Variational Light Field Analysis for Disparity Estimation and Super-Resolution

Sven Wanner, Member, IEEE, and Bastian Goldluecke, Member, IEEE

Abstract—We develop a continuous framework for the analysis of 4D light fields, and describe novel variational methods for disparity reconstruction as well as spatial and angular super-resolution. Disparity maps are estimated locally using epipolar plane image analysis without the need for expensive matching cost minimization. The method works fast and with inherent subpixel accuracy, since no discretization of the disparity space is necessary. In a variational framework, we employ the disparity maps to generate super-resolved novel views of a scene, which corresponds to increasing the sampling rate of the 4D light field in spatial as well as angular direction. In contrast to previous work, we formulate the problem of view synthesis as a continuous inverse problem, which allows us to correctly take into account foreshortening effects caused by scene geometry transformations. All optimization problems are solved with state-of-the-art convex relaxation techniques. We test our algorithms on a number of real-world examples as well as our new benchmark dataset for lightfields, and compare results to a multiview stereo method. The proposed method is both faster as well as more accurate. Data sets and source code are provided online for additional evaluation.

Index Terms—Light fields, epipolar plane images, 3D reconstruction, super-resolution, view interpolation, variational methods

1 INTRODUCTION

The 4D light field has been established as a promising paradigm to describe the visual appearance of a scene. Compared to a traditional 2D image, it offers information about not only the accumulated intensity at each image point, but separate intensity values for each ray direction. Thus, the light field implicitly captures 3D scene geometry and reflectance properties.

The additional information inherent in a light field allows a wide range of applications. Popular in computer graphics, for example, is light field rendering, where the scene is displayed from a virtual viewpoint [26], [21]. The light field data also allows to add effects like synthetic aperture, i.e. virtual refocusing of the camera, stereoscopic display, and automatic glare reduction as well as object insertion and removal [18], [16], [10]. As we will exploit in this work, the continuous disparity space also admits non-traditional approaches to the multiview stereo problem that do not rely on feature matching [7], [35].

However, in practice, it used to be difficult to achieve a dense enough sampling of the full light field. Expensive custom-made hardware was designed to be able to acquire several views of a scene. Straightforward but hardware-intensive are camera arrays [31]. Somewhat more practical and less expensive is a gantry construction consisting of a single moving camera [19], [30], which is restricted to static scenes. Recently, however, the first commercial plenoptic cameras have become available on the market. Using an array of microlenses, a single one of these cameras essentially captures an full array of views simultaneously. This makes such cameras very attractive for a number of industrial applications, in particular depth estimation and surface inspection, and they can also acquire video streams of dynamic scenes [11], [22], [23].

However, plenoptic cameras usually have to deal with a trade-off between spatial and angular resolution. Since the total sensor resolution is limited, one can either opt for a dense sampling in the spatial (image) domain with sparse sampling in the angular (view point) domain [23], or vice versa [22], [6], [11]. Increasing angular resolution is therefore a paramount goal if one wants to make efficient use of plenoptic cameras. It is equivalent to the synthesis of novel views from new viewpoints, which has also been a prominent research topic in the computer graphics community [21], [19].

Beyond the need for super-resolution, there is a growing demand for efficient and robust algorithms which reconstruct information directly from light fields. However, while there has been a lot of work on for example stereo and optical flow algorithms for traditional image pairs, there is a lack of similar modern methods which are specifically tailored to the rich structure inherent in a light field. Furthermore, much of the existing analysis is local in nature, and does not enforce global consistency of results.
Contributions. In this paper, we first address the problems of disparity estimation in light fields, where we introduce a novel local data term tailored to the continuous structure of light fields. The proposed method can locally obtain robust results very fast, without having to quantize the disparity space into discrete disparity values. The local results can further be integrated into globally consistent depth maps using state-of-the-art labeling schemes based on convex relaxation methods [24], [29].

In this way, we obtain an accurate geometry estimate with subpixel precision matching, which we can leverage to simultaneously address the problems of spatial and angular super-resolution. Following state-of-the-art spatial super-resolution research in computer vision [25], [14], we formulate a variational inverse problem whose solution is the synthesized super-resolved novel view. As we work in a continuous setting, we can for the first time correctly take into account foreshortening effects caused by the scene geometry.

