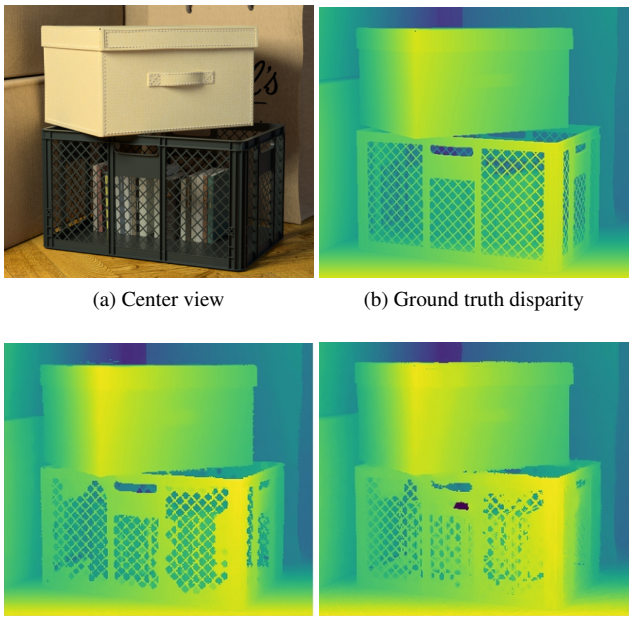
Trust your Model: Light Field Depth Estimation with inline Occlusion Handling

Hendrik Schilling, Maximilian Diebold, Carsten Rother, Bernd Jähne
Heidelberg Collaboratory for Image Processing (HCI)
hendrik.schilling@posteo.de

Abstract

We address the problem of depth estimation from light-field images. Our main contribution is a new way to handle occlusions which improves general accuracy and quality of object borders. In contrast to all prior work we work with a model which directly incorporates both depth and occlusion, using a local optimization scheme based on the PatchMatch algorithm. The key benefit of this joint approach is that we utilize all available data, and not erroneously discard valuable information in pre-processing steps. We see the benefit of our approach not only at improved object boundaries, but also at smooth surface reconstruction, where we outperform even methods which focus on good surface regularization. We have evaluated our method on a public light-field dataset, where we achieve state-of-the-art results in nine out of twelve error metrics, with a close tie for the remaining three.



(a) Center view (b) Ground truth disparity

(c) **OURS** (d) SPO-MO, Sheng *et al.* [8]

Figure 1: **Improved reconstruction** through our inline occlusion handling approach, in comparison with Sheng *et al.* [8, SPO-MO]. Note the considerably improved reconstruction of the partially occluded content within the box and on the right side of the box. The improvement can also be measured quantitatively by the percentage of bad pixels (error > 0.07 px), here 10.8 for ours and 15.5 for Sheng *et al.*

1. Introduction

Depth estimation from multiple images is a central task in computer vision, with a long-standing history. Depending on the application area, different types of depth sensors are utilized, ranging from stereo cameras, over depth cameras, to light field cameras. If depth accuracy is the most important factor, compared to e.g. financial budget or portability, then light field cameras are the best choice. This is true for various application scenarios, such as special effects for movies.

Light-field imaging allows for highly accurate depth estimation, by sampling a scene from many viewpoints. The oversampling increases depth accuracy and the large number of viewpoints reduce the chance of encountering a sample which is occluded in all other views. As for related tasks, such as stereo and optical flow, proper occlusion handling is essential for obtaining high-quality depth reconstructions. An inaccurate occlusion model will immediately reduce the reconstruction quality, since foreground and background samples are confused within the data-term around object boundaries. This is a well-known problem and virtually all state-of-the-art methods for light-field depth estimation implement some form of occlusion handling. However, they differ in the way how they perform this. Proper occlusion handling is the *main* topic of this work.

One may think of three different paradigms to handle occlusion, each with a different level of complexity. At one end of the spectrum there would be approaches which formulate an elaborate model for jointly estimating depth and occlusions, ideally for all views jointly. This explicit joint optimization has been formulated by Kolmogorov and Zabih [5], however their approach is prohibitively slow with existing solvers, even when restricting the problem to stereo

and single pixel accuracy [10]. Hence, we are not aware of any practical realization of such an approach for light-field imaging. At the other end of the spectrum, there are all the existing approaches to light-field depth estimation. In a nutshell, they employ a pre-processing step to filter out all potentially occluded pixels in each view. The way to achieve this differs, however. After this pre-processing step one (or sometimes multiple) cost volume(s) are derived (explicitly or implicitly) from the image data. The cost volume(s) are then used to derive the depth for *e.g.* the center view of the camera. The hope is that the cost volume is free of the influence of occlusion. Obviously, such a two stage procedure is sub-optimal for various reasons. One major problem is that wrongly discarded non-occluded pixels are lost for the remaining computation steps.

