A Taxonomy and Evaluation of Dense Light Field Depth Estimation Algorithms

Ole Johannsen[1,a], Katrin Honauer[2,a], Bastian Goldluecke[1,a],
Anna Alperovich[1], Federica Battisti[3], Yunsu Bok[4], Michele Brizzi[3], Marco Carli[3],
Gyeongmin Choe[5], Maximilian Diebold[2], Marcel Gutsche[2], Hae-Gon Jeon[5],
In So Kweon[5], Jaesik Park[6], Jinsun Park[5], Hendrik Schilling[2], Hao Sheng[7],
Lipeng Si[8], Michael Strecke[1], Antonin Sulc[1], Yu-Wing Tai[9], Qing Wang[8],
Ting-Chun Wang[10], Sven Wanner[11], Zhang Xiong[7], Jingyi Yu[12], Shuo Zhang[7], Hao Zhu[8]

[1]University of Konstanz, Germany
[2]Heidelberg University, Germany
[3]Roma Tre University, Italy
[4]ETRI, Republic of Korea
[5]KAIST, Republic of Korea
[6]Intel Visual Computing Lab, USA
[7]Beihang University, China
[8]Northwestern Polytechnical University, China
[9]Tencent, China
[10]UC Berkeley, USA
[11]Lumitec, Germany
[12]ShanghaiTech University, China

[a] contact@lightfield-analysis.net

The first two authors contributed equally.

Abstract

This paper presents the results of the depth estimation challenge for dense light fields, which took place at the second workshop on Light Fields for Computer Vision (LF4CV) in conjunction with CVPR 2017. The challenge consisted of submission to a recent benchmark [7], which allows a thorough performance analysis. While individual results are readily available on the benchmark web page http://www.lightfield-analysis.net, we take this opportunity to give a detailed overview of the current participants. Based on the algorithms submitted to our challenge, we develop a taxonomy of light field disparity estimation algorithms and give a report on the current state-of-the-art. In addition, we include more comparative metrics, and discuss the relative strengths and weaknesses of the algorithms. Thus, we obtain a snapshot of where light field algorithm development stands at the moment and identify aspects with potential for further improvement.

1. Introduction

Over the last decade, light field analysis has grown from a niche topic to an active and established part of the computer vision community. The key difference to the classical multi-view scenario is the dense and regular sampling, which allows to develop novel and highly accurate methods for depth reconstruction which can correctly take occlusions into account to recover fine details. In recent years, a variety of algorithms was published [9, 11, 12, 16, 19, 20, 23, 25, 27, 29], but objective comparison of their strengths and weaknesses is not straight-forward.

While the HCI Light Field Benchmark by Wanner et al. [26] provided several popular data sets and could be considered a default for testing in the past three years, evaluation on it was not standardized with respect to which metrics and datasets should be included in the evaluation. To establish a comparable performance analysis, Honauer, Johannsen et al. [7] introduced a novel benchmark with synthetic light field data and a comprehensive evaluation methodology. They presented an evaluation of five algorithms as a baseline but their focus was on validating the proposed scenes and metrics.

In order to capture and analyze a representative state-of-the-art, we initiated the light field depth estimation challenge as part of the second workshop on Light Fields for Computer Vision (LF4CV), held at CVPR 2017. The challenge was open to submissions of novel as well as already published light field methods. Challenge participants submitted their estimated disparity maps and runtimes for the four stratified and eight photorealistic scenes on figure 1. Ground truth is unknown for four of the photorealistic scenes, the other scenes can be used to train parameters, which however have to be the same for all scenes.

In this paper, we present the results of the depth estimation challenge. With seven challenge participants, two additional benchmark submissions, and five baseline submis-

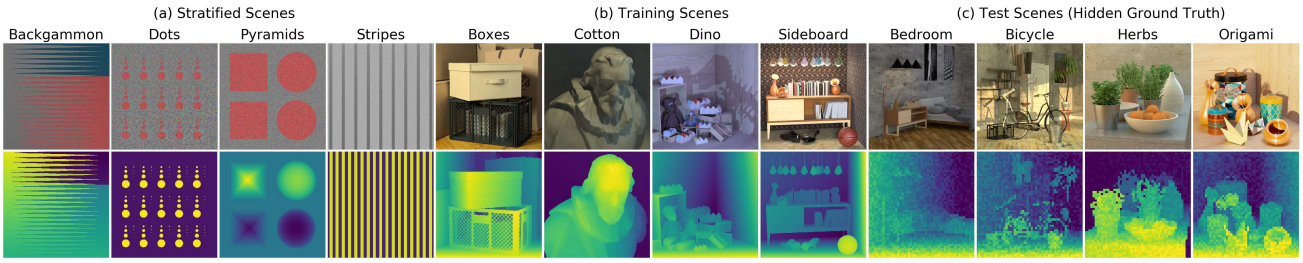

Figure 1. The 12 benchmark scenes used for evaluation: four stratified and four photorealistic training scenes with publicly available ground truth and four test scenes with hidden ground truth.

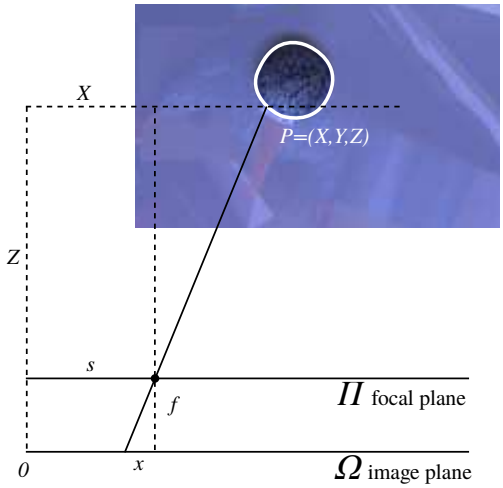

Figure 2. Light field coordinates and projection. Note that by convention, we parameterize (x, y)-coordinates relative to the principal point of a pinhole camera located at (s, t).

sions, we analyze a total of 14 algorithms. We believe this to be a good basis for a taxonomy and thorough evaluation of the current state-of-the-art. In line with [7], the aim is not to identify a single best algorithm for the entire benchmark or for all light field data sets - indeed, we believe this is not possible. Instead, we try to compare relative strengths and weaknesses of different approaches in order to learn which components lead to good results for certain scene properties or structures, and to learn for which scenarios there is still room for improvement.

2. Light field depth estimation

In this section, we will give a short introduction to the structure of light fields and briefly review possible strategies to infer depth. These can roughly be classified according to the different representations of the light field they rely upon, and thus give one category for a taxonomy of algorithms.

For the purpose of this paper, we understand a 4D light field as the radiance function sampled on a space of rays. This 4D ray space is parameterized by the two intersection points of each ray $\boldsymbol{r}$ with two different planes. The *image plane* Ω is parameterized in $\boldsymbol{p} = (x, y)$ coordinates, while the *focus plane* Π is parameterized in $\boldsymbol{c} = (s, t)$ coordinates. Both planes are parallel to each other. Thus, the 4D light field is a function

$$\begin{aligned} &L : \Omega \times \Pi \to \mathbb{R}, \\ &(x, y, s, t) \mapsto L(x, y, s, t) = L(\boldsymbol{p}, \boldsymbol{c}). \end{aligned} \tag{1}$$

In practice, it often has several components, i.e. takes values in RGB color space $\mathbb{R}^3$.

Subaperture views and disparity. The light field can be resampled into several popular representations, see also Levoy [13] for a more fundamental introduction into light field principles and parameterizations. In the maybe most intuitive representation, we fix (s, t) coordinates. If (x, y) coordinates vary, we obtain for each pair (s, t) an image $I_{(s,t)}$ as captured by an ideal pinhole camera. These cameras have parallel optical axes orthogonal to the planes and identical focal length, corresponding to the distance between the planes. The pinhole views obtained in this way are called subaperture images (see figure 3).

Let a 3D scene point be at a distance Z from the focal plane, then the coordinates of its projections in the subaperture views follow the pinhole projection. Two different rays $\boldsymbol{r}_1, \boldsymbol{r}_2$ passing through this point thus are related by

$$L(\boldsymbol{p}_2, \boldsymbol{c}_2) = L(\boldsymbol{p}_1 - \frac{f}{Z}(\boldsymbol{c}_2 - \boldsymbol{c}_1), \boldsymbol{c}_1), \tag{2}$$

see figure 2. The quantity $d = \frac{Z}{f}$ is called the disparity. It relates all rays emanating from a single point in space whose radiance is captured in the light field. Thus, if this 3D point lies on the scene surface and the scene is Lambertian, all those rays should share the same radiance. This assumption is the basis for all disparity estimation algorithms discussed in this paper, and it is exploited to infer depth.

The disparity correspondence relation (2) gives rise to multi-view stereo matching methods, for which there is a vast literature [17]. Indeed, it implies the intensity relationship

$$I_{\boldsymbol{c}_1}(\boldsymbol{p}_1) = I_{\boldsymbol{c}_2}(\boldsymbol{p}_1 - d(\boldsymbol{c}_2 - \boldsymbol{c}_1)) \tag{3}$$

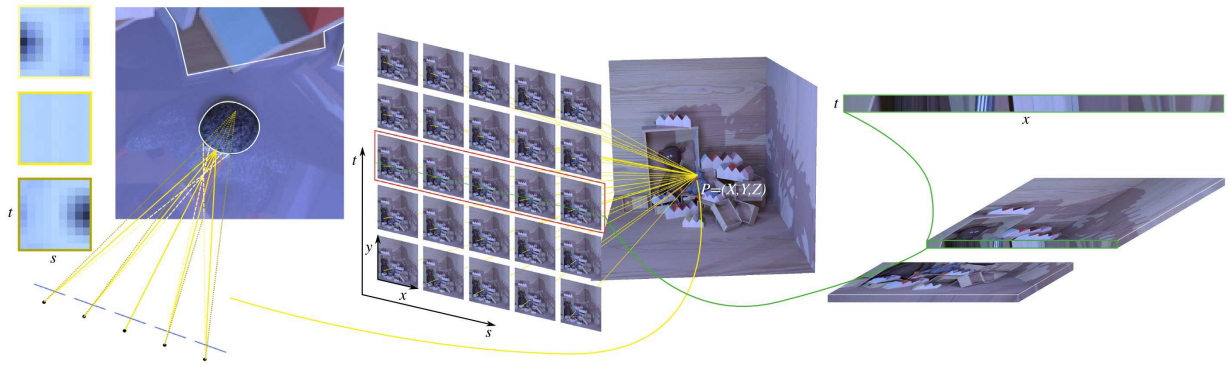

Figure 3. Different representations of the light field. In the most common form, the light field is given as a collection of subaperture views on a regular grid of pinhole cameras with parallel optical axes (center). If one takes viewpoints along a line (red), stacks them on top of each other and computes a cut through the stack (right), one obtains an epipolar plane image (green). Scene points are projected onto lines in the epipolar plane image. To obtain angular patches (left), one collects all the corresponding projections of a 3D point in all of the subaperture views. If the 3D point lies in front of an unoccluded Lambertian surface, the patch is constant (solid yellow). If the virtual 3D point does not lie on a surface, the different view show different colors. The pattern is mirrored horizontally and vertically when considering points at a certain distance in front of or behind the surface, respectively (dashed and dotted yellow rays).

between subaperture views, where d is the disparity of the scene point visible in the pixels related by the equation. Multi-view stereo typically works on the principle of patch comparison and finds the best correspondence among the images for a range of disparities. One of the subaperture views is sometimes special, called the reference or center view, as it is frequently in the center of the sampling range within Π. By convention, we assign it the focal coordinate $\boldsymbol{c} = \mathbf{0}$. Most algorithms we discuss later on compute disparity only on the reference view.