The present work unifies and extends our previous conference publications [35] and [36]. As additional contributions, we focus in depth on interactive techniques applicable in practice, and analyze parameter choices, scope and limitations of our method in detailed experiments. To facilitate speed, we provide a new fast method to combine local epipolar plane image disparity estimates into a global disparity map. Furthermore, we give an extensive comparison to a multiview stereo method on our new light field benchmark dataset.

Advantages and limitations of the method. Our method exploits the fact that in a light field with densely sampled view points, derivatives of the intensity can be computed with respect to the view point location. The most striking difference to standard disparity estimation is that at no point, we actually compute stereo correspondences in the usual sense - we never try to match pixels at different locations. As a consequence, the run-time of our method is completely independent of the desired disparity resolution, and we beat other methods we have compared against in terms of speed, while maintaining on average similar or better accuracy. A typical failure case (just as for stereo methods) are regions with strong specular highlights or devoid of any texture. By design, there are no problems with repetitive structures due to the local nature of the slope estimation.

However, as we will explore in experiments, the sampling of view points must be sufficiently dense such that disparities between neighbouring views are less than around two pixels to achieve reasonable accuracy. Furthermore, for optimal results it is also recommended that the view points form a two-dimensional rectangular grid, although in principle a one-dimensional line of view points is sufficient.

In particular, this is not the case for established datasets like the Middlebury stereo benchmark, which has far too large disparities and therefore is beyond the design scope of our algorithm. For this reason, we evaluate on our own set of synthetic benchmarks, which more closely resembles the data from a plenoptic camera and is more representative for what we consider the advantages of light field data compared to traditional multi-view input. It is available online together with complete source code to reproduce all experiments [33], [15].

2 RELATED WORK

The concept of light fields originated mainly in computer graphics, where image based rendering [27] is a common technique to render new views from a set of images of a scene. Adelson and Bergen [1] as well as McMillan and Bishop [21] treated view interpolation as a reconstruction of the plenoptic function. This function is defined on a seven-dimensional space and describes the entire information about light emitted by a scene, storing an intensity value for every 3D point, direction, wavelength and time. A dimensionality reduction of the plenoptic function to 4D, the so called Lumigraph, was introduced by Gortler et al. [15], and Levoy and Hanrahan [19]. In their parametrization, each ray is determined by its intersections with two planes.

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 (EPIs), which can be viewed as 2D slices of constant angular and spatial coordinate through the Lumigraph. Due to a dense sampling in angular direction, corresponding pixels are projected onto lines in EPIs, which can be more robustly detected than point correspondences.

Geometry estimation in a 4D light field. One of the first approaches using EPIs to analyze scene geometry was published by Bolles et al. [7]. They detect edges, peaks and troughs with a subsequent line fitting in the EPI to reconstruct 3D structure. Another approach is presented by Criminisi et al. [10], 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 Matoušek et al. [20], 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. [4] deal with the simultaneous segmentation of EPI-tubes by a region competition method using active contours, imposing geometric properties to enforce correct occlusion ordering.

As novelty to previous work, we suggest to employ the structure tensor of an EPI to obtain a fast and robust local disparity estimation. Furthermore, we enforce globally consistent visibility across views by restricting the spatial layout of the labeled regions. Compared to methods which extract EPI-tubes sequentially [7], [10], 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 in [4], they use a level set approach, which makes it expensive and cumbersome to deal with a large number of regions.

**Spatial and angular super-resolution.** Super-resolving the four dimensions in a light field amounts to generating high-resolution novel views of the scene which have not originally been captured by the cameras. A collection of images of a scene can be interpreted as a sparse sampling of the plenoptic function. Consequently, image-based rendering approaches [27] treat the creation of novel views as a resampling problem, circumventing the need for any explicit geometry reconstruction [21], [19], [17]. However, this approach ignores occlusion effects, and therefore is only really suitable for synthesis of views reasonably close to the original ones.

Overall, it quickly became clear that one faces a trade-off, and interpolation of novel views in sufficient enough quality requires either an unreasonably dense sampling or knowledge about the scene [8]. A different line of approaches to light field rendering therefore tries to infer at least some geometric knowledge about the scene. They usually rely on image registration, for example via robust feature detecting and tracking [28], or view-dependent depth map estimation based on color consistency [12].