The aim of this work is to find a way to handle occlusions in a more integrated fashion than existing approaches, and in this way to make the most use of the available data. At the same time, we obviously need a computationally feasible procedure which estimates depth in the presence of a model which contains the complex interactions of occlusion. To achieve this we borrow from PatchMatch [1], which can optimize Markov Random Field models where spatial terms of the objective function do not need to be pre-computed. In our case these spatial terms involve the traditional data-term, but subject to the occlusion information of neighboring pixels. In effect, we continuously update the occlusion information during the processing, which means that it is always consistent with the estimated depth, and by virtue of this synchronization the occlusion information is implicitly improved during the processing. In PatchMatch the local errors directly sum up to a global energy which is implicitly minimized, as there are no local interactions. However, while we also perform only local evaluations and updates, because of the interaction between depth model and occlusion, these local updates do not give any guarantees with respect to the global error. By using PatchMatch we are able to achieve our goal of efficiently estimating a depth model where occlusion information does not have to be pre-computed. By doing so, we observe a substantial improvement in reconstruction quality, both qualitatively and quantitatively. Interestingly, our improvements are not only located at object boundaries, but also the quality of interior surface reconstruction improves. This stems from the fact that we can make better use of the available data than other methods, even those methods with a strong focus on regularization.

In the following we summarize our main contributions:

- We present a new way to perform occlusion handling for light-field depth estimation, by directly integrating occlusions into the depth model. Compared to all prior methods, this maximizes the use of the available data.

- Despite the complex occlusion model a PatchMatch [1] based scheme based on local updates is able give good estimates on this model, and in competitive processing time.

- Although the method does not guarantee globally optimal solutions, we achieve state-of-the-art results in nine out of twelve error metrics, for a publicly available benchmark, with a close tie for the remaining three.

In addition, our approach can easily be extended with additional depth cues or model constraints. This is demonstrated by combining our approach with a *normals-from-specular* approach [2], resulting in accurate depth reconstructions for a glossy, untextured object.

2. Related Work

In the following we briefly introduce existing approaches, focusing our description on the occlusion handling.

Where the methods are also included in the quantitative evaluation, the abbreviation is noted in square brackets. Abbreviations are identical to the ones submitted by the respective authors to the 4D Lightfield Benchmark [3, 4] and all method results, including ours, can also be compared on the benchmark website [3].

Neri *et al*. [7, RM3DE] perform multi-resolution block matching, adapting the window size with some local gradient measure, and performing matching independently for different viewpoint directions from the center view. Occlusions are handled by using only the best match from the directional EPIs for the final median filter based post-processing.

Lin et al. [6] build a focal stack from the light-field data, and exploit the symmetry around the true depth in the stack to provide depth estimates, which are then optimized in a cost volume. A heuristic is employed to generate a separate occlusion map which is used to switch to an alternate cost for occluded pixels prior to the cost volume optimization.

Strecke *et al*. [9, OFSY_330/DNR] extend on this idea by improving the occlusion handling using four partial focal stacks representing the four viewpoint directions of a cross hair subset of the light field, and using only the minimal cost from the horizontal and vertical direction, which should be less affected by occlusions. The method is notable for the explicit optimization of surface normals in addition to depth, which improves the surface quality of the reconstruction.

Williem and Park [14] introduce two independent cost functions. Angular entropy, which is a correspondence cost based on the entropy of photo-consistency, and an adaptive defocus cost, both of which show some robustness against occlusion. Reconstruction is then based on cost-volume filtering with graph cut. In a later work they improve this method, [15, CAE] modifying both cost functions to further improve the robustness against occlusion.

The Spinning Parallelogram Operator by Zhang *et al*. [16, SPO] scans the depth volume with a histogram com-

parison operation, which compares the areas left and right of the EPI line, defined by the respective disparity. This histogram comparison is relatively robust to at least single occlusions, hence no extra occlusion handling is performed in the guided filter based cost volume processing of the local cost estimates. Sheng *et al*. [8, SPO-MO] expand on this approach and add explicit occlusion handling by regarding multi-orientation EPIs and selecting a single unoccluded one for the calculation of the cost volume, according to an occlusion heuristic.

All of these methods make use of some form of cost volume optimization [6, 9, 14, 15, 16, 8], if not using a simple filter based approach [7]. Occlusion handling is always separated from the cost volume optimization and comes in several variants: By using cost functions robust against occlusions [14, 15, 16], by using the minimal cost from several EPI directions [7, 9] or by switching between separate cost functions for occluded/unoccluded samples [6].

The works focusing on cost functions robust to occlusions show an interesting pattern. While the original publications only use the proposed robust cost functions [14, 16]. Later works mainly focus on the occlusion handling either by further improving robustness against occlusion or by adding explicit occlusion handling [9, 8]. It seems that even though cost functions exist which show *some* robustness against occlusion, these cost functions do not return optimal results.

On the other hand, methods that handle occlusions by selecting the minimal cost from several, possibly partial EPIs, discard a lot of samples from the input light field. This reduces the number of samples over which the data cost can be calculated and hence reduces accuracy.