Epipolar plane images. Besides subaperture views, it is common to consider other 2D slices through the 4D ray space. If we fix a pair (t, y) or (s, x), we obtain horizontal or vertical *epipolar plane images (EPIs)* [3], respectively, see figure 3. According to (2), 3D points project onto lines on the epipolar plane images, whose slope is related to depth. This leads to their characteristic structure, which for Lambertian scenes seems to consist of patterns of overlapping lines. Several methods thus turn the problem of estimating depth into estimating the slope of these patterns. An additional advantage of the rich structure of the EPIs is that it gives an elegant way to analyze more complex scenes. For example, one can simultaneously estimate the slope of superimposed line patterns as they occur in semitransparent or partially reflective regions [25, 12].

Surface cameras (SCams). The angular patch or surface camera (SCam) $A_{\boldsymbol{p},d}$ for a pixel $\boldsymbol{p}$ in the reference view and disparity d is a function of focal point $\boldsymbol{c}$, and samples radiance for all corresponding projections of a scene point at the respective depth [28]. From (2),

$$A_{\boldsymbol{p},d}(\boldsymbol{c}) = L(\boldsymbol{p} - d\boldsymbol{c}, \boldsymbol{c}). \tag{4}$$

See figure 3 for an illustration. Again, for a Lambertian surface, the angular patch will have constant radiance for an unoccluded point sampled at the correct depth. Thus, a cost function for depth reconstruction can e.g. be built based on minimizing angular patch variance [5].

Angular patches can be leveraged to analyze occlusions. If a scene point is visible only in a subset of cameras, only a subset of the patch will have low variance. This observation can help to determine both, the cameras which see a point and the correct disparity [4]. Another interesting insight is that the separating line between occluded and non-occluded cameras in the angular patch has the same orientation as the image edge in the subaperture views. This has also been leveraged for sophisticated occlusion analysis [23].

We briefly remark on the relation to the previous representations. First, a line on an EPI with disparity d corresponds to the sampling of an angular patch along a line of viewpoints at disparity d. Hence, EPI-based methods use a subset of the angular patch. Second, a multi-view stereo method based on (3) which uses only a single pixel in each view per disparity constructs the cost solely on the angular patch, and is thus equivalent to a SCam-based method. Typically, however, they use spatial patches in the views, i.e. also aggregate more information to increase robustness.

Focal stack. A useful feature of the light field structure is that it becomes possible to construct a refocused image I_Z as captured by a virtual camera focused at a specific depth Z. For this, one needs to sample the aperture in the focus plane over all rays which emanate from a point at this specific depth,

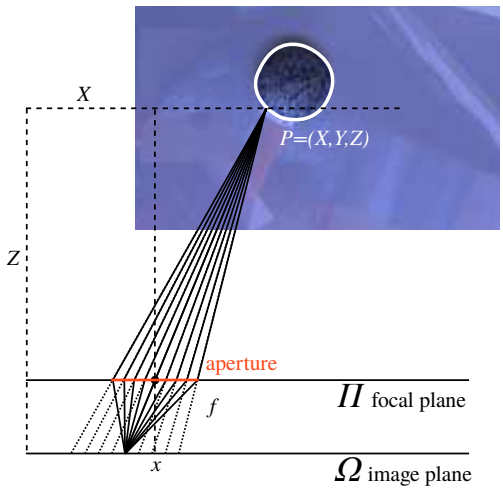

Figure 4. To construct a refocused image at pixel $\boldsymbol{p}$ in the reference view, with camera focused at depth Z, one has to sample over all rays in the subaperture views which correspond to $\boldsymbol{p}$. The solid lines correspond to the rays in the *refocused* light field, while the dotted rays correspond to the actual rays in a metric light field.

$$\begin{aligned} I_Z(\boldsymbol{p}) &= \int_\Pi w(\boldsymbol{c})\, L\left(\boldsymbol{p} - \frac{f}{Z}\boldsymbol{c},\, \boldsymbol{c}\right) \mathrm{d}\boldsymbol{c} \\ &= \int_\Pi w(\boldsymbol{c}) A_{\boldsymbol{p},d}(\boldsymbol{c})\, \mathrm{d}\boldsymbol{c}, \end{aligned} \tag{5}$$

where w is an aperture filter. See figure 4 for an illustration. In effect, refocusing thus means integrating over the angular patches at that depth. By varying Z, one generates a focus stack over the center view, where the individual views are focused to different depth layers. A focus measure can then be used to compute classical depth from focus [15]. In light field algorithms, this is typically used to augment other cost functions [22].

Angular patch and focal stack symmetry. An interesting aspect of angular patches is their behavior if sampled away from the correct surface. Under the assumption that the surface is fronto-parallel at correct disparity d, the angular patch at $d-\delta$ will be a copy of the angular patch at $d+\delta$, but mirrored both vertically and horizontally, see figure 3. As focal stack values are computed by integrating over the angular patches as in (5), this means that in disparity units, the focal stack is also symmetric around the correct disparity value [14]. This idea was extended to partial focal stack symmetry with occlusion awareness in [20].

3. Challenge participants

Before we commence with the taxonomy of current state-of-the-art light field algorithms, we will first give a short description of each of the algorithms considered in this paper. The algorithms are divided into three groups: baseline, challenge, and other algorithms. The baseline algorithms are marked with an asterisk and were included in the initial evaluation of [7]. Two algorithms were not submitted for the challenge but to the benchmark and pose interesting concepts/insights for this survey and thus were included. They are marked with '.

*`EPI1` [12] analyzes the orientation of patterns in EPIs. They build a dictionary with atoms of fixed disparity, and use sparse coding on patches of the EPI to find those dictionary elements which best describe the patch. The key idea is that only those patches with the correct orientation will be selected by sparse coding. Thus, from the coding coefficients, an initial depth map can be constructed which is later refined using anisotropic TGV smoothing. This approach also allows the reconstruction of multi-layered depth maps, e.g. in regions with transparencies.

*`EPI2` [27, 25] analyze the orientations of patterns on the EPIs. They take an analytic approach and compute the orientation of epipolar lines using the structure tensor. Different approaches can be taken to construct a depth map from the structure tensor, reaching from occlusion consistent depth labeling [24] to faster regularization-based approaches [27]. By applying the second order structure tensor, this approach also allows the reconstruction of multi-layered depth maps [25].

*`LF` [9] uses the sum of absolute differences as well as the sum of gradient differences in small rectangular patches to build a cost volume. Each slice of the cost volume is regularized individually using the center view as guidance. Afterwards, a multilabel optimization using graph cuts is performed using sparse SIFT features as guidance. The discrete disparity map is refined in a final step by fitting quadratic functions.

*`LF_OCC` [23] builds upon the method of Tao et al. [21] and refines occlusion boundaries. Candidates for refinements are areas around edges in the center view. For these regions the angular patches are divided according to the edge orientation in occluded and non-occluded regions. Afterwards, an MRF is regularized with binary costs that include an occlusion prediction.

*`MV` is a lab implementation of an occlusion aware multi-view algorithm. It uses the crosshair of views. Four different cost volumes are built comparing the center view pixels to pixels in views to the left, right, top, and bottom, respectively. As a cost function the average of the L1-norm is used. Afterwards, the cost volumes are combined by summing the minimum costs in horizontal and vertical direction. Finally, an initial disparity map is generated using the winner-takes-all strategy, which is regularized using TGV-L^1 denoising.

'`OFSY_330DNR` [20] builds a cost volume based on focal stack symmetry. By selecting views only along certain directions for constructing the focal stacks, the cost computation is made robust to occlusion. From the cost volume,

they compute an initial disparity map using sub-label accurate global optimization. This disparity map is refined in a final step to not only smoothen the disparities but also the normals.

OMG_occ [1] models the case of multiple occluders and derives a relationship between angular and spatial patches of the light field for this case. Selecting the non-occluded views given an initial disparity estimation and a k-means clustering yields a cleaner cost volume which is later regularized using an MRF with weights of the pairwise term adjusted to occlusion boundaries.

PS_RF [10] uses four different cost volumes based upon the sum of absolute differences, the sum of gradient differences, zero-mean normalized cross correlation, and the census transform. All are implemented using the phase shift theorem for warping. Afterwards, two cascade random forests are built. One for classification, i.e. choosing the important combinations of costs and one for regression to infer a disparity value with sub-pixel precision.

RM3DE[16] computes the L2 differences between the center view and the outer views in small 1D windows. To account for occlusions they take the minimum residual between the costs for a view $\boldsymbol{c}$ and the opposite view $-\boldsymbol{c}$. Additionally, they only consider the window in one of the directions depending on the baseline between the views. This dataterm is implemented in a course-to-fine scheme to account for regions with little to no texture.

SC_GC [19] computes the average residual of the 50% views of each SCam patch with the lowest error and combines these costs patchwise into a cost volume. This cost volume is optimized in an edge and occlusion aware manner. Afterwards local plane fitting refines the estimate.

SPO [29] operates on epipolar plane images and estimates the orientation of the epipolar lines. Two regions - slightly to the left and right of the line in question - are defined and histograms are computed for both. As weights for the contribution of each pixel the derivative of Gaussian is used. Comparing these histograms yields a cost function over different discrete disparities. The costs volumes for horizontal and vertical direction are combined according to a confidence measure. Afterwards, the cost volume is regularized for each depth label individually and a disparity map generated by the winner-takes-all strategy. No sub label refinement is performed.