The creation of super-resolved images requires subpixel-accurate registration of the input images. Approaches which are based on pure 2D image registration [25] are unsuitable for the generation of novel views, since a reference image for computing the motion is not available yet. Super-resolved depth maps and images are inferred in [6] with a discrete super-resolution model tailored to a particular plenoptic camera. A full geometric model with a super-resolved texture map is estimated in [14] for scenes captured with a surround camera setup. Our approach is mathematically closely related to the latter, since [14] is also based on continuous geometry which leads to correct point-wise weighting of the energy gradient contributions. However, we do not perform expensive computation of a global model and texture atlas, but instead compute the target view directly.

### 3 4D LIGHT FIELD STRUCTURE

Several ways to represent light fields have been proposed. In this paper, we adopt the light field parametrization from early works in motion analysis [7]. 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, figure 2. 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 is a map

\[ L : Ω \times Π \to \mathbb{R}, \quad (x, y, s, t) \mapsto L(x, y, s, t). \]  

(1)

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

For the problem of estimating 3D structure, 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 $\Sigma_{y^*,t^*}$. The resulting map is called an epipolar plane image (EPI),

$$S_{y^*,t^*} : \Sigma_{y^*,t^*} \rightarrow \mathbb{R},$$

$$(x, s) \mapsto S_{y^*,t^*}(x, s) := L(x, y^*, s, t^*).$$

(2)

Let us consider the geometry of this map, figure 2. A point $P = (X, Y, Z)$ within the epipolar plane corresponding to the slice projects to a point in $\Omega$ depending on the chosen camera center in $\Pi$. If we vary $s$, the coordinate $x$ changes according to [7]

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

(3)

where $f$ is the distance between the parallel planes. Note that to obtain this formula from figure 12, $\Delta x$ has to be corrected by the translation $\Delta s$ to account for the different local coordinate systems of the views.

Interestingly, a point in 3D space is thus projected onto a line in $\Sigma_{y^*,t^*}$, where the slope of the line is related to its depth. This means that the intensity 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 [7], [10]. In the next section, we describe and evaluate our novel approach how to obtain consistent slope estimates.

### 4 Disparity Estimation

The basic idea of our approach is as follows. We first compute local slope estimates on epipolar plane images for the two different slice directions using the structure tensor. This gives two local disparity estimates for each pixel in each view. These can be merged into a single disparity map in two 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 even slightly more accurate.

Obviously, 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.

#### 4.1 Disparities on epipolar plane images

**a) Local disparity estimation on an EPI.** We first consider how we can estimate the local direction of a line at a point $(x, s)$ in an epipolar plane image $S_{y^*,t^*}$, 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 $r_{y^*,t^*}(x, s) \in [0, 1]$, which gives a measure of how reliable the local disparity estimate is. Both local estimates will 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 $J$ of the epipolar plane image $S = S_{y^*,t^*}$,

$$J = \begin{bmatrix}
G_x * (S_x S_x) & G_x * (S_x S_y) \\
G_y * (S_x S_y) & G_y * (S_y S_y)
\end{bmatrix} = \begin{bmatrix}
J_{xx} & J_{xy} \\
J_{xy} & J_{yy}
\end{bmatrix}.$$  

Here, $G_x$ represents a Gaussian smoothing operator at an outer scale $\sigma$ and $S_x, S_y$ denote the gradient
components of $S$ calculated on an inner scale $\rho$.

The direction of the local level lines can then be computed via [5]

$$n_{y^*,t^*} = \frac{\Delta x}{\Delta s} = \begin{bmatrix} \sin(\varphi) \\ \cos(\varphi) \end{bmatrix} \quad \text{with} \quad \varphi = \frac{1}{2} \arctan \left( \frac{J_{yy} - J_{xx}}{2J_{xy}} \right),$$

from which we derive the local depth estimate via equation (3) as

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

Frequently, a more convenient unit is the disparity $d_{y^*,t^*} = \frac{1}{Z} = \frac{\Delta x}{\Delta s} = \tan \phi$, 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 the paper. As the natural reliability measure we use the coherence of the structure tensor [5],

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

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 direction, respectively, one can now proceed to directly compute disparity maps in a global optimization framework, which is explained in section 4.2. However, it is possible to first enforce global visibility constraints separately on each of the EPIs, which we explain in the next section.

b) Consistent disparity labeling on an EPI. 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 3.