Common to all methods is the fact that the used occlusion information is independent of the final optimized depth estimate. The additional scene knowledge available after optimizing the depth model is not reflected by the used cost function, which is limited to the initial occlusion estimates. Our proposed method addresses this point by using the current model to calculate the occlusions inline, during the processing, and therefore improves the utilization of the available light-field data.

Note that there are other methods which optimize the occlusions, like the works by Wanner and Goldlücke [13, 12] where they filter local depth estimates with a model enforcing global consistency with respect to occlusion. However, the accuracy of this approach is limited by the fact that only local estimates are used as priors in a regularization approach, and no updates on the cost are performed for updates in the occlusion model.

3. Method

Given the fact that the depth model which we try to reconstruct implicitly contains the occlusion information required for proper occlusion handling, we formulate a cost function

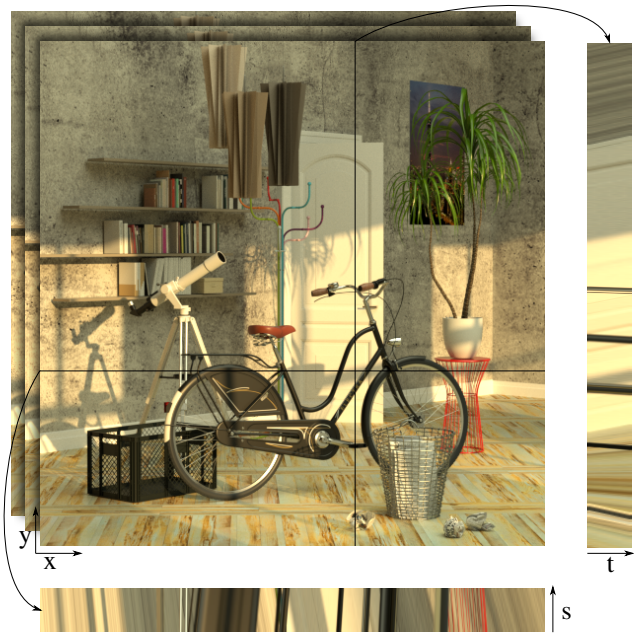


Figure 2: **Epipolar Plane Images (EPIs)** are extracted from a linear 3D subset of the 4D light field, by extracting all rows (for a horizontal subset) and stacking them together, shown at the bottom. For the vertical stack the same is done with columns. Because the apparent motion of scene points between the different viewpoints depends on the depth of the point within the scene, the orientation of features in the EPI encodes the depth of the respective points. Note that the EPI shown here is pre-shifted so a disparity of 0 is not at infinity but rather within the scene, hence disparities may also be negative.

in a way that makes direct use of the occlusion information encoded within the model. This makes occlusion a first class citizen of the model.

This cost could in principle be optimized with some global optimization method. However, as the resultant optimization problem is highly ill-posed, this approach would probably be extremely slow (compare [5, 10]). Therefore we base our approach on PatchMatch [1] to perform only local optimization, and introduce extra constraints into the cost term to avert suboptimal solutions arising from this fast but globally suboptimal optimization.

Apart from the implications of the occlusion handling, our approach is formulated as a standard minimization problem with a cost based on a regularization term and a data term, where both are influenced by the occlusion handling.

3.1. Model and Data

The model we are using is the disparity map of the central view. To simplify occlusion handling we confine the data to

the subset of viewpoints shifted only horizontally or only vertically from the central viewpoint (cross-hair configuration). The volume of the horizontal 3D subset can be sliced row-wise to obtain a set of epipolar plane images (EPIs, compare fig. 2), which represent the full information content of the subset. The central row of an EPI corresponds to a row of the center view, which directly maps to the same row in the disparity map. The same applies to columns in the vertical 3D subset. A single sample from the disparity map corresponds to a 2D line in the respective EPIs, where the slope of the line represents the disparity and hence encodes the depth, compare fig. 3. The cost function $E_i(d)$ for a single sample i of our model (a pixel of the center view disparity map D), based on the data term $\xi_i(d)$ and the regularization term $\zeta_i(d)$ is formulated as the cost associated with a disparity d, where the disparity map D is held constant for the evaluation of the sample:

$$E_i(d) = \rho \cdot \zeta_i(d) + \xi_i(d), \tag{1}$$

where ρ is a regularization weight.

3.2. Occlusion Handling

Compared to the methods in section 2, we obtain occlusion information from our depth model, and not via some heuristic external to the optimization. This simplifies our occlusion metric to a simple threshold θ_d. We consider a disparity sample d in the disparity map to be potentially occluded by any other sample d_i if $d_i - d > \theta_d$.

The actual decision whether a sample is occluded or not is performed during the evaluation of the cost terms, which means that updates to the model performed during an iteration of the optimization directly affect the costs of all future evaluations, which speeds up the propagation of locally good solutions, compare PatchMatch [1].

3.3. Data Term

Because we only consider either horizontal or vertical camera movement, relative to the central view, only samples from the same row (or column, respectively), can occlude any given sample in an EPI, compare fig. 2 and fig. 3. In the following we will always assume that we are looking at horizontal EPIs, but all statements apply to vertical EPIs via a corresponding $90°$ rotation of EPI, view and disparity map.

