **32**(4), 1–11.

[67] Matoušek, M., Werner, T., and Hlavác, V. (2001). Accurate correspondences from epipolar plane images. In *Proc. Computer Vision Winter Workshop*, pages 181–189.

[68] Meister, S. (2014). *On Creating Reference Data for Performance Analysis in Image Processing*. Ph.D. thesis, University of Heidelberg.

[69] Ng, R. (2006). *Digital Light Field Photography*. Ph.D. thesis, Stanford University. Note: thesis led to commercial light field camera, see also `www.lytro.com`.

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

[71] Nickolls, J., Buck, I., Garland, M., and Skadron, K. (2008). Scalable parallel programming with cuda. *Queue*, **6**(2), 40–53.

[72] Perwass, C. and Wietzke, L. (2010). The next generation of photography, www.raytrix.de.

[73] Perwass, C. and Wietzke, L. (2012). Single lens 3d-camera with extended depth-of-field. In *SPIE Electronic Imaging 2012*, pages 22–26.

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

[75] Ponce, J., Forsyth, D., Willow, E.-p., Antipolis-Méditerranée, S., d'activité RAweb, R., Inria, L., and Alumni, I. (2011). Computer vision: a modern approach. *Computer*, **16**, 11.

[76] Protter, M. and Elad, M. (2009). Super-Resolution With Probabilistic Motion Estimation. *IEEE Transactions on Image Processing*, **18**(8), 1899–1904.

[77] Roosendaal, T. (1998). Blender, http://www.blender.org.

[78] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, **60**(1), 259–268.

[79] Ruiters, R. and Klein, R. (2009). Heightfield and spatially varying BRDF Reconstruction for Materials with Interreflections. *Computer Graphics Forum (Proc. Eurographics)*, **28**(2), 513–522.

[80] Sard, A. (1942). The measure of the critical values of differentiable maps. *Bull. Amer. Math. Soc*, **48**(12), 883–890.

[81] Scharr, H. (2000). Optimale operatoren in der digitalen bildverarbeitung.

[82] Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[83] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, **47**, 7–42.

[84] Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–195. IEEE.

[85] Seitz, S. M. and Dyer, C. R. (1997). View morphing: Uniquely predicting scene appearance from basis images. In *Proc. Image Understanding Workshop*, pages 881–887.

[86] Shade, J., Gortler, S., He, L.-w., and Szeliski, R. (1998). Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242. ACM.

[87] Shan, J. and Toth, C. K. (2008). *Topographic laser ranging and scanning: principles and processing*. CRC Press.

[88] Shum, H. and Kang, S. B. (2000). Review of image-based rendering techniques. In *VCIP*, pages 2–13. Citeseer.

[89] Shum, H.-Y. and He, L.-W. (1999). Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 299–306. ACM Press/Addison-Wesley Publishing Co.

[90] Sinha, S., Kopf, J., Goesele, M., Scharstein, D., and Szeliski, R. (2012). Image-Based Rendering for Scenes with Reflections. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, **31**(4), 100:1–100:10.

[91] Smith, O. (1961). Eigenvalues of a symmetric 3x3 matrix. *Communications of the ACM*, **4**(4), 168.

[92] Stein, A., Hoiem, D., and Hebert, M. (2007). Learning to Find Object Boundaries Using Motion Cues. In *Proc. International Conference on Computer Vision*.

[93] Strekalovskiy, E. and Cremers, D. (2011). Generalized Ordering Constraints for Multilabel Optimization. In *Proc. International Conference on Computer Vision*.

[94] Strekalovskiy, E., Goldluecke, B., and Cremers, D. (2011). Tight Convex Relaxations for Vector-Valued Labeling Problems. In *Proc. International Conference on Computer Vision*.

[95] The HDF Group, http://www.hdfgroup.org/HDF5 (2000-2010). Hierarchical data format version 5.

[96] Tsin, Y., Kang, S., and Szeliski, R. (2006). Stereo Matching with Linear Superposition of Layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(2), 290–301.

[97] Vaish, V., Wilburn, B., Joshi, N., and Levoy, M. (2004). Using plane+ parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–2. IEEE.

[98] van Rossum, G. and Drake (eds), F. (2001). Python reference manual, pythonlabs, virginia, usa, 2001 available at http://www.python.org.

[99] Wang, Z. and Bovik, A. (2009). Mean Squared Error: Love it or Leave it? *IEEE Signal Processing Magazine*, **26**(1), 98–117.

[100] Wanner, S. and Goldluecke, B. (2012a). Globally consistent depth labeling of 4D light fields. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 41–48.

[101] Wanner, S. and Goldluecke, B. (2012b). Spatial and angular variational superresolution of 4D light fields. In *Proc. European Conference on Computer Vision*.

[102] Wanner, S. and Goldluecke, B. (2013a). Reconstructing reflective and transparent surfaces from epipolar plane images. In *Pattern Recognition*, pages 1–10. Springer.

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

[104] Wanner, S., Fehr, J., and Jähne, B. (2011). Generating EPI representations of 4D Light fields with a single lens focused plenoptic camera. *Advances in Visual Computing*, pages 90–101.

[105] Wanner, S., Meister, S., and Goldluecke, B. (2013a). Datasets and benchmarks for densely sampled 4d light fields. In *Vision, Modeling & Visualization*, pages 225–226. The Eurographics Association.

[106] Wanner, S., Straehle, C., and Goldluecke, B. (2013b). Globally Consistent Multi-Label Assignment on the Ray Space of 4D Light Fields. In *Proc. International Conference on Computer Vision and Pattern Recognition*.

[107] Wetzstein, G., Ihrke, I., Lanman, D., and Heidrich, W. (2011). Computational plenoptic imaging. In *Computer Graphics Forum*, volume 30, pages 2397–2426. Wiley Online Library.

[108] Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. (2005a). High performance imaging using large camera arrays. *ACM Trans. Graph.*, **24**, 765–776.

[109] Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. (2005b). High performance imaging using large camera arrays. *ACM Transactions on Graphics*, **24**, 765–776.

[110] Zach, C., Gallup, D., Frahm, J.-M., and Niethammer, M. (2008). Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In *Vision, Modeling and Visualization Workshop VMV 2008*.

[111] Zach, C., Niethammer, M., and Frahm, J.-M. (2009). Continuous Maximal Flows and Wulff Shapes: Application to MRFs. In *Proc. International Conference on Computer Vision and Pattern Recognition*.

[112] Zickler, T., Belhumeur, P., and Kriegman, D. (2002). Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction. *International Journal of Computer Vision*, **49**(2–3), 215–227.

[113] Zwicker, M., Matusik, W., Durand, F., Pfister, H., and Forlines, C. (2006). Antialiasing for automultiscopic 3D displays. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, page 107. ACM.