In our conference paper [35], we have shown that by using a variational labeling framework based on ordering constraints [29], 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 4. 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 paper, and instead evaluate only the interactive technique, using results from the local structure tensor computation directly.

4.2 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 $u : \Omega \rightarrow \mathbb{R}$ for each view $(s^*, t^*)$. This is the objective of the following section.
Mean error depending on disparity for Buddha 9x9

We see that it is still quite noisy, furthermore, edges are not yet localized very well, since computing the structure tensor entails an initial smoothing of the input data. For this reason, a fast method to obtain quality disparity maps is to employ a TV-smoothing scheme, where we encourage discontinuities of the local smoothness with a measure of the coherence of the structure tensor. As we establish guidelines to select optimal inner and outer scale parameters of the structure tensor. As we establish guidelines to select optimal inner and outer scale parameters of the structure tensor.

\[ E(u) = \int_\Omega g |Du| + \frac{1}{2\lambda} |u - f| \, d(x, y), \tag{8} \]

where we minimize a functional of the form

\[ E(u) = \int_\Omega g |Du| + \rho(u, x, y) \, d(x, y). \tag{9} \]

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

\[ \rho(u, x, y) := \min(r_{y^*,t^*}(x, s^*) \left| u - d_{y^*,t^*}(x, s^*) \right|, \]

\[ r_{x^*,s^*}(y, t^*) \left| u - d_{x^*,s^*}(y, t^*) \right|). \tag{10} \]

We compute globally optimal solutions to the functional \((9)\) using the technique of functional lifting described in [24], which is also implemented in cocolib [13]. 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.

4.3 Performance analysis for interactive labeling

In this section, we perform detailed experiments with the local disparity estimation algorithm in section 4.1(a) to analyze both quality as well as 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
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.

Fig. 7: 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 from section 4.1 and (c) the result after TV-L¹ denoising according to section 4.2. 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.

a quality measurement, we use the percentage of depth values below a relative error \( \epsilon = \frac{|u(x, y) - r(x, y)|}{r(x, y)} \), 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 5, 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.

**Minimum sampling density.** In a second step, we investigate what sampling density we need for an optimal performance of the algorithm. To achieve this, we tested all datasets over the full angular resolution range with the optimal parameter selection found in figure 5. The results are illustrated in figure 6, 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 \( \pm 1 \) pixels, which makes sense from a sampling perspective since the derivatives in the structure tensor are computed on a \( 3 \times 3 \) 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.** A second interesting fact is observable on the right hand side of figure 6, where we test the robustness against noise. Within a disparity range of \( \pm 1 \), 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-L¹ 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.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average run time(s)</th>
<th>Accuracy SSIM</th>
<th>MSE</th>
<th>Disparity error ≥ 1.0</th>
<th>≥ 0.5</th>
<th>≥ 0.1</th>
<th>Depth error ≥ 1.0</th>
<th>≥ 0.5</th>
<th>≥ 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed EPI algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure tensor only</td>
<td>6s</td>
<td>0.87</td>
<td>0.029</td>
<td>0.70</td>
<td>1.95</td>
<td>16.60</td>
<td>0.32</td>
<td>3.66</td>
<td>55.69</td>
</tr>
<tr>
<td>TV-L¹ smoothing</td>
<td>1.6s</td>
<td>0.94</td>
<td>0.018</td>
<td>0.43</td>
<td>1.28</td>
<td>9.85</td>
<td>0.18</td>
<td>2.45</td>
<td>47.83</td>
</tr>
<tr>
<td>Global optimization</td>
<td>5.5h</td>
<td>0.93</td>
<td>0.019</td>
<td>0.54</td>
<td>1.4</td>
<td>8.82</td>
<td>0.23</td>
<td>2.44</td>
<td>46.71</td>
</tr>
<tr>
<td>Multiview stereo (all 9 x 9 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datatment only</td>
<td>101s</td>
<td>0.77</td>
<td>0.037</td>
<td>0.85</td>
<td>2.89</td>
<td>14.61</td>
<td>0.45</td>
<td>6.93</td>
<td>46.26</td>
</tr>
<tr>
<td>TV-L¹ smoothing</td>
<td>1.6s</td>
<td>0.91</td>
<td>0.023</td>
<td>0.54</td>
<td>1.71</td>
<td>9.13</td>
<td>0.19</td>
<td>3.68</td>
<td>41.65</td>
</tr>
<tr>
<td>Global optimization</td>
<td>5.5h</td>
<td>0.92</td>
<td>0.027</td>
<td>0.71</td>
<td>1.9</td>
<td>7.51</td>
<td>0.08</td>
<td>3.48</td>
<td>37.34</td>
</tr>
<tr>
<td>Multiview stereo (center view crosshair)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datatment only</td>
<td>44s</td>
<td>0.74</td>
<td>0.043</td>
<td>0.93</td>
<td>3.48</td>
<td>18.73</td>
<td>0.58</td>
<td>8.77</td>
<td>51.44</td>
</tr>
<tr>
<td>TV-L¹ smoothing</td>
<td>1.6s</td>
<td>0.91</td>
<td>0.021</td>
<td>0.46</td>
<td>1.64</td>
<td>10.91</td>
<td>0.18</td>
<td>4.14</td>
<td>45.21</td>
</tr>
<tr>
<td>Global optimization</td>
<td>5.5h</td>
<td>0.92</td>
<td>0.023</td>
<td>0.57</td>
<td>1.71</td>
<td>8.34</td>
<td>0.09</td>
<td>3.17</td>
<td>39.64</td>
</tr>
</tbody>
</table>