To evaluate the data error for some disparity d at location i in the disparity map, we sample along the corresponding line $\Gamma_{d,i}(s)$, see fig. 3, by evaluating $\Gamma_{d,i}$ for all rows s of the EPI. A sample $\Gamma_{d,i}(s) = x$ corresponds to a pixel position at the image coordinate (x, i_y) of view s. While i_y is an integer, x is a fraction, hence the actual pixel value $C_s(x, i_y)$ is derived by interpolation in the horizontal direction. To actually calculate the data error we generate all intersections between $\Gamma_{d,i}$ and all other lines $\Gamma_{d,j}$ of the EPI which fulfill

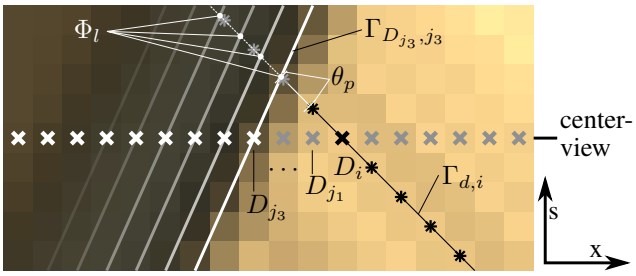


Figure 3: **Occlusion handling in an EPI:** The lines Γ are defined by the respective disparities D_j in the center view, represented by a cross ($\times$), while the EPI samples on $\Gamma_{d,i}$ are shown as star ($\ast$). From the intersections Φ_l (white dots), the one closest to the center view is obtained with Γ_{d,j_3}, hence all samples behind this point minus a safety distance if one pixel are disabled (grayed out).

the occlusion condition in section 3.2. Note that lines from samples to the left of i can only intersect above the center view, while samples to the right can intersect below. Given these left/right intersections as Φ_l and Φ_r, respectively, the occlusion term $nocc(s, \Phi_l, \Phi_r)$ is set to zero or one.

The occlusion area is extended by one pixel from the intersection point, to avoid mixing of foreground and background when deriving the actual color sample $C_s(x, i_y)$ from the input view s via linear interpolation. Given the occlusion terms the data error is simply the variance of all visible samples. We extend the previous definitions by the subscripts h and v to denote the horizontal and vertical EPI variants respectively (following terms with respect to a fixed sample i and a fixed disparity d):

$$\xi'_i(d) = \frac{\begin{array}{l}\sum_s (\mu - C(\Gamma_h, s))^2 \cdot nocc_h(s, \Phi_{h,l}\Phi_{h,r}) + \\ \sum_t (\mu - C(\Gamma_v, t))^2 \cdot nocc_v(t, \Phi_{v,l}\Phi_{v,r})\end{array}}{\begin{array}{l}\sum_s nocc_h(s, \Phi_{h,l}\Phi_{h,r}) + \\ \sum_t nocc_v(t, \Phi_{v,l}\Phi_{v,r})\end{array}}, \tag{2}$$

where μ is the mean of all unoccluded samples for (i, d).

To avoid failures due to the local nature of our approach, we also threshold the data term on the number of unoccluded samples, and set the error to infinity if less than θ_o samples are unoccluded, because otherwise, moving individual samples (incorrectly) towards the background can reduce the variance in flat areas, by reducing the number of unoccluded samples.

Even with this occlusion constraint there is a second case where the local solution can substantially deviate from the correct depth. This can be observed on purely horizontal or vertical structures in the scene. For such structures the data error is zero for one direction, hence, if *e.g.* for a vertical

structure, the vertical component of the data term is zero, then if a large connected block of the vertical structure is moved into the background, the remaining horizontal component also becomes zero because we observe only a single sample in that direction. We protect against this by checking, for each candidate, whether the chosen disparity leads to a single pixel wide background structure, as measured by θ_d over a range of 10 pixels. If such a case is detected the error is set to infinity.

3.4. Smoothness Term

For a disparity sample d at location i in the disparity map, the smoothness error is defined by:

$$\zeta_i(d) = (d - \Omega_i(d))^2 \tag{3}$$

Where Ω is a smoothing filter based on the bilateral filter. This filter smooths the disparity map using a weighted mean, with weights derived from the color and disparity difference against a central sample. The filter uses hard thresholds θ_d and θ_c to determine which samples are allowed to influence the smoothing, which gives well defined borders without disparity bleeding. Given the color values of the center view as C, and the current disparity map as D, the smoothing filter Ω is given by:

$$\Omega_i(d) = \frac{\sum_j \lambda_{i,j}(d) \cdot D_j}{\sum_j \lambda_{i,j}(d)}, \tag{4}$$

where j indexes a 7×7 window around i.