ZCTV and OBER [2] are two alternative post-processing steps to a sparse EPI based line fit algorithm. Edges in the EPI are detected with subpixel accuracy using the zero crossings of the second derivative in the horizontal direction. Lines are then constructed using a RANSAC-like procedure. ZCTV uses a total variation approach to perform inpainting and denoising from the sparse depth map. The total variation parameters are adapted by an edge map and the line fit variance. For each pixel, OBER iteratively minimizes a smoothness metric plus the variance along the corresponding line in the EPI, discounting occluded pixels. The smoothness term is based on a bilateral filter. 'OBER-cross uses the crosshair of views and combines the horizontal and vertical information into a joint disparity map.

4. Taxonomy of light field algorithms

The algorithms submitted to the challenge differ greatly in the representation of the light field they are based upon, as well as the optimization steps they take. The typical pipeline is to build a cost volume based on one or more representations, then perform global optimization to build an initial disparity map, and then perform further refinement steps.

Unfortunately, optimization and refinement steps seem too different to employ them for a useful grouping. We nevertheless give a brief overview on these. The most expressive classification comes from the light field representations the methods work on. However, a method might be based on several different representations and work with them in very different ways.

One additional aspect should be mentioned at that point. For light fields, the desired accuracy is far less than one pixel, thus there has been discussion on whether standard bilinear or bicubic interpolation is sufficient to obtain accurately shifted images. Some algorithms [10, 9] therefore follow [18] and make use of the phase shift theorem to perform shifting in the Fourier domain. Specific interpolation examples indeed show that the use of this method is superior to the other two. Due to the high variance between the different algorithms in terms of data terms and final optimization, however, we cannot really determine the influence of this factor on the final result, More specifically controlled experiments are required at this point, however, it seems reasonable to believe that if an approach is actually better in practice, then it will be indeed phase shifting. An overview of the different algorithm aspects is given in table 1.

4.1. Classification according to representation

Methods based on EPIs. Five different methods estimate disparity by analyzing orientation on EPIs. *EPI1 builds a disparity aware dictionary and uses sparse coding, *EPI2 estimates the slope of epipolar lines by using the structure tensor, SPO builds a cost volume for a discrete set of disparities by comparing regions left and right of epipolar lines, and finally the zero crossings based methods ZCTV, OBER and 'OBER-cross perform a low level feature search in each row of the EPI and match lines using a RANSAC scheme.

For *EPI1, the recommended patch size is 5×5. The dictionary elements model patches as constant in disparity. Thus, larger patches are less flexible to local disparity

algorithm	dataterm	views	occlusion aware dataterm	interpolation	cost volume	optimization	refinement
*EPI1	EPI	crosshair (5×5)	no	-	50	variational	variational TGV-L^1
*EPI2	EPI	crosshair	no	-	-	-	variational TGV-L^2
*LF	MultiView	full	no	phase shift	100	MRF	iterative refinement
*LF_OCC	SCam	full	yes	linear	200	MRF	occlusion aware MRF
*MV	MultiView	crosshair	yes	linear	100	variational	variational TGV-L^1
'OBER-cross	EPI	crosshair	no	linear	-	-	bilateral refinement
'OFSY_330DNR	Focus	crosshair	yes	linear	330	variational	variational normal regularization
OBER	EPI	horizontal	no	linear	-	-	bilateral refinement
OMG_occ	SCam	full	yes	linear	100	MRF	occlusion aware MRF
PS_RF	MultiView	full	no	phase shift	151	random forest	weighted median
RM3DE	MultiView	crosshair + diagonals	yes	-	-	-	weighted median
SC_GC	SCam	full	yes	linear	256	MRF	second order smoothness in MRF
SPO	EPI	crosshair	no	linear	256	winner takes all	guided filtering (on cost volume)
ZCTV	EPI	horizontal	no	-	-	-	second order TV

Table 1. This table gives a simplified overview of the examined algorithms and labels them according to the taxonomy presented.

changes, produce smoother depth maps, but decrease accuracy at occlusion boundaries. *EPI2 computes EPI gradients and disparity needs to be small enough to allow this in a robust way. Thus, disparity range is limited to around -1.5 to 1.5px, but can be increased by pre-shearing at the cost of runtime [6].

In contrast, the other two methods are not inherently restricted with respect to disparity range and the views close to the center view. Thus, they can utilize the larger baseline of the outer views to estimate depth more precisely. Zero crossing based approaches have the advantage that the accuracy is very high due to subpixel accurate features and matching. On the other hand, SPO seems to be very robust as not only individual pixels/features/points are compared, but small weighted regions. Maximizing the histogram distance between the two regions appears to be robust in the presence of occlusions as well. A problem for all EPI based approaches is how to integrate the estimates on horizontal and vertical EPIs. Usually, a weighting between the two estimates is performed based upon a residual or statistics of the estimated cost volume/distribution function.

Methods based on angular patches. Methods solely based on angular patches are *LF_OCC, OMG_occ and SC_GC. The first two explicitly estimate the distribution of non-occluded pixels in the angular patch by analyzing orientations of edges in the angular as well as spatial patches. While *LF_OCC only models the case of single occlusions, OMG_occ models the case where multiple occluders are present and thus more than one occlusion edge can exist in an angular patch. SC_GC skips the explicit segmentation of the patches into occluded or non-occluded regions by just considering the 50% pixels that are closest in radiance to the center view pixel. In general, approaches based on angular patches are not significantly limited by the number of views or disparity range.

Methods based on the focal stack. Although more exist in the literature, e.g. [22], the only method based on the focal stack evaluated so far is 'OFSY_330DNR, which exploits focal stack symmetry. They build an occlusion-aware cost volume by computing the minimum of the cost function over partial focal stacks integrated along the directions of a crosshair within the angular patch. The reasoning - which is also used in multi-view stereo methods - is that occlusion occurs only in one direction, so they can always compare the parts of the stack which are occlusion free. However, this does not account for multiple or very small occluders. into account.

Methods based on multi-view stereo (MVS). The most straight-forward method is a baseline multi-view-stereo algorithm *MV with a point-wise L^1-dataterm based on (3). In addition, similar to the idea above, the minimum residual of the view $\boldsymbol{c}$ and the view $-\boldsymbol{c}$ on the opposite side of the center view is taken to reduce occlusion effects. This simple approach builds the cost volume based only on angular patches, so in principle, it would fit into the category above as well. Typically, however, MVS algorithms compare not only point- but patch-wise. Three algorithms fall into this category, *LF, PS_RF and RM3DE.

The first one builds a cost volume based upon a combination of the sum of absolute differences and the sum of gradient differences for small rectangular patches. The cost volume is later smoothed using weighted median filtering to preserve occlusion boundaries.

PS_RF additionally uses the zero-mean normalized cross correlation and the census transform as data terms. They build individual cost volumes for all four of the data terms, and train random forests to choose an optimal weighting of dataterms as well as infer disparity.

RM3DE in contrast considers the L^2-differences of small 1D patches. The orientation of the 1D patches corresponds to the orientation of the baseline between the center view and the other views. This improves performance at occlusion boundaries.

In general, patch based approaches make the depth estimation more robust to noise compared to approaches based only on the angular patch, as also the spatial neighborhood in a view is considered. However, if not modeled explicitly, they tend to run into more problems in the presence of occlusions.

4.2. Initial depth map extraction

Methods based on cost volume computation at this point have to extract an initial depth map from the cost volume. Typically, these are either MRF/graph cut or variational approaches, which employ additional regularization, often adapted to center view edges and other factors. Many algorithms first perform edge-aware regularization of the respective cost volume slices.

In general, the methodology in this step varies greatly, see section 3, and it is hard to determine a useful classification based on the applied technique. We therefore only briefly mention it in the overview table 1.

4.3. Refinement

The last step for generating a dense depth map might have even greater variance than the one before, and there is no eminent common strategy. The goal is often just to regularize the initial result further, and tends to be quite heuristic in nature. It is difficult to evaluate which of these heuristics are really a universal improvement.

Some techniques employ local filtering approaches for post processing, i.e. weighted median (`RM3DE`, `PS_RF`), or bilateral filters (`OBER`, `'OBER-cross`). A second type consists of global regularization models for the disparity map, with different kinds of image adaptive or occlusion aware weighting. `*LF_OCC` and `OMG_occ` build an MRF, while `*EPI1`, `*EPI2`, `*MV`, `ZCTV`, and `'OFSY_330DNR` use variational models with different dataterms and different regularizers to obtain a regularized depth map.

We would again like to point out some unique approaches. `SC_GC` performs local plane fitting. Although this obviously leads to problems with curved surfaces, it also has a very nice property, as it can regularize different layers of the scene even across holes. Another interesting special case is `'OFSY_330DNR`, which performs regularization directly on the normals and not on the depth labels, yielding very smooth surfaces.

As all of these approaches are fairly different in detail, the influence of this final step is also hard to predict and requires targeted additional research.

5. Evaluation methodology

The evaluation methodology is largely based on the metrics, scenes, and concepts as introduced in the original light field benchmark paper [7]. In addition, we add the concepts of the *PerPixBest* and *PerPixMedian* algorithms as well as additional high accuracy and surface reconstruction metrics.

5.1. Challenge details

Challenge participants had to submit estimated disparity maps and runtimes on the four stratified and eight photorealistic scenes as depicted in figure 1. As input, $9 \times 9 \times 512 \times 512$ RGB input images and the disparity range are provided for each scene. The outer views are shifted towards the center view, thus, the approximate disparity range of the scenes is $[-2,\ 2]px$ (see supplemental material of [7] for details).

For algorithm validation and parameter tuning, ground truth depth and disparity maps of the center views are available for the stratified and training scenes as well as for 16 additional scenes. On all scenes, a boundary region of 15 pixels is ignored during evaluation of the error metrics. While this allows for a certain sloppiness in the handling of image boundaries, we decided for it as the algorithms cannot make use of the complete range of views at the boundaries due to pixels mapped into non-captured regions. For the submission, a single choice of parameters is required for all scenes. While automated adaptations to local scene properties are accepted, scene-specific parameter settings are not.

Please note that all runtimes have been reported by the authors, and are not the result of running the algorithm on a standardized system. Due to different hardware configurations and implementation details, runtime should only be used as a rough indicator on whether an algorithm is "rather fast" or "rather slow".

5.2. Additional algorithms

We evaluate the 14 algorithms as described in section 3. In addition, we use two artificial algorithms in our evaluation: *PerPixBest* and *PerPixMedian*. For the *PerPixBest* algorithm, we take the disparity estimate with the lowest absolute difference to the ground truth at each individual pixel, based on the estimates of all 14 algorithms. This algorithm is used as an approximate "upper limit" of algorithm performance. For the *PerPixMedian*, we take the disparity estimate of the algorithm with the median absolute error for each pixel. This algorithm is used as an approximate "average" algorithm performance.