1 All algorithms implemented on the GPU, running on a nVidia GTX 580 hosted on an Intel Core i7 board.
2 The structural similarity measure (SSIM) ranges from -1 to 1, larger values mean higher similarity.
3 Tuned for optimal accuracy instead of speed: three channel RGB structure tensor, convolution kernel size 9 x 9.
4 By design, the algorithm always computes disparity for all views at the same time.
5 Run time for optimization only, excludes computation of the data term (same for subsequent optimization methods).

Fig. 8: Average disparity reconstruction accuracy and speed for our method compared to a multi-view stereo method on our light field benchmark database. Parameters for all methods were tuned to yield an optimal structural similarity (SSIM) measure [32] (bolded column), but for completeness we also show mean squared disparity error (MSE), as well as the percentage of pixels with disparity or depth error larger than a given quantity. Our method is the only one which is fast enough to yield disparity maps for all 9 x 9 views at near-interactive frame rates, while also being the most accurate. Results on individual data sets can be observed in figure 9.

In figure 7 we observe the distribution of the errors, and can see that almost all large-scale error is concentrated around depth discontinuities. This is a quite common behaviour of depth reconstruction schemes, and improving it a central topic of possible further investigations.

4.4 Comparison to multi-view stereo
Our method uses a paradigm which is quite different from multi-view stereo, so it is of course of interest how it fares in competition to these methods. We therefore compare it to a straight-forward stereo data term, while using the exact same smoothing and optimization schemes for all data terms. Since in the proposed algorithm, we restrict the computation of the local disparity to slices in s- and t-direction, it is also of interest how many views are actually required to produce optimal results. The question is whether by means of this restriction we do not throw away potentially useful information. However, during optimization, all local information is integrated into a global functional, and spatially propagated. We will see that using more views for the local data term than the ones in those two slices does not actually improve the optimized results anymore.

**Competing method.** We compute a local stereo matching cost for a single view as follows. Let $V = \{(s_1, t_1), \ldots, (s_N, t_N)\}$ be the set of $N$ view points with corresponding images $I_1, \ldots, I_N$, 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 $\Lambda$ 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 $\rho_{all}(x, l)$ for label $l \in \Lambda$ at location $x \in I_c$ computed on all neighbouring views is then given by

$$\rho_{all}(x, l) := \sum_{(s_n, t_n) \in V} \| I_n(x + lv_n) - I_c(x) \|, \quad (11)$$

where $v_n := (s_n - s_c, t_n - t_c)$ is the view point displacement. 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

$$\rho_{crosshair}(x, l) := \sum_{v_n = (s_c, t_c) \in V} \| I_n(x + lv_n) - I_c(x) \|. \quad (12)$$

In effect, this cost function thus uses exactly the same number of views as we do to compute the local structure tensor.