The relative weight $\lambda_{i,j}(d)$ of the disparity map sample D_j is calculated depending on the color difference $\Delta_{i,j} = \alpha|C_i - C_j|$ and the disparity difference $\delta_j(d) = \beta|d - D_j|$ between the sample j and the central sample i, with α and β as parameters which steer the relative weighting of color and disparity differences. The weights are calculated as

$$\lambda_{i,j}(d) = \max\{\epsilon_d, \sqrt{\Delta_{i,j}^2 + \Delta_{i,j} \cdot \delta_j(d)}\}^{-1}, \tag{5}$$

if $\Delta_{i,j} \leq \theta_d$ and $\delta_j \leq \theta_c$, and

$$\lambda_{i,j}(d) = \max\{\epsilon_c, \sqrt{\Delta_{i,j}^2 + \delta_j^2(d)}\}^{-1}, \tag{6}$$

if $\frac{\Delta_{i,j}}{\beta} > \theta_d$ and $\delta_j \leq \theta_c$. Otherwise $\lambda_{i,j}(d)$ is set to zero. The thresholds θ_d and θ_c set the maximum difference for disparity based weighting (if $\frac{\Delta_{i,j}}{\beta} \leq \theta_d$ and $\delta_j \leq \theta_c$) or color based weighting (if $\frac{\Delta_{i,j}}{\beta} > \theta_d$ and $\delta_j \leq \theta_c$).

The ϵ are used to provide damping against zero differences, and ϵ_c also provides some adaption to noise in the input images, using $\epsilon_c = \epsilon_d + \theta_e \cdot E'_i(d_0)$, where E'_i is identical to E_i, aside from changing ϵ_c to $\epsilon_c = \epsilon_d$. Hence $E'_i(d_0)$ is the initial error at this iteration, using the initial disparity d_0. This increases the minimal blurring of the smoothing filter, when no good candidates where found in the previous iteration - which after a few iterations is mostly due to noise in the input images.

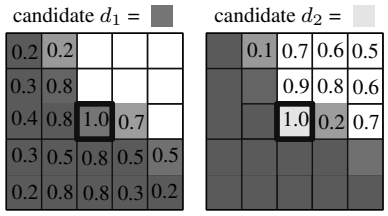


Figure 4: **Switching behavior of the smoothness term.** The two grids represent the identical neighborhood around a central disparity sample d, indicated by the brightness of the cells. Depending on the value of a candidate d_i, the weights, given as numbers within the cells, change according to eqs. (5) and (6), which by design leads to a distribution which generates a smoothing of those samples most similar to the central candidate in both color and disparity.

The crucial part is the usage of the current disparity candidate d within the filter, which lets the smoothing filter adapt to the value of the candidate. The current disparity at i from the model, D_i is not used during the evaluation. This means that the smoothness term can switch, for example at an object border, from averaging over the foreground to averaging over the background, depending on the evaluated disparity candidate, as shown in fig. 4.

The thresholds encourage the smoothing according to the model (*i.e.* disparity map) by making the disparity difference the dominating weight term for small disparity differences ($\frac{\Delta_{i,j}}{\beta} \leq \theta_d$). The color differences play a secondary role and encourage smoothing along similar colors. At the same time the hard thresholds mean that the weight is quickly set to zero if the differences in color and/or disparity become too large, ensuring that only those samples are taken into account for which it is likely that they belong to the same object, both from the color and the disparity similarities.

The simple smoothness term as described above limits the estimation accuracy in two ways. Firstly, the method tends to over-smooth at object edges when both sides of the object are visible, because the edge of the object will be averaged with the neighbors from both sides. Secondly, planes with a steep inclination tend to show staircase artifacts, as the thresholding in the filter encourages areas to be piecewise planar.

We extend the filter to preserve normals and planes separately. In the smoothing filter, consistent normals between the central sample i and some other sample j are detected by comparing the local gradients in D. If the gradient difference is below θ_g, then D_j is corrected by this normal when it is used in eq. (4).

For planar surfaces we add an metric which detects purely

planar surfaces, by taking four samples around the central sample, located at the corners of a square with a size of 11×11, and fitting a plane through these four corners. If the residual from the fit is below θ_f and the distance between the plane and disparity candidate are below θ_d, we evaluate the plane at i and use this result instead of Ω.

Both of these metrics are applied with a damping factor, where the correction with normal and plane is weighted with the original smoothing filter with a weight of 0.5 to prevent overshooting.

3.5. Local Optimization

Both the data term and the smoothness term are formulated with a strong focus on correct occlusion handling with hard thresholds in disparity and color differences. While this encourages well defined borders in the model, it makes the problem harder to optimize, owing both to the sudden onset of the influence of samples, and to the complex interaction between samples due to occlusion. Pre-calculating the error terms for a number of discrete disparity labels and building a cost volume is also not possible, as both terms deliberately depend on the current state of the model. Therefore we base our method on PatchMatch [1]. The method iterates the disparity map and, at each sample, calculates the local error E_i for the current disparity d_0, as well as for several disparity candidates. If any of the candidates has a lower error over the previous solution, the model is immediately updated, which allows propagation of locally good solution.

We use four predictors to provide the disparity candidates which are evaluated with the local error term.