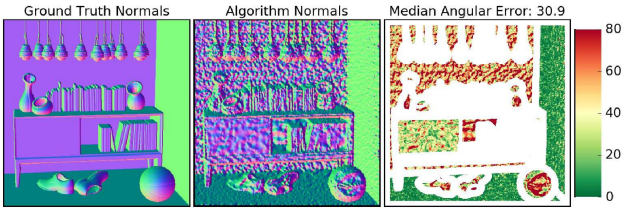

Figure 5. Surface reconstruction performance is quantified with the median angular error (*MAE*) between the ground truth normals and algorithm normals. For the evaluation, it is computed separately on planar and non-planar continuous surfaces.

5.3. Additional metrics

Based on observations on the submitted algorithms, we add two types of additional metrics.

High accuracy metrics Figure 11 shows the percentage of correct pixels with increasingly higher error thresholds. The relative rankings between algorithms change significantly for different thresholds, e.g. between 0.01 and 0.03. Moreover, the relative performance differences between algorithms are very small for high error thresholds though quite significant for lower thresholds. Figure 12 illustrates that this is especially true for easier scenes such as *Cotton*. We therefore add *BadPix(0.01)* and *BadPix(0.03)* as additional *BadPix* scores.

Furthermore, we add *Q25*, representing the accuracy at the 25th percentile of the disparity estimates on a given scene. Thus, it measures the maximum error on the best 25% of pixels for each algorithm (see third row in figure 12). In effect, it provides an idea of the "best case accuracy" of a given algorithm. In line with the *MSE*, the absolute disparity difference is multiplied by 100.

Surface reconstruction metrics The bumpiness score as defined in [7] quantifies local smoothness, but it does not account for situations like a smoothly estimated plane which is rotated with respect to the ground truth. However, accurate surface orientations play an important role when depth estimation is incorporated into more sophisticated algorithms which also try to estimate shading/illumination or material properties. Therefore, we add the median angular error (*MAE*) of the depth map surface normals as a generalization of the *local misorientation* metric as proposed by Honauer *et al.* [8]. An example of ground truth and algorithm surface normals as well as the per-pixel angular error is shown in figure 5. Similar to the bumpiness metrics, we compute the *MAE* separately on planar and non-planar continuous surfaces.

6. Evaluation of algorithms

In this section, we thoroughly assess and compare the depth estimation performance of the 14 algorithms. We also hypothesize and gain insights on which specific approaches and algorithm aspects lead to good performance at planar surfaces, occlusion areas etc. However, without reference implementations, these insights cannot be thoroughly validated and should therefore be treated with caution.

6.1. Performance overview

The radar charts on figure 6 provide a notion of the relative performance of the 14 algorithms for each metric, depicting the median score per metric across (a) all stratified scenes and (b) all photorealistic scenes. Lower scores towards the center represent better performance.

According to these charts, there is no single best algorithm which outperforms all other algorithms. Instead, some algorithms have very specific strengths which also come with certain drawbacks. For example, on the photorealistic scenes `'OBER-cross` is best on high accuracy metrics but struggles with fine thinning (figure 6b). `'OFSY_330DNR` is best on the surface metrics but not on fine structures. `SPO` features the best tradeoff on discontinuities and fine structures but not on surface metrics. By contrast, `RM3DE` is rarely among the top three algorithms on the photorealistic scenes but it shows a good overall performance and - in contrast to most other algorithms - no explicitly strong weakness.

Even though there is no clear winner algorithm, some algorithms do outperform other algorithms in most aspects. On the stratified scenes, `RM3DE` outperforms `OMG_occ` on all axes (see figure 6a). On the photorealistic scenes, `RM3DE` outperforms `*EPI1` and `*LF_OCC` on all aspects. `'OBER-cross` outperforms `*LF`, `*LF_OCC`, `OMG_occ`, and `PS_RF` on all aspects except for fine thinning. Similarly, `OBER` outperforms `*EPI1`, `*LF`, `*MV`, `*LF_OCC`, `OMG_occ`, and `PS_RF` on all aspects except for fine thinning and `*EPI2` on all aspects except for runtime. `'OFSY_330DNR` beats `*EPI2` everywhere except for runtime and `*LF` everywhere except for fine thinning.

Figures 7 and 8 explicitly depict disparity maps, ground truth error maps, and median error maps for all algorithms on the stratified and training scenes. For the median error map, the median of the absolute disparity differences $|Algo - GT|$ of all algorithms is computed for each pixel. For the visualization, $|Algo - GT|$ of the respective algorithm is then subtracted from the "median error". This error map gives a notion on which image regions algorithms perform above or below average algorithm performance.

As depicted on the training scenes in figure 8, most algorithms tend to be either good at discontinuities or continuous surfaces. On *Cotton*, *Dino*, and *Boxes*, `OBER`, `PS_RF`, `RM3DE`, and `SPO` are very good at depth discontinuities but

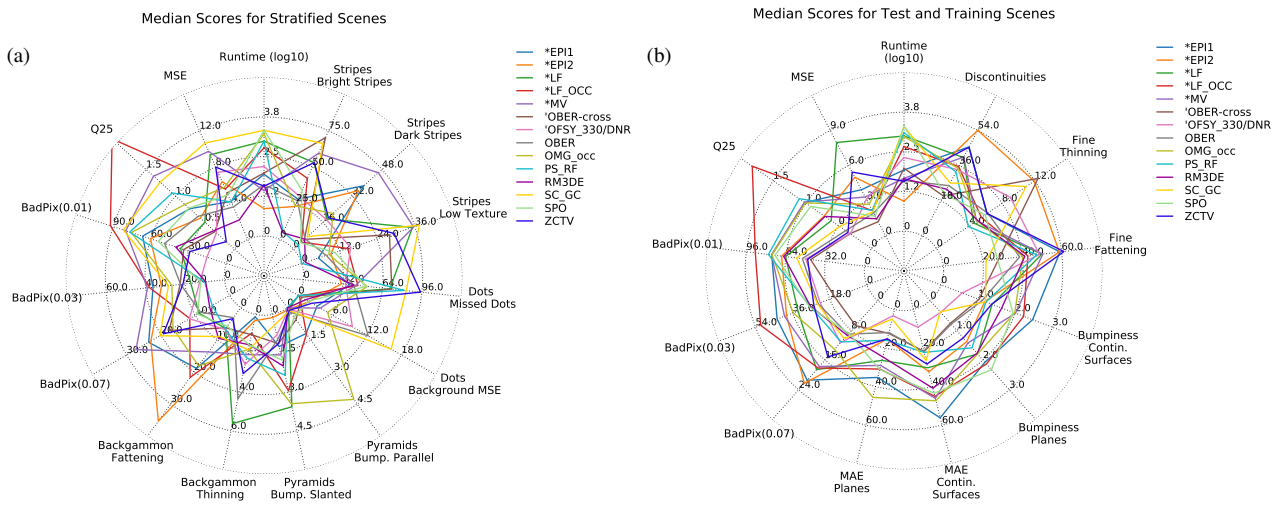

Figure 6. The radar chart illustrates how the scores of all evaluated algorithms compare against each other, depicting one metric per axis with the median score per metric across (a) all stratified scenes and (b) all photorealistic scenes. Lower scores towards the center are better. *We recommend using the interactive visualization on the benchmark website for a less cluttered version of the radar charts.* There is no clear winner algorithm which outperforms all other algorithms. `'OBER-cross` is best on high accuracy metrics but not on fine thinning, `'OFSY_330DNR` is best on the surface metrics but not on fine structures, and `SPO` features the best tradeoff on discontinuities and fine structures but not on surface metrics.

below average at reconstructing e.g. the surface of the *Cotton* statue. By contrast, `'OFSY_330DNR` and `ZCTV` perform well on continuous surfaces but below average at discontinuity regions. `'OBER-cross` performs well on both regions but struggles with the planar background on *Cotton*.

Performance on the stratified scenes is shown in figure 7. On *Backgammon*, `'OBER-cross`, `'OFSY_330DNR`, `OBER`, `SPO`, and `ZCTV` excel with low fattening artefacts between the peaks. On *Stripes*, `*EPI1`, `*EPI2`, `PS_RF`, and `RM3DE` prove very robust towards the low-texture areas on the lower part of the scene. On *Dots*, most algorithms tend to be either good at reconstructing the dots or the background. `*LF_OCC` and `SPO` manage to get a good tradeoff performance in reconstructing both.

6.2. Individual performance analysis

According to the radar chart on figure 6b, the baseline algorithm `*EPI1` features an average overall performance. It does not score very well on the surface metrics, which may also cause the rather high scores on the per-pixel metrics. `*EPI1` uses only 50 depth labels in the disparity aware dictionary and is limited in the maximum disparity range it can handle. This may lead to high error rates at background planes such as on *Herbs*. Additionally, the TGV-L^1 smoothing used in this implementation does not perform as well as an L^2 smoothing when it comes to the bumpiness metrics as e.g. used by `*EPI2`. This is reflected by the noisier normal map of `*EPI1` as compared to `*EPI2` (see figure 9). As no edge awareness is built into the regularizer, the occlusion performance is subpar.

`*EPI2` is very good at smoothly estimating the surfaces on *Pyramids* and good at estimating the surfaces on the photorealistic scenes in general. By contrast, it struggles at discontinuities and fine structures. The TGV-L^2 smoothing leads to good bumpiness scores, while the missing edge awareness leads to heavy fattening.

`*LF` scores average on most metrics and well for fine thinning on the photorealistic scenes. It produces the highest overall *MSE* error. This is most likely routed in the way regularization is performed. The algorithm uses 100 depth labels and tries to fit quadratic functions as a refinement step. This works fine for surfaces roughly perpendicular to the optical axis, but produces heavy staircasing in regions with slanted surfaces (see *Pyramids* on figure 7 and walls in *Dino* on figure 8).

`*LF_OCC` features an average overall performance. It does not score very well on the surface and high accuracy metrics but above average on the *MSE*. The below-average scores for surface and high accuracy metrics may be due to the use of a discrete set of disparity labels and could probably be improved by applying some kind of refinement step.