The local cost function can be used to compute fast point-wise results, optionally smoothing them afterwards, or also integrated into a global energy functional

$$E(u) = \int_\Omega \rho(x, u(x)) \, dx + \lambda \int_\Omega |Du| \quad (13)$$

for a labeling function $u : \Omega \rightarrow \Lambda$ on the image domain $\Omega$, which is solved to global optimality using the method in [24].

**Experiments and discussion.** In figures 8 and 9, we 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 [32], 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 multiview stereo data terms achieve the same accuracy, see figure 8 - 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 is also the fastest, achieving near-interactive performance for computing disparity maps for all of the views simultaneously. Note that in figure 8, we give computation times when our method is tuned for maximum quality (i.e. three channel RGB structure tensor with a convolution kernel size of $9 \times 9$). At the loss of some accuracy, one can work with grayscale light fields (three times faster) or reduce the convolution kernel size (again up to three times faster). 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.

### Table 1: Structural Similarity Error for Different Methods on Individual Scenes from Our Benchmark Data Set

<table>
<thead>
<tr>
<th>Scene</th>
<th>Data term only</th>
<th>Global optimum</th>
<th>Data term only</th>
<th>Global optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buddha</td>
<td>0.86</td>
<td>0.94</td>
<td>0.91</td>
<td>0.94</td>
</tr>
<tr>
<td>Horses</td>
<td>0.75</td>
<td>0.92</td>
<td>0.84</td>
<td>0.92</td>
</tr>
<tr>
<td>Medieval</td>
<td>0.61</td>
<td>0.93</td>
<td>0.85</td>
<td>0.97</td>
</tr>
<tr>
<td>Mona</td>
<td>0.85</td>
<td>0.91</td>
<td>0.90</td>
<td>0.94</td>
</tr>
<tr>
<td>Papillon</td>
<td>0.72</td>
<td>0.90</td>
<td>0.83</td>
<td>0.93</td>
</tr>
<tr>
<td>StillLife</td>
<td>0.89</td>
<td>0.91</td>
<td>0.92</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Fig. 9: Structural similarity error for different methods on individual scenes from our benchmark data set. All light fields have $9 \times 9$ views, with image resolutions between $768 \times 768$ and $1024 \times 768$. The multi-view stereo method makes use of all views to compute the data term. One can see that we obtain in most cases more accurate results in much shorter computation time.
5 SUPER-RESOLUTION VIEW SYNTHESIS

In this section, we propose a variational model for the synthesis of super-resolved novel views. To the best of our knowledge, it is the first of its kind. 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 [14], which computes super-resolved textures for a 3D model from multiple views, and shares the same favourable 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 \( v_i : \Omega_i \rightarrow \mathbb{R} \) of a scene available, which are obtained by projections \( \pi_i : \mathbb{R}^3 \rightarrow \Omega_i \). Each pixel of each image stores the integrated intensities from a collection of rays from the scene. This subsampling process is modeled by a blur kernel \( b \) for functions on \( \Omega_i \), and essentially characterizes the point spread function for the corresponding sensor element. It can be measured for a specific imaging system [2]. 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 \( u : \Gamma \rightarrow \mathbb{R} \) of the light field from a novel view point, represented by a camera projection \( \pi : \mathbb{R}^3 \rightarrow \Gamma \), where \( \Gamma \) is the image plane of the novel view. The basic idea of super-resolution is to define a physical model for how the subsampled 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 \( u \).

5.1 Image formation and model energy

In order to formulate the transfer of information from \( u \) to \( v_i \) correctly, we require geometry information [8]. Thus, we assume we know (previously estimated) depth maps \( d_i \) for the input views. A point \( x \in \Omega_i \) is then in one-to-one correspondence to 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 10.