Propagation: Depending on the iteration number, the solver iterates over the disparity map either left-to-right and top-to-bottom, or the reverse. The disparities of all neighbors (either direct or over the corner) which were already processed in the current iteration are used as candidates for evaluation. As the model is always directly updated when a lower error is found, an improved estimate at one sample will directly be used in the data and smoothness term of the next sample, within the same iteration. Hence, as the improved disparity at a sample is provided as a candidate to the solver for the next sample, good solutions can quickly spread over the whole disparity map.

Random improvement: At each iteration, candidates d_i are generated by sampling u from a uniform distribution between -1 and 1 as:

$$d_i = d_0 + \tau \operatorname{sign}(u) u^2 \tag{7}$$

where τ is the parameter which steers the max range of the refinement. The quadratic term ensures that smaller changes are sampled with a much higher frequency than larger ones.

The following two predictors are only activated if the error of the current model is above an activation threshold θ_a.

Random neighbor: For some scenes a feasible candidate might be not directly adjacent but further away, *e.g.* when a surface is partly occluded by some detailed foreground object, like a smooth background behind the branches of some plant. For this reason we also use distant neighbors, by sampling uniformly within a range of ± 15 px.

Random Guess: Finally we also sample randomly from the valid disparity range.

3.6. Initialization

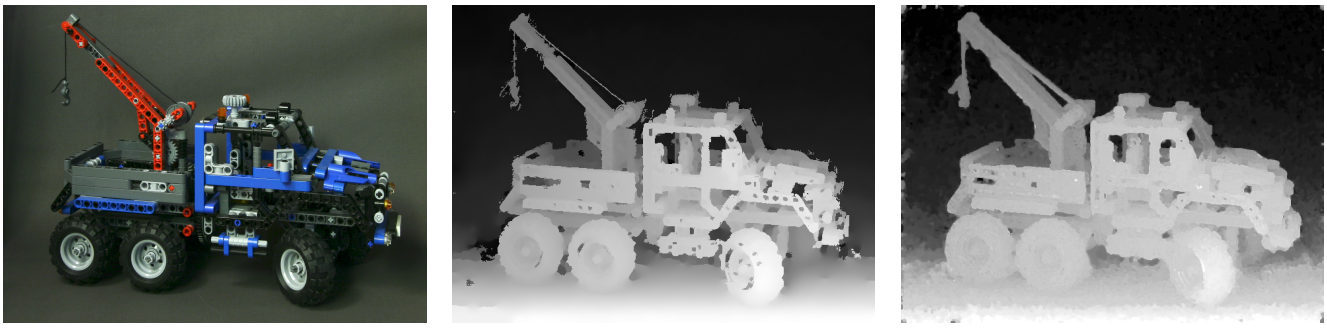
Both data and smoothness term require a model which is at least approximately correct, as they rely on the model to determine occlusion. As initialization we use a simple depth estimation method, based on RANSAC line fits in the EPI. The fitted line features are the zero crossings of the second order derivative in the horizontal direction. This method only detects foreground objects and produces a sparse depth estimate consisting of object borders and strong features. To retrieve an initialization of the disparity map, these sparse estimates are projected into the disparity map and missing samples are linearly interpolated from the sparse set. The initialization is very fast, quite smooth, fills flat areas from samples of the object borders and tends to produce foreground biased estimates.

4. Experimental Results

We have tested our method on several light-field datasets, including real and synthetic data. In the following we describe the results in more detail and demonstrate the improved occlusion handling, see figs. 1, 5 and 7, but also the excellent surface regularization, see figs. 5, 7 and 8, owing in part to the improved utilization of data from the input light field, as we discard less information due overzealous occlusion handling, as well as to the improved detection of object borders. More results of our method are available on the website of the 4D Lightfield Benchmark [3]. All results presented here use 20 iterations and, apart from fig. 5 use the parameters shown table 1.

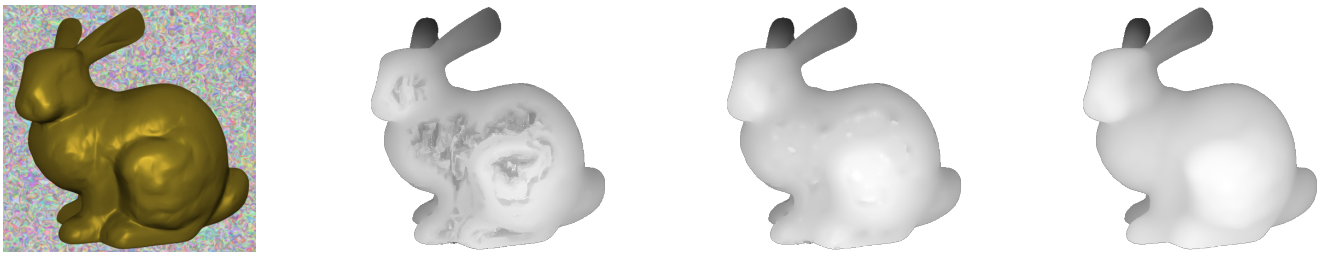
4.1. Qualitative Results

In fig. 5 we show our results on the truck scene from the (new) Stanford Light Field Archive [11]. For comparison we also show the result of Strecke *et al*. [9] (OFSY). While the results leave room for improvement, the detail reconstruction shows the effectiveness of the occlusion handling. At the same time the regularization is also improved, which is otherwise a strength of OFSY (compare fig. 7). We have also combined our method with a *normals-from-specular*