`OBER` scores second to fourth on almost all metrics with intermediate fine thinning scores and no strong weakness. `'OBER-cross` outperforms `OBER` on most aspects, except for much stronger fine structure thinning and a longer runtime. It scores best at discontinuities and the high accuracy metrics. On the stratified scenes, `'OBER-cross` has difficulties on *Stripes* and *Dots*. The strength of the two

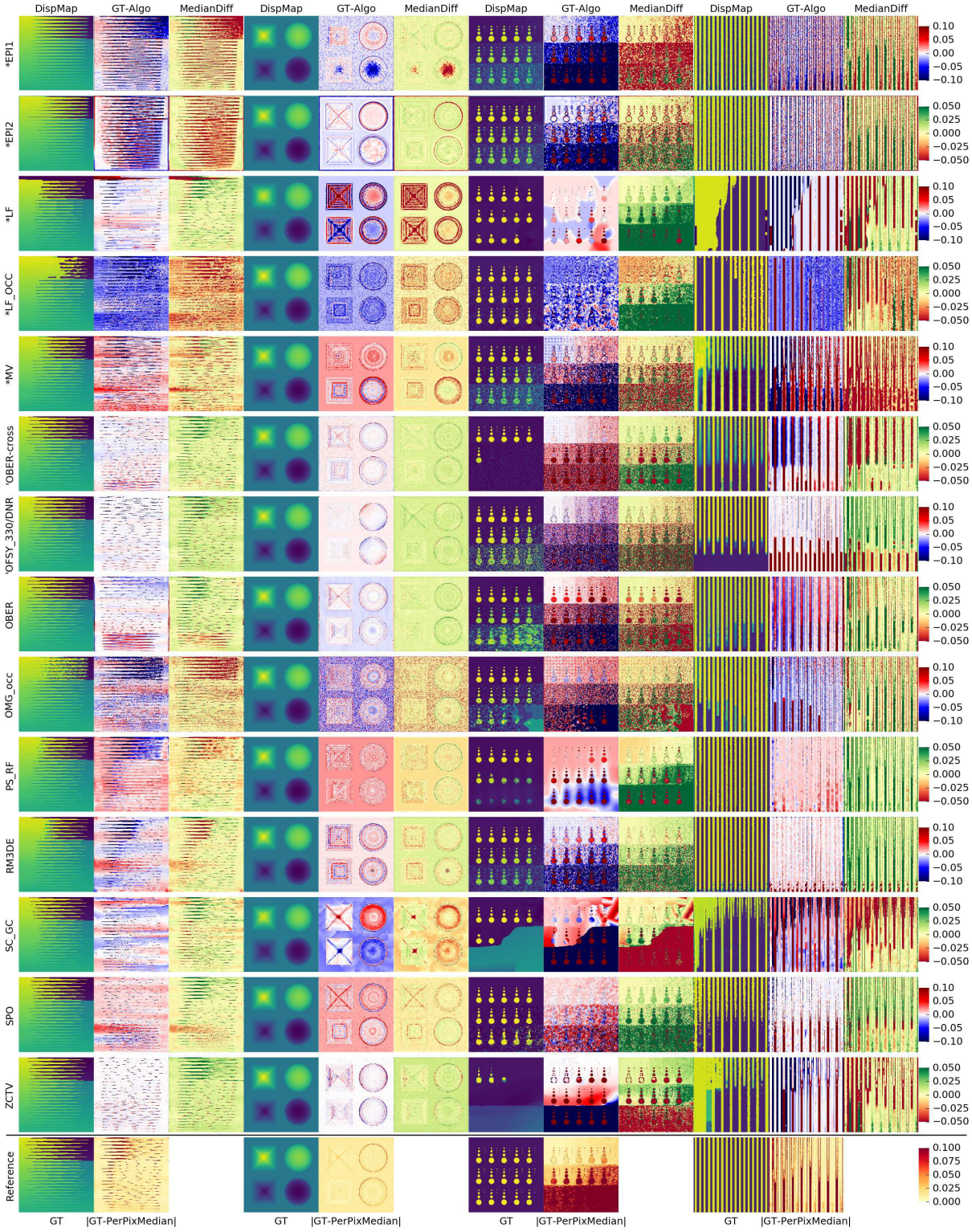

Figure 7. The first column for each of the stratified scenes illustrates the disparity maps of the 14 algorithms. The second column depicts the disparity difference to the ground truth. Estimates are highly accurate at white areas, too close at blue areas and too far at red areas. Please note that the visualization is scaled to $[-0.1, 0.1]$. The third column illustrates how algorithms perform relative to the median algorithm performance. Yellow represents average, green above-average and red below-average performance.

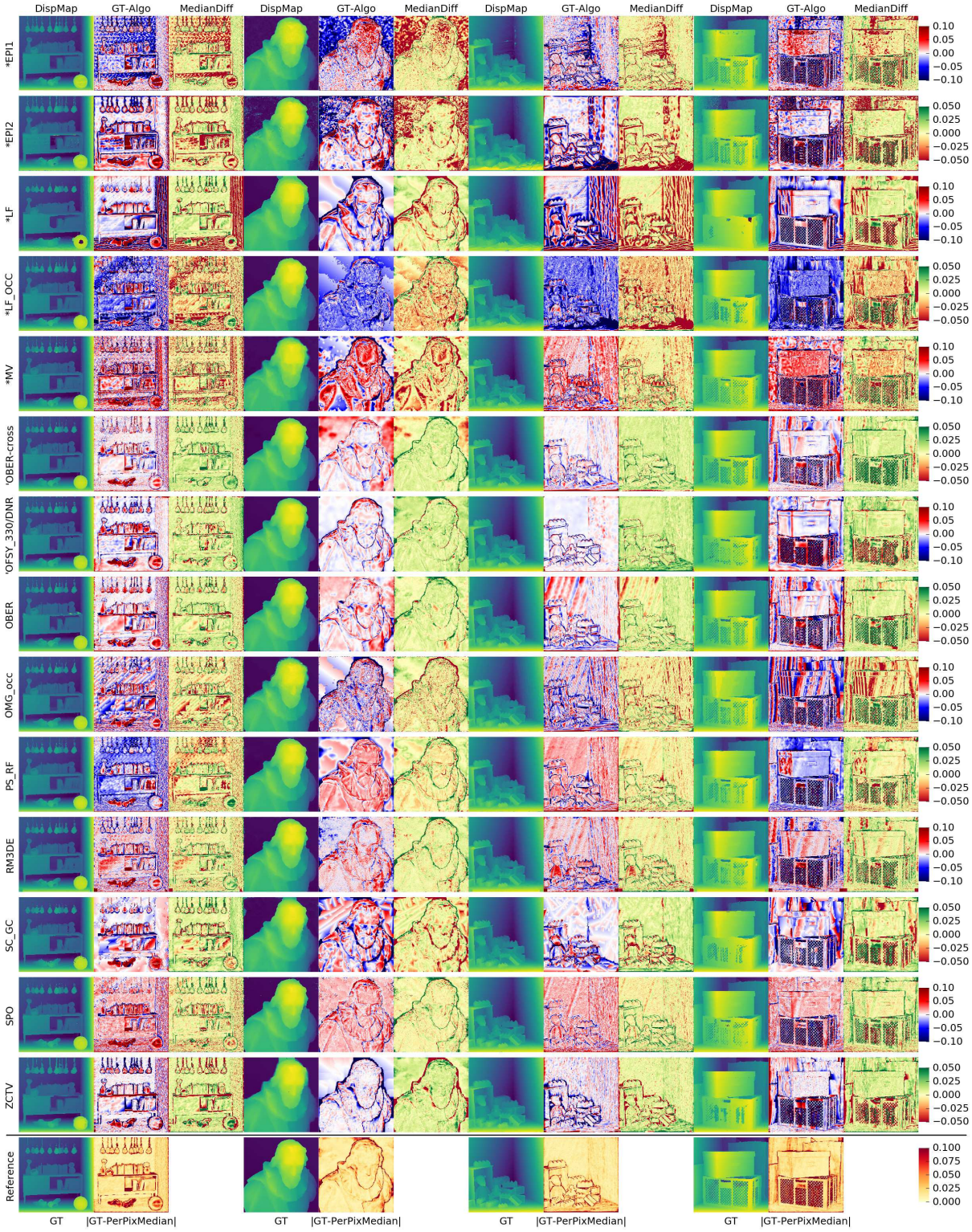

Figure 8. Please see figure 7 for an explanation of the visualization. The *Dino* scene illustrates nicely how algorithms perform differently on discontinuity regions and planar surfaces. For example, `PS_RF`, `RM3DE`, and `SPO` tend to be above-average at occlusion regions but below average on the background planes. By contrast, `'OFSY_330DNR` and `ZCTV` have the opposite strengths and weaknesses. `'OBER-cross` performs well on both areas.

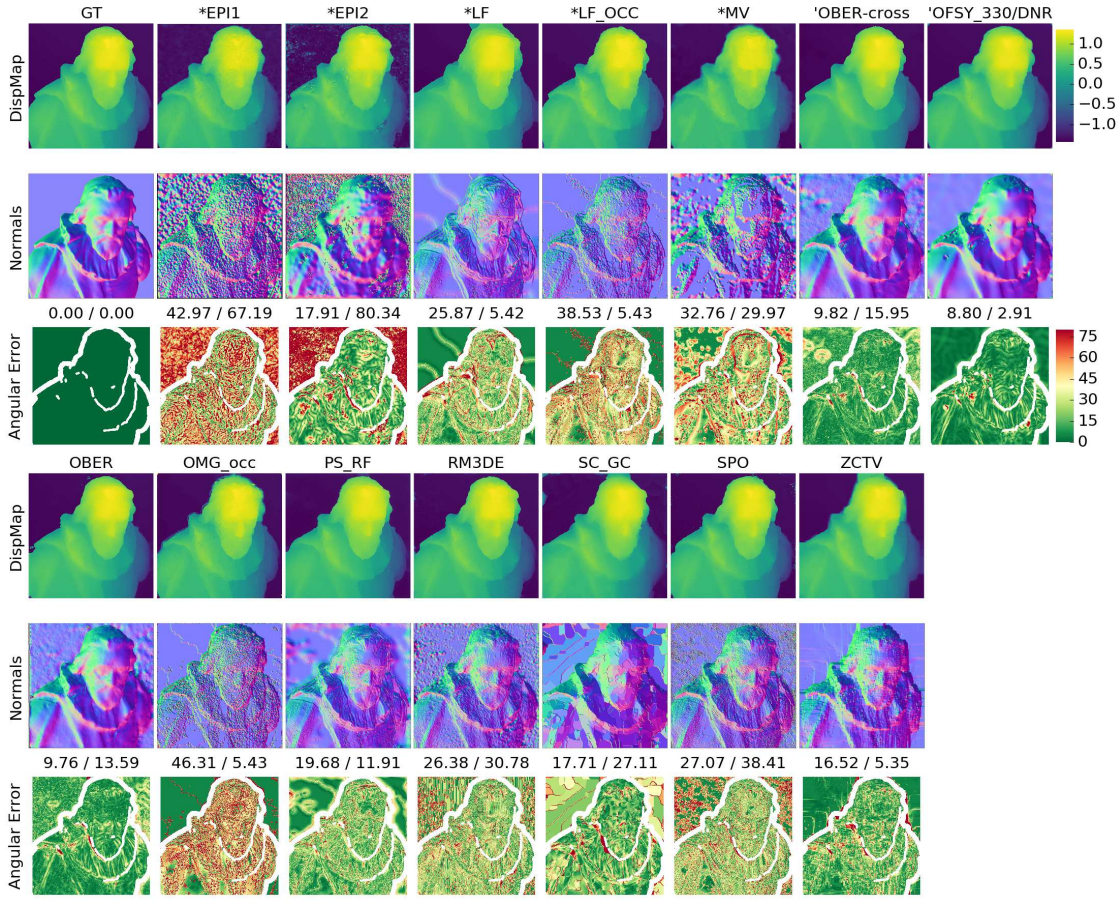

Figure 9. The third and sixth row depict the angular error between the ground truth normals and algorithm normals. Scores are computed separately for non-planar and planar surfaces. 'OFSY_330DNR performs best at reconstructing both kinds of surfaces, featuring smooth and accurate normals. For SC_GC the locally fitted planes are clearly visible on the normal map.