The process explained above induces a backwards warp map \( \tau_i : \Omega_i \rightarrow \Gamma \) which tells us where to look on \( \Gamma \) for the color of a point, as well as a binary occlusion mask \( m_i : \Omega_i \rightarrow \{0,1\} \) which takes the value 1 if and only if a point in \( \Omega_i \) is also visible in \( \Gamma \). Both maps only depend on the scene surface geometry as seen from \( \pi_i \), 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 11.

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 downsampling kernel, we obtain \( v_i \approx b \ast (u \circ \tau_i) \) on the subset of \( \Omega_i \) 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

\[
E(u) = \sigma^2 \int_{\Gamma} |Du| + \sum_{i=1}^{n} \frac{1}{2} \int_{\Omega_i} m_i(b \ast (u \circ \tau_i) - v_i)^2 \, dx.
\]

which is the MAP estimate under the assumption of Gaussian noise with standard deviation \( \sigma \) on the input images. It resembles a classical super-resolution model [2], which is made slightly more complex by the inclusion of the warp maps and masks. In the energy (14), the total variation acts as a regularizer or objective prior on \( u \). Its main tasks are to eliminate outliers and enforce a reasonable inpainting 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 leads to a convex model which can be very efficiently minimized.

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 \( \tau_i \) 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 \( \Gamma \), see figure 11. We obtain for the derivative of a single term of the sum in (14)

\[
\begin{align*}
    dE^i_{\text{data}}(u) &= |\det D\beta_i| \ (m_i \tilde{b} * (b * (u \circ \tau_i) - v_i)) \circ \beta_i \\
    &= |\det D\tau_i|^{-1} \circ \beta_i, 
\end{align*}
\]

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 [14].

The term \( |\det D\beta_i| \) in equation (15) introduces a pointwise weight for the contribution of each image to the gradient descent. However, \( \beta_i \) depends on the depth map on \( \Gamma \), 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 \( \Omega_i \) and multiply it with \( m_i \) to create a single weighted mask. Note that

\[
|\det D\beta_i| = |\det D\tau_i|^{-1} = |\det D\tau_i|^{-1} \circ \beta_i. 
\]

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

\[
\begin{align*}
    dE^i_{\text{data}}(u) &= (m_i \tilde{b} * (b * (u \circ \tau_i) - v_i)) \circ \beta_i \\
    &= |\det D\tau_i|^{-1}. \quad \text{An example weighted mask is visualized in figure 11.}
\end{align*}
\]

\[\text{Fig. 11: Illustration of the terms in the super-resolution energy. The figure shows the ground truth depth map for a single input view and the resulting mappings for forward- and backward warps as well as the visibility mask } m_i. \text{ White pixels in the mask denote points in } \Omega_i \text{ which are visible in } \Gamma \text{ as well.}\]

5.2 Specialization to 4D light fields

The model introduced in the previous section is hard to implement efficiently in fully general form. This paper, however, focuses 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 to the depth, as explained in section 3.

How the disparity maps are obtained does not matter, but in this work, naturally, they will be computed using the technique described in the previous section.

View synthesis in the light field plane. The warp maps required for view synthesis become particularly simple when the target image plane \( \Gamma \) lies in the common image plane \( \Omega \) of the light field, and \( \pi \) resembles the corresponding light field projection through a focal point \( c \in \Pi \). In this case, \( \tau_i \) is simply given by a translation proportional to the disparity,

\[
\tau_i(x) = x + d_i(x)(c - c_i),
\]

see figure 12. Thus, one can compute the weight in equation (17) to be

\[
|\det D\tau_i|^{-1} = |1 + \nabla d_i \cdot (c - c_i)|^{-1} 
\]

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 \( \Delta c \) 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 12. If $\Delta c \cdot \nabla d_i$ gets closer to $-1$, then more information from $\Omega_i$ is being condensed onto $\Gamma$, 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 $\Gamma$. 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, 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 (19) 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 [19], [12], [17], [25].

5.3 Super-resolution results

For the optimization of the (convex) energy (14), we transform the gradient to the space of the target view via equation (17), discretize, and employ the fast iterative shrinkage and thresholding algorithm (FISTA) found in [3]. All steps are explained in our previous work [36], and an implementation is available on our web site, so we omit the details here for brevity. 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 have subpixel accuracy, since this is a necessary requirement for super-resolution to work.