(a) center view (b) **OURS** (c) OFSY_330/DNR, Strecke *et al*. [9]

Figure 5: **Disparity estimates on the truck dataset [11]**, which is challenging due to the large amount of noise, therefore (b) was computed with a version of the dataset scaled down to half size in the spatial domain. Note that although our method uses half size images, the reconstruction is much more detailed, see for example the rope at the top left, or the structure below the driver cab. Smoothing is also improved, although some artifacts remain, like the rough ground before and behind the truck, or the "fireflies" around some object edges. The hole at the back of the cargo area is wrong with both methods because there is a specular reflection visible from several viewpoints.



(a) Center view of a specular object

(b) Our approach with color constancy

(c) Our approach combined with *normals-from-specular* from Gutsche *et al*. [2]

(d) Ground truth disparity

Figure 6: **Disparity estimates when integrating *normals-from-specular* [2]** within our optimization, tested on a synthetic dataset, where we know the exact location of the light source. In (b) the assumption of color constancy does not allow reliable depth estimates in the presence of specular reflections. In (c) specularity is exploited to obtain surface normals.

θ_d	$0.05K$	θ_g	$0.025K$	θ_f	$0.01K$
θ_c	3	θ_o	$0.25V$	θ_a	0.01
α	0.15	β	20	ϵ_d	0.5
ρ	$0.0375I$	τ	$0.2K$	θ_e	400

Table 1: List of parameters used for all results but fig. 5, where V is the total number of views, K the disparity range of the scene and I the current iteration number.

method [2] to enable depth estimation in the presence of glossy reflections, shown in fig. 6. For this we exploit our local optimization approach by exchanging the data term with the fit error of [2] in glossy regions. We still employ the same smoothing term, just augmented with the normals returned by the *normals-from-specular* solver. The result still shows some artifacts, but also highlights the gains in exploiting reflectance information from the light field for depth reconstruction.

4.2. Quantitative Results

The quantitative evaluation is based on the public 4D Lightfield Benchmark by Honauer *et al*. [3]. The benchmark does not report a single score, but instead calculates 12 different error metrics, which consider a range of different failure cases, using well known global metrics like $BadPix$ and MSE, but also surface quality metrics, and more specific errors metrics, like fine thinning/fattening. For details please see their paper [3] and the benchmark survey [4]. The benchmark is performed by generating disparity maps for 12 scenes, 8 of which have publicly available ground truth disparity, while for 4 scenes the ground truth is kept secret. Algorithm results are uploaded to a web-service and all re-

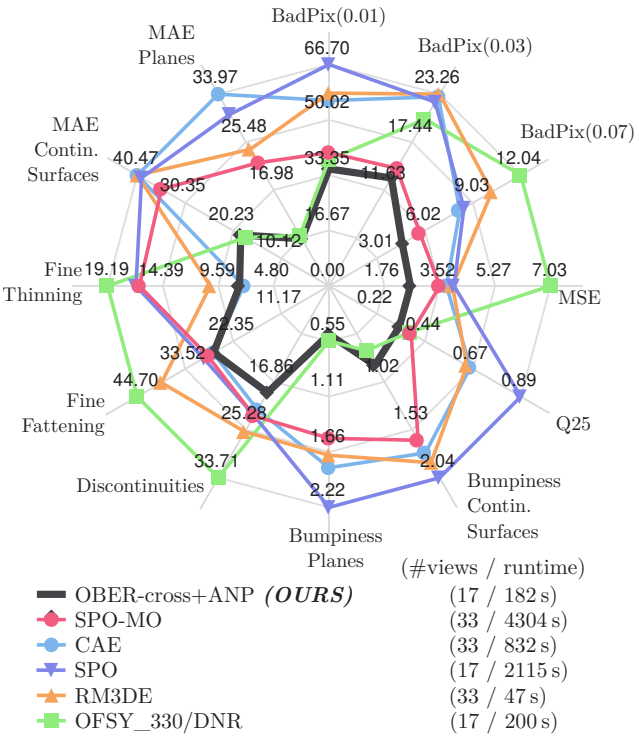


Figure 7: **Mean errors over all twelve benchmark scenes**, evaluated with the twelve error metrics of the 4D Light-field Benchmark [3] and visualized on a radar chart. The legend gives the number of viewpoints and the (approximate) runtime. All metrics are expressed as an average error over twelve datasets. Lower values are better, and located closer to the center. As we can see our method (OBER-cross+ANP) is located closest to the center on average, and manages an improvement over the previous state of the art on most metrics, without exposing a specific weakness. The main challengers which surpass our method in some metric (CAE and OFSY) manage so only by accepting subpar performance on other metrics.

sults, including ours, are available on the benchmark website [3] – our method is abbreviated *OBER-cross+ANP*.