OBER methods may be due to the high accuracy of the initial depth estimate (zero crossings) which is then refined in an occlusion aware way resulting in very good occlusion boundaries (see *Bicycle* disparity maps on figure 10) and smooth surfaces (see *Cotton* normal maps on figure 9). We would speculate that the key here is that for the final refinement step the variance along the corresponding epipolar lines is taken into account, thus linking the final refinement step to the input data.

'OFSY_330DNR performs very well on reconstructing planar and non-planar surfaces (see normal maps on figure 9) but it features mediocre to poor performance at fine structures. On figure 6b, 'OFSY_330DNR scores well on the high accuracy metrics *Q25*, *BadPix(0.01)*, and *BadPix(0.03)*. The good surface reconstruction and good overall accuracy may be due to the data term, the large amount of labels, the sub-label accurate cost volume optimization, or the normal smoothing. Occlusion performance is only mediocre (see *Bicycle* disparity map on figure 10), despite explicit occlusion handling for the data term as well as the anisotropic/binary weighting of the regularizer.

OMG_occ scores well at fine thinning, average on discontinuities and fattening, and below average on the surface metrics. Unfortunately, it is hard to speculate about OMG_occ as we have limited information about this algorithm. The specific selection of unoccluded views from SCams is theoretically sound but seems to yield only average results at discontinuities. This might be due to the optimization and refinement of the cost volume. OMG_occ struggles at the surface metrics as it does not perform any sublabel refinement, thus, heavy staircasing is present (see e.g. *Sideboard* and *Boxes* in figure 8).

PS_RF scores above average on most aspects except for the high accuracy metrics and fine fattening. PS_RF is remarkably robust on *Stripes*. The offset in the *Pyramids* background plane and the below-average high accuracy scores may be caused by the comparatively low number of depth labels.

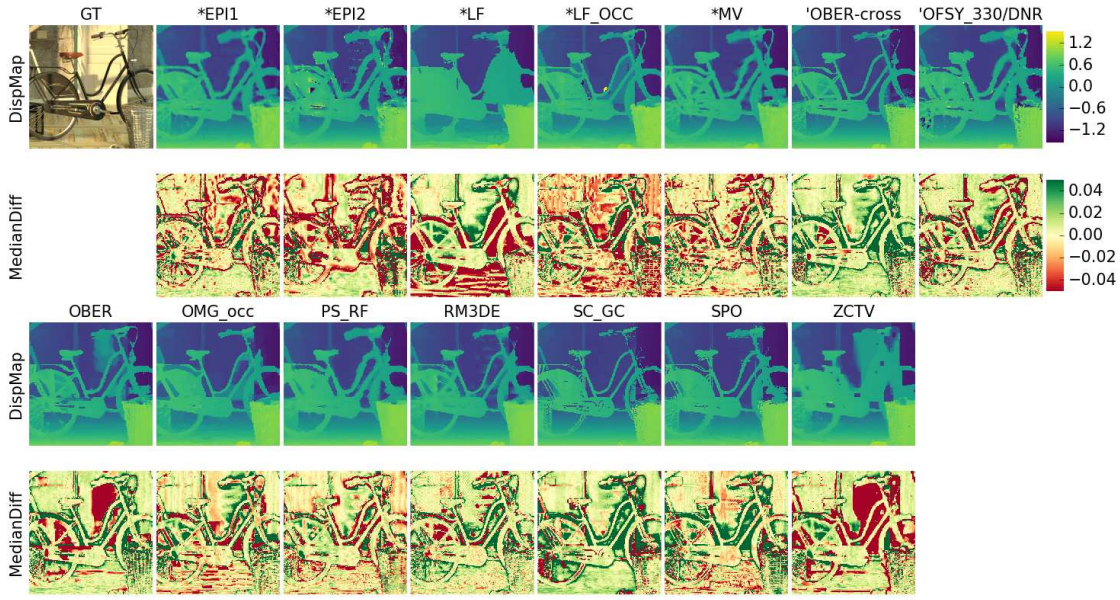

Figure 10. Most algorithms struggle with accurately reconstructing the fine details of the bicycle. Almost all algorithms suffer from "bleeding" of the foreground structure into the low texture background. SC_GC, 'OBER-cross, and SPO perform remarkably well on this difficult situation.

RM3DE scores well to average on all aspects of the photorealistic and stratified scenes. It is among the best algorithms for the *MSE* and discontinuity scores. RM3DE suffers from fine structure fattening on *Backgammon* but is very robust on *Stripes* and *Dots*. The rather good tradeoff between background and reconstructed dots in the presence of noise may be due to the multi-resolution approach and the occlusion aware patch wise data term.

SC_GC shows a similarly strong but slightly inferior performance profile compared to 'OBER-cross (see figure 6b). It performs among the top five algorithms on all aspects except for fine thinning and runtime. It outperforms all algorithms on reconstructing planar surfaces except for the comparable performance of 'OFSY_330DNR. SC_GC performs particularly well at the background planes of *Dino* and *Sideboard* but has difficulties with strong discontinuities and some continuous, non-planar surfaces (see figure 8). The plane fitting allows the algorithm to regularize surfaces jointly which are partially occluded. This behavior can be observed in the *Bicycle* scene (see figure 10). SC_GC produces very crisp boundaries and an accurate estimation of the poorly textured door in the background where most algorithms suffer from heavy fattening.

SPO is the only algorithm performing very well at both, fine structure thinning and fattening as well as general discontinuities (see radar chart on figure 6b and the bicycle on figure 10). By contrast, it performs below average at surface reconstruction. SPO is rather robust on *Dots* and very good at *Backgammon* but it has difficulties at the low texture areas of *Stripes*. This behavior could be altered by changing the number of histogram bins used.

ZCTV performs very well at high accuracy metrics and well on reconstructing surfaces but it features heavy edge fattening (see *Cotton* and *Dino* on figure 8 and *Bicycle* on figure 10). The strong smoothing seems to cause problems on *Dots* and *Stripes*. The good performance on *Backgammon* is probably positively influenced by the fact that only views of the horizontal line are used.

6.3. High accuracy and surface reconstruction

In this section, we assess and compare the maximum accuracy achieved by the algorithms. Figure 11 shows the percentage of correct disparity estimates on the photorealistic scenes for increasing error thresholds.

One important observation from this figure is that the order of algorithms changes for different thresholds. For the joint percentage over all test and training scenes 'OBER-cross scores best from 0.01 onwards, while 'OFSY_330DNR scores slightly better for even smaller thresholds. One interpretation might be that 'OFSY_330DNR can handle simple, well textured surfaces very well, but runs into problems when occlusions or other complicated surfaces are introduced, while 'OBER-cross can handle these regions with higher accuracy.

The curve of ZCTV has almost the same amount of correctly estimated pixels for the 0.03 and the 0.07 threshold, i.e. it already "saturates" at 0.03px accuracy. Apparently, ZCTV can estimate depth with high accuracy, but is prone

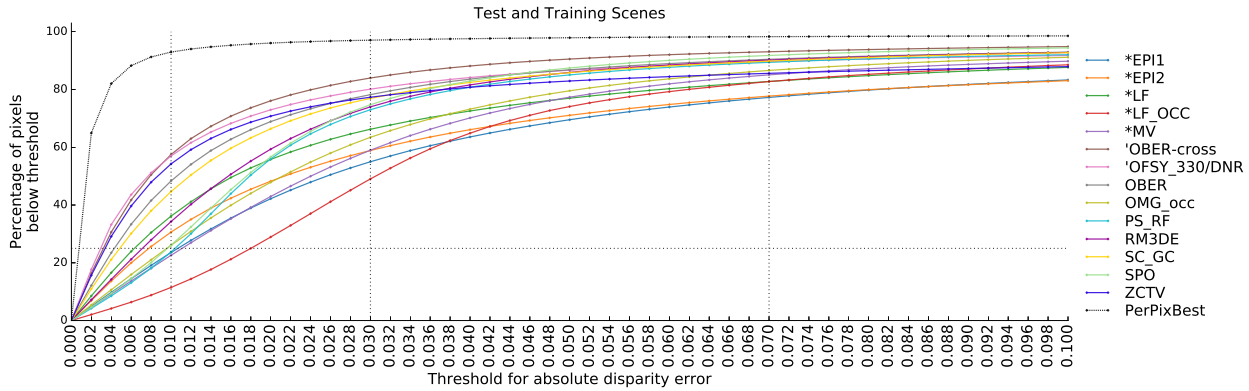

Figure 11. For each algorithm, the percentage of correct disparity estimates on the photorealistic scenes is plotted for the increasing error thresholds on the x-axis. The *PerPixBest* algorithm is shown as an approximate upper bound of algorithm performance. Relative algorithm rankings change significantly for different thresholds: `ZCTV` performs well at strict thresholds but only moderately for bigger thresholds. By contrast, `RM3DE` is among the top methods at a threshold of 0.07 but only moderate for very strict thresholds.