View interpolation and super-resolution. 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 14, 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 downsampling kernel $b$, we use a simple box filter of size equal to the downsampling 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 4.1, 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 $3 \times 3$ 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 [9] of the center view synthesized in the first experiment. While the reconstruc-
Fig. 14: 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 $3 \times 3$, and for bilinear interpolation (IP) and TV-Zooming (TV) [9] to $3 \times 3$ 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 6. Input image resolution is $384 \times 384$.

<table>
<thead>
<tr>
<th>Views</th>
<th>Conehead</th>
<th></th>
<th>Buddha</th>
<th></th>
<th>Mona</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 5$</td>
<td>31.6 29.3 27.4 26.5</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
<tr>
<td>$9 \times 9$</td>
<td>31.6 29.4 27.5 26.5</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
<tr>
<td>$17 \times 17$</td>
<td>31.2 30.4 27.3 26.0</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
<tr>
<td>$5 \times 5$</td>
<td>31.1 29.3 27.1 25.8</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
<tr>
<td>$9 \times 9$</td>
<td>31.4 29.4 27.6 26.2</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
<tr>
<td>$17 \times 17$</td>
<td>31.5 30.9 25.9 24.3</td>
<td>TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td>1x1 3x3 TV IP</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 15: Closeups of the upsampling 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 [9], super-resolved reconstruction. The super-resolved result shows increased sharpness and details.

Figures 19 and 18 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 $9 \times 9$ views using the method in [34]. Since no ground truth for the scene is available, the input views were downsampled 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 subpixel disparity as well as a high-quality super-resolved intermediate view.

**Disparity refinement.** As we have seen in figure 16, 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 16 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 17. 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.

6 CONCLUSIONS

We developed a continuous framework for light field analysis which allows us to both introduce novel data
terms for robust disparity estimation, as well as the first fully continuous model for variational super-resolution view synthesis. Disparity is estimated locally using dominant directions on epipolar plane images, which are computed with the structure tensor. The local estimates can be consolidated into global disparity maps using state-of-the-art convex optimization techniques. Several such methods are compared, trading off more and more modeling accuracy and sophistication for speed.

We also give a detailed analysis about optimal parameter choices, the ideal use cases as well as limitations of the method. As expected, the method is best suited to densely sampled light fields, as for example obtained by recent commercial plenoptic camera models. Experiments on new benchmark data sets tailored to the light field paradigm show state-of-the-art results, which surpass a traditional stereo-based method in both accuracy as well as speed.

The subpixel-accurate disparity maps we obtain are the pre-requisite for super-resolved view synthesis. As a theoretical novelty, we can within our framework analytically derive weighting factors for the contributions of the input views caused by foreshortening effects due to scene geometry. Extensive experiments on synthetic ground truth as well as real-world images from a recent plenoptic camera give numerical evidence about the competitive performance of our method, which is capable of achieving near-interactive frame rates.

REFERENCES

Fig. 18: 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. Numerical quality of the estimate is computed in figure 19. 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 [3], super-resolved reconstruction. One can find additional detail, for example the diagonal stripes in the Euro note, which were not visible before.

Sven Wanner received his Diploma from the University of Heidelberg for his work on “Interactive rendering of data from wind-driven watersurfaces and event classification”. His main research interests lie in the topic area of light field image processing. Currently, he focuses on developing algorithms for robust 3D reconstruction and segmentation of light fields within the Lumigraph parametrization, which he obtains for different kinds of raw data from camera arrays, plenoptic cameras as well as from simulation. He is a member of the “Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences” (HGS).

Bastian Goldluecke received a PhD on “Multi-Camera Reconstruction and Rendering for Free-viewpoint Video” from the MPI for computer science in Saarbrücken in 2005. Subsequently, he held PostDoc positions at the University of Bonn and TU Munich, where he developed variational methods and convex optimization techniques for high-accuracy 3D and texture reconstruction. His specialty are GPU implementations of efficient algorithms for variational inverse problems and multilabel problems, for which he develops and maintains the open source library cocolin. In current research, he focuses on variational methods for light field analysis, and he is heading the corresponding junior research group at the Heidelberg Collaboratory for Image Processing.