We report our results in comparison to the state of the art, as represented by the top five published methods, when sorted by the average $BadPix_{0.07}$ score, as of 2017/11/11 . The averaged errors over all 12 scenes are shown in fig. 7. Note that our method takes the lead for 9 of the 12 error metrics, and is close behind for the remaining 3.

This is even more remarkable if we consider that several of the error metrics are often traded in against each other, as is the case for bumpiness versus discontinuities and for fine fattening versus fine thinning, which have a strong tendency to revert the order of the methods between the respective error metrics.

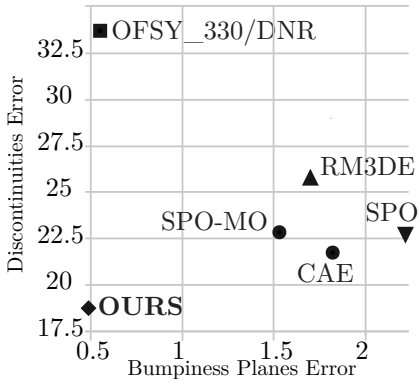


Figure 8: **Trade-off between smoothing and object border preservation**, comparing the *Discontinuities* metric with the *Bumpiness Planes* metric [3]. Results are averages over all 12 benchmark scenes. Note how the good smoothness score for OFSY reflects the focus on the regularization, while the other methods are optimized towards correct object borders. Our method leads both metrics, making the trade-off obsolete.

Indeed by plotting the *Discontinuities* metric, which gives the errors around depth discontinuities, and one of the smoothness metrics, like *Bumpiness Planes*, we can directly evaluate the trade-off between smoothing and preservation of object boundaries, see fig. 8. As we can see all tested methods fall into one extreme, favoring either border handling of smoothing, however our method manages not only to find a favorable trade-off, but instead completely dominates the other methods on both of these metrics.

5. Conclusion

In this work we have presented a new method of depth estimation from light-field images. We inline the occlusion handling into the depth estimation. This represents an improvement over previous methods, which separate occlusion handling and optimization. In addition to the improved data terms we show an efficient method for depth estimation with this type of model, based on PatchMatch. The drawback is that this does not give any guarantees with respect to the global energy. Still, by integrating the occlusion handling we demonstrate a performance increase over the state of the art for object borders as well as for smooth surface reconstruction at a very competitive runtime.

Acknowledgements The work was carried out during a research cooperation between the Computational Imaging Group at the Stuttgart Technology Centre of Sony Europe Limited and the Heidelberg Collaboratory for Image Processing (HCI).

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, July 2009. 2, 3, 4, 6

[2] M. Gutsche, H. Schilling, M. Diebold, and C. Garbe. Surface normal reconstruction from specular information in light field data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1735–1742. IEEE, 2017. 2, 7

[3] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*. Springer, 2016. *http://lightfield-analysis.net*. 2, 6, 7, 8

[4] O. Johannsen, K. Honauer, B. Goldluecke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, M. Gutsche, H.-G. Jeon, I. S. Kweon, A. Neri, J. Park, J. Park, H. Schilling, H. Sheng, L. Si, M. Strecke, A. Sulc, Y.-W. Tai, Q. Wang, T.-C. Wang, S. Wanner, Z. Xiong, J. Yu, S. Zhang, and H. Zhu. A taxonomy and evaluation of dense light field depth estimation algorithms. In *Conference on Computer Vision and Pattern Recognition - LF4CV Workshop*, 2017. 2, 7

[5] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Computer Vision—ECCV 2002*, pages 8–40, 2002. 1, 3

[6] H. Lin, C. Chen, S. Bing Kang, and J. Yu. Depth recovery from light field using focal stack symmetry. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3451–3459, 2015. 2, 3

[7] A. Neri, M. Carli, and F. Battisti. A multi-resolution approach to depth field estimation in dense image arrays. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3358–3362. IEEE, 2015. 2, 3

[8] H. Sheng, P. Zhao, S. Zhang, J. Zhang, and D. Yang. Occlusion-aware depth estimation for light field using multi-orientation epis. *Pattern Recognition*, 2017. 1, 3

[9] M. Strecke, A. Alperovich, and B. Goldluecke. Accurate depth and normal maps from occlusion-aware focal stack symmetry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 2, 3, 6, 7

[10] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. *Computer Vision–ECCV 2006*, pages 16–29, 2006. 2, 3

[11] V. Vaish and A. Adams. The (new) stanford light field archive. *Computer Graphics Laboratory, Stanford University*, 2008. 6, 7

[12] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4d light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE, 2012. 3

[13] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619, 2014. 3

[14] W. Williem and I. Kyu Park. Robust light field depth estimation for noisy scene with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4396–4404, 2016. 2, 3

[15] W. Williem, I. K. Park, and K. M. Lee. Robust light field depth estimation using occlusion-noise aware data costs. *(preprint) IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2, 3

[16] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159, 2016. 2, 3