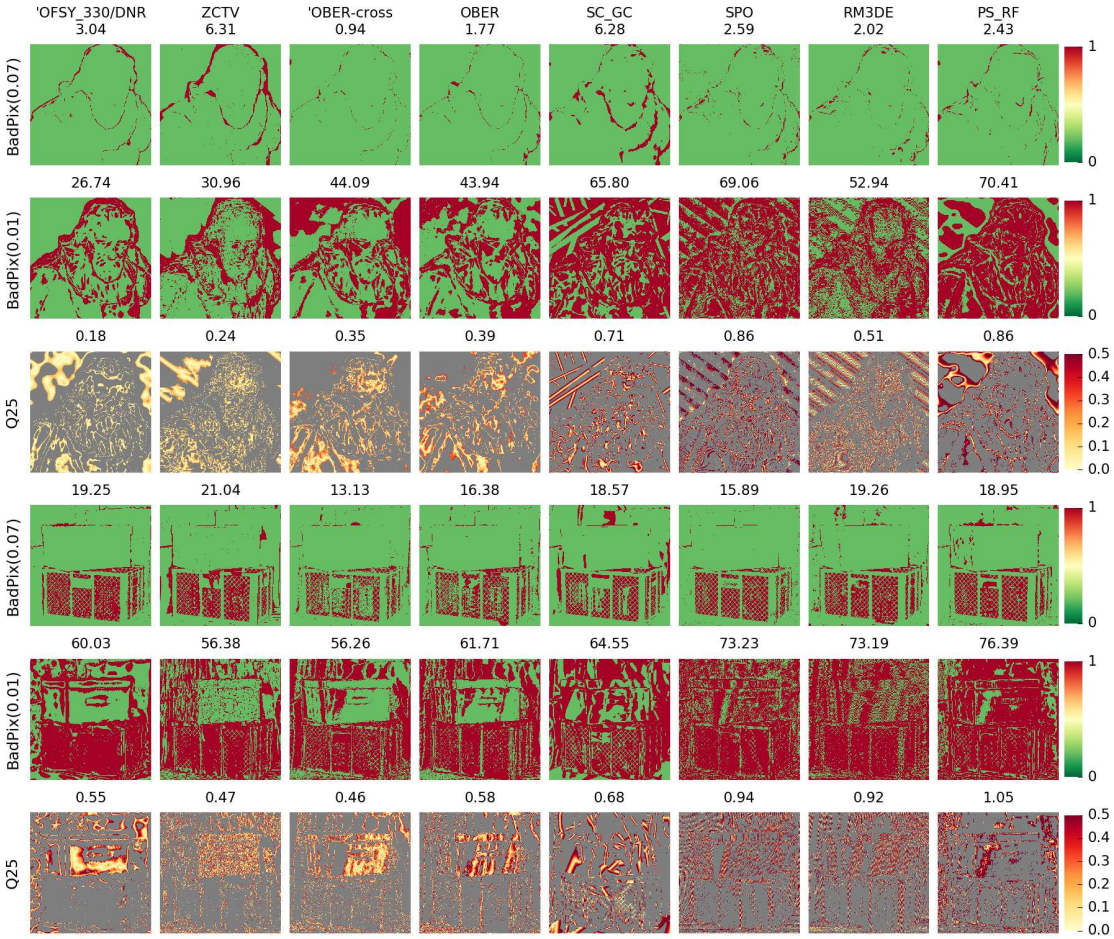

Figure 12. Visualizations for *BadPix(0.07)*, *BadPix(0.01)* and *Q25* performance are shown for the eight most accurate algorithms. On *Cotton*, algorithm performance is very similar for *BadPix(0.07)* on all regions except for strong discontinuities. With *BadPix(0.01)* the smooth surface reconstruction of `'OFSY_330DNR` and plane fitting artefacts of `SC_GC` become visible. The third row shows the absolute error of the 25% of the best pixels for each algorithm.

to problems at more complicated image regions like occlusions or discontinuities.

Another interesting curve is the one of PS_RF. Compared to most other curves, it has relatively few estimates with an error below 0.01, but is among the better algorithms at 0.03 and 0.07. This might be due to few labels (151) or the tuning of the algorithm for a certain threshold (amount of trees in the random forests). A similar behavior can be observed for SPO, which starts slow but is the second best among all algorithms at a threshold of 0.05.

Figure 12 illustrates how the accurate pixels of the top performing algorithms of figure 11 are distributed locally on *Cotton* and *Boxes*. It depicts the regions that fall into the *BadPix(0.07)* and *BadPix(0.01)* areas as well as the accuracy for those pixels that fall into the *Q25*, i.e. the regions with the 25% best accuracy for each algorithm. The visualization suggests that the top performing algorithms for the smaller error thresholds on figure 11 are those which are best at estimating the big, continuous surfaces which make up a huge proportion of the total pix count.

For three of the top four algorithms - 'OFSY_330DNR, OBER, and 'OBER-cross - the regions in which the high accuracy is achieved is mostly continuous. Figure 9 shows that these algorithms also feature the smoothest and most accurate normal maps. Comparing the zero crossing based algorithms OBER, 'OBER-cross, and ZCTV one can see that the variational regularization of ZCTV does not perform on par with the bilateral filter approach used by OBER when it comes to continuously high accurate surfaces. This is also apparent on the normals of the *Cotton* statue in figure 9 which are much noisier for ZCTV as compared to OBER and 'OBER-cross. The clustering of high accuracy regions can also be observed for 'OFSY_330DNR, and might be rooted in the use of the normal regularization as well as the use of 330 labels and the sublabel accurate cost volume optimization. By contrast, the *Q25* regions of less accurate algorithms are very discontinuous.

For SC_GC, the *Q25* visualization on figure 12 and the normal map on figure 9 illustrate how the plane fitting effects the position and kind of error. Despite the obvious drawbacks, this approach has one major advantage as it can jointly regularize discontinuous surfaces like e.g. the foreground grid at the bottom of *Boxes*. It is the only algorithm to correctly estimate that large regions on the grid with high accuracy (see *BadPix(0.01)* row on figure 12).

6.4. Occlusion handling

In this section, we evaluate the occlusion handling performance of different algorithms. Next to the metrics, a good way to evaluate algorithm performance at discontinuities is to take a look at figures 7 and 8. Algorithms with above average occlusion performance will show green halos at discontinuities. An important question is the influence of the explicit modeling of occlusions in data terms as well as regularizers.

Two algorithms, *LF_OCC and OMG_occ, explicitly model occlusion boundaries in SCams. For the complicated case of occlusions at the grid in *Boxes* (see figure 15 for an explanation why it is challenging) it can be seen that they perform above average. For easier cases of occlusions, both generally perform below average on discontinuities as is reflected by the discontinuity metric and a view at the median comparison figures.

SC_GC uses a very similar approach as these two algorithms, but selects the correct views automatically by choosing the 50% best views. SC_GC seems to generally perform better at occlusion boundaries, except it introduces fine thinning. It remains unclear if the difference in performance is due to the dataterm or the post processing.

Algorithms that perform above average at occlusion boundaries are 'OBER-cross, SPO, RM3DE, OBER, and SC_GC. It is especially interesting that SPO performs that well in this category as their dataterm does not model occlusion boundaries explicitly. Apparently, comparing region based histograms is quite robust to occlusion boundaries.

The way views are selected and combined to yield occlusion awareness is particularly interesting. Looking at the top performing algorithms for the discontinuities metric, only SC_GC uses the whole light field. It seems like for discontinuities the size of the input light field is not as important as the technique applied.

6.5. Influence of view configuration

As shown in figure 13, most algorithms either use the full light field or the crosshair. As special cases, RM3DE uses the diagonals in addition to the crosshair, *EPI1 uses only a subset of the crosshair. OBER and ZCTV use only a horizontal line.

Looking at the *BadPix(0.03)* and *Q25* metrics in figure 13, no definite connection between the number of views and the performance is visible. However, more views generally tend to lead to higher runtimes. The crosshair might be the best trade-off between the amount of data used (and thus runtime) and the achieved accuracy. But runtime and accuracy vary strongly, especially among algorithms with crosshair setups. Hence, individual algorithm aspects seem to be more important than view configuration.

Another interesting question is the relation between the number of depth labels and the maximum accuracy achieved by an algorithm. As depicted on the right of figure 13, algorithms with more than 200 depth labels tend to have better *Q25* scores, though the correlation is not very strong.

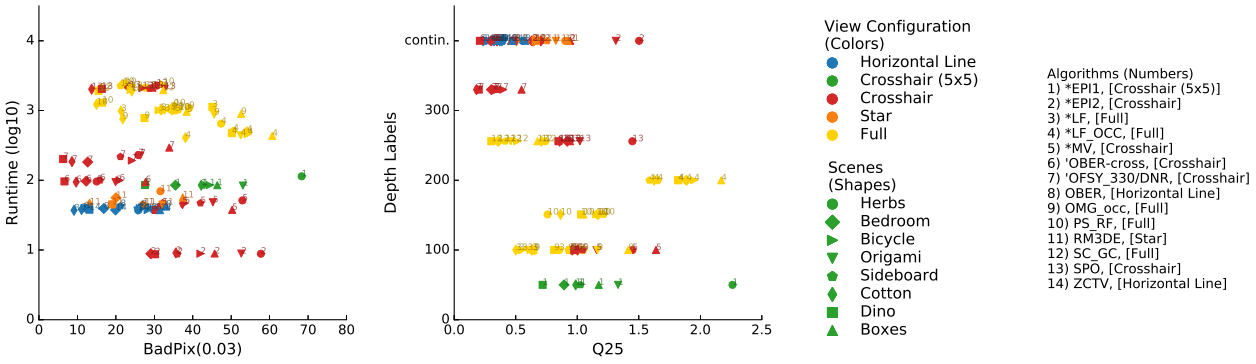

Figure 13. Colors represent the view configuration while shapes indicate scenes and numbers identify algorithms. **Left:** The 6 algorithms taking into account the full 81 views (yellow) tend to be slower than other algorithms. However, the additional views or runtimes do not lead to significantly better *BadPix* scores. **Right:** Algorithms with more than 200 depth labels tend to have better *Q25* scores.

7. Analysis of local scene difficulty

Figure 14 illustrates the absolute disparity errors of the artificial *PerPixMedian* and *PerPixBest* algorithms as introduced in Section 5.2. The *PerPixMedian* results show that "the average algorithm" struggles mostly with occlusions, the noisy image regions on *Dots* and with low texture regions such as the lower part of *Stripes*, the lamps on *Bedroom*, or the door in *Bicycle*. By contrast, the planar walls and floors as well as most continuous surfaces are accurately reconstructed.

The spatial variation of scene characteristics on *Dots* and *Stripes* makes these scenes challenging to be solved with a single algorithm and parameterization. However, the *PerPixBest* results on these scenes indicate that the individual parts can be solved accurately. Three other types of image areas remain challenging, even for the *PerPixBest* algorithm: complex occlusions, very thin structures, and low texture areas.

Figure 15 illustrates why these areas are particularly complicated. Figure 15c shows a vertical slice of the left part of *Backgammon*. Both ends of the background EPIs are cut off, making estimation difficult. Similar challenges with complex occlusion occur on the grids of *Boxes* and the plants on *Herbs* and *Bicycle*. Figure 15d and 15e show that these situations are particularly challenging on *Boxes*. The books in the background of the box have very little texture, making it almost impossible to estimate the slope of the corresponding EPIs.

Figure 15b demonstrates how the very thin tips of *Backgammon* merge with the background due to aliasing effects. This leads to problems while estimating disparity, as it appears like a scene where two different depth layers are superimposed, breaking the Lambertian assumption. Similarly, the shoes on *Sideboard* and the cap on *Origami* are non-Lambertian and pose additional challenges to the algorithms. Looking at these regions from the perspective of EPIs, specularities create non-linear patterns, which can only partially be handled by algorithms that build on the Lambertian assumption. Thus, objects with arbitrary BRDFs represent an interesting direction for future datasets.

8. Conclusion and outlook

In this paper, we review representations of the light field and strategies for disparity estimation to introduce a taxonomy of current light field depth estimation algorithms. We characterize and categorize algorithms according to their data terms, optimization techniques, and refinement steps. We thoroughly evaluate 14 algorithms in a variety of ways, e.g. with respect to their occlusion handling performance, their robustness to errors on the input images and their ability to produce smooth surfaces. In addition to the benchmark proposed in [7], we introduce novel metrics to evaluate the algorithm's best-case accuracy as well as the error in surface normal estimation.

The evaluation reveals that most challenge participants easily outperform the initial baseline algorithms provided with the benchmark. Algorithms with considerably strong performance in certain aspects are `SPO` at discontinuities, `'OFSY_330DNR` at continuous surfaces, and `'OBER-cross` for highest accuracy and overall good performance. However, there is no single algorithm that excels in every category.

Most algorithms consist of multiple components and it is difficult to establish which of these are most influential for performance. While our experiments indicate that occlusion aware dataterms and a good approach to surface regularization which does not oversmooth at depth discontinuities play a crucial role for excellent algorithm results, it seems necessary to separate the evaluation of cost functions and optimization or post-processing techniques, respectively. For a more in-depth taxonomy and evaluation, algorithm components should be analyzed individually. A

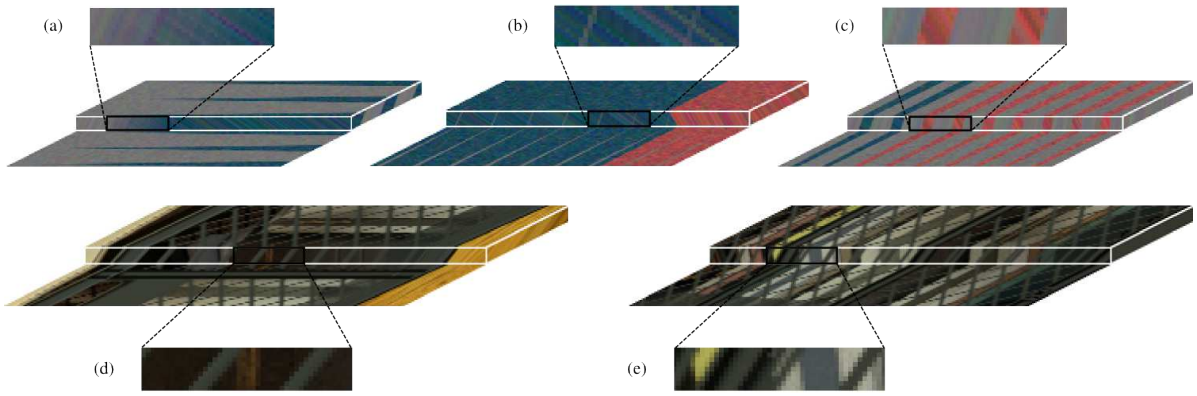

Figure 15. The selected EPIs of *Backgammon* and *Boxes* illustrate why these scenes are particularly challenging. **Top:** On *Backgammon*, disparity estimation is challenging due to (a) superimposed orientations caused by aliasing (b) thin structures merging with the background, and (c) complex occlusions where epipolar lines are cut off at both ends. **Bottom:** Similar problems can be observed on *Boxes*. The books of the background are occluded at both ends of the epipolar line. The low texture of the books makes disparity estimation particularly challenging.

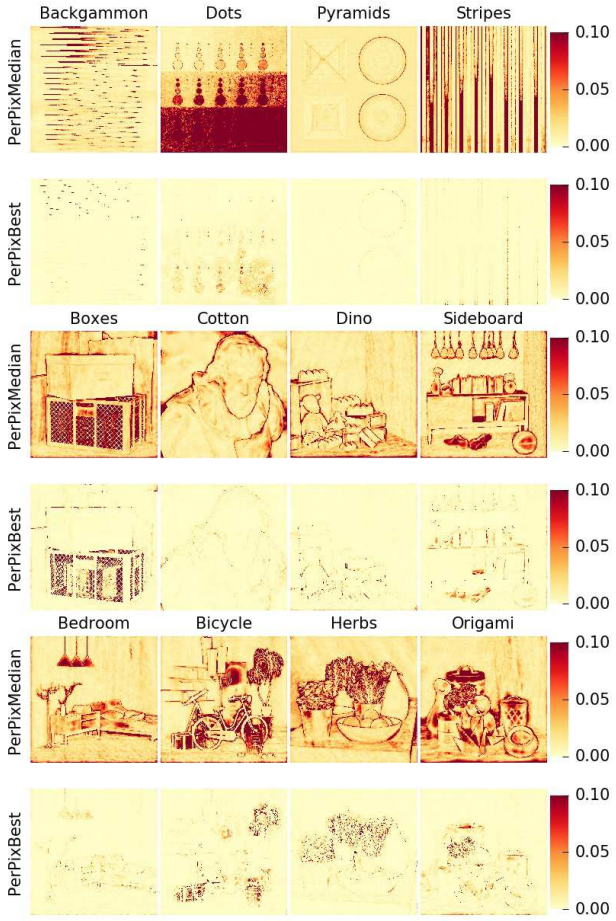

Figure 14. Based on the average and optimum performance of the *PerPixMedian* and *PerPixBest* algorithms, the three biggest challenges are: complex occlusions (e.g. grid on *Boxes*, plants on *Herbs*), low texture areas (e.g. shoes on *Sideboard*), and very thin structures (e.g. peaks on *Backgammon*).

first approach could be to provide standardized cost volumes together with the datasets in order to evaluate optimization and post-processing, as well as standardized optimization schemes to evaluate individual dataterms.

Although the evaluation demonstrates an overall very good performance of the state-of-the-art algorithms, there are still challenging open problems. Among these are better occlusion modeling, better discontinuity-aware regularization, improving runtime while keeping quality similar, and a good way of deciding which light field representation yields optimal results in which situation.

In addition, open challenges lie in the reconstruction of non-Lambertian surfaces and BRDF estimation, which are hard to impossible to tackle with only a sparse set of views. Here, the structure of light fields can help to arrive at unique and novel solutions. Thus, for future work, we aim at creating more datasets and evaluation methodology for more diverse and challenging scenes including challenging BRDFs and geometry, and also include real world datasets with carefully measured ground truth.

9. Acknowledgments

We gratefully acknowledge financial support for this research by: ERC Starting Grant “Light Field Imaging and Analysis” (LIA 336978, FP7-2014); Heidelberg Collaboratory for Image Processing (HCI) within the Institutional Strategy ZUK49 “Heidelberg: Realizing the Potential of a Comprehensive University”, Measure 6.4 including matching funds from the industry partners of the HCI; AIT Austrian Institute of Technology, Vienna, Austria; SFB Transregio 161 “Quantitative Methods for Visual Computing”.

References

[1] Anonymous. Occlusion-model guided anti-occlusion depth estimation in light field. *Submitted as `OMG_occ` to lightfield-analysis.net*, 2017. 5

[2] Anonymous. Zero crossings for depth estimation in light fields. *Submitted as `ZCTV`, `OBER`, and `'OBER-cross` to lightfield-analysis.net*, 2017. 5

[3] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987. 3

[4] C. Chen, H. Lin, Z. Yu, S.-B. Kang, and Y. J. Light field stereo matching using bilateral statistics of surface cameras. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[5] A. Criminisi, S. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer vision and image understanding*, 97(1):51–85, 2005. 3

[6] M. Diebold and B. Goldluecke. Epipolar plane image refocusing for improved depth estimation and occlusion handling. In *Vision, Modeling and Visualization (VMV)*, 2013. 6

[7] 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*, pages 19–34. Springer, 2016. 1, 2, 4, 7, 8, 16

[8] K. Honauer, L. Maier-Hein, and D. Kondermann. The HCI Stereo Metrics : Geometry-Aware Performance Analysis of Stereo Algorithms. *Proc. International Conference on Computer Vision (ICCV)*, pages 2120–2128, 2015. 8

[9] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 4, 5

[10] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon. Depth from a light field image with learning-based matching costs. *Submitted as `PS_RF` to lightfield-analysis.net. The work has been submitted the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5

[11] O. Johannsen, A. Sulc, and B. Goldluecke. Occlusion-aware depth estimation using sparse light field coding. In *German Conference on Pattern Recognition*, pages 207–218. Springer International Publishing, 2016. 1

[12] O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3270, 2016. 1, 3, 4

[13] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006. 2

[14] H. Lin, C. Chen, S.-B. Kang, and J. Yu. Depth recovery from light field using focal stack symmetry. In *Proc. International Conference on Computer Vision (ICCV)*, 2015. 4

[15] S. Nayar and Y. Nakagawa. Shape from Focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, 1994. 4

[16] 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. 1, 5

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002. 2

[18] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. 5

[19] L. Si and Q. Wang. Dense depth-map estimation and geometry inference from light fields via global optimization. In *Asian Conference on Computer Vision*, pages 83–98. Springer, 2016. 1, 5

[20] 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)*, 2017. 1, 4

[21] M. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proc. International Conference on Computer Vision (ICCV)*, 2013. 4

[22] M. Tao, P. Srinivasan, S. Hadap, S. Rusinkiewicz, J. Malik, and R. Ramamoorthi. Shape estimation from shading, defocus, and correspondence using light-field angular coherence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 4, 6

[23] T. Wang, A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015. 1, 3, 4

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

[25] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition (Proc. GCPR)*, 2013. 1, 3, 4

[26] S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4D light fields. In *Vision, Modelling and Visualization (VMV)*, 2013. 1

[27] S. Wanner, C. Straehle, and B. Goldluecke. Globally consistent multi-label assignment on the ray space of 4D light fields. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 4

[28] J. Yu, L. McMillan, and S. Gortler. Surface camera (scam) light field rendering. *International Journal of Image and Graphics*, 4(04):605–625, 2004. 3

[29] 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. 1, 5