

Fast and Accurate 4D Light Field Depth Estimation

Numair Khan, Min H. Kim and James Tompkin

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-20-01
August 2020

Fast and Accurate 4D Light Field Depth Estimation

Numair Khan¹, Min H. Kim², and James Tompkin¹

¹ Brown University

{numair_khan, james_tompkin}@brown.edu

² KAIST

minhkim@vclab.kaist.ac.kr

Abstract. We present an algorithm for accurate depth estimation from 4D light fields that runs almost an order of magnitude faster than classical methods. Our proposed approach use epipolar-plane image edges to estimate sub-pixel disparity at a small set of pixels in the 4D space. By optimizing constraints at these pixels we are able to diffuse the sparse set in an occlusion-aware manner to obtain dense disparity maps. Qualitative and quantitative results on both synthetic and real-world light fields show that we have comparable, or better performance than existing methods, while being significantly faster (8–11 $\times$) than current non-learning-based methods.

1 Introduction

Light fields record small view changes on a scene. This allows them to store both the spatial and angular distribution of light. The additional angular dimension allows many novel imaging applications such as synthetic aperture photography, high-dynamic-range imaging, and view interpolation [25][12]. Most of these applications can be directly implemented in image space using image-based rendering (IBR) techniques [17][8]. But for applications such as light field editing and augmented reality it is often useful to have an explicit scene representation in the form of a 3D mesh or depth map.

However, light field depth estimation is a difficult problem. The extra angular dimension of light fields carries large data costs which makes most depth estimation algorithms computationally inefficient. Recent methods have sought to overcome this barrier by learning data-based priors. But while deep learning is common and effective, it requires specialized hardware, training data, and has many memory-consuming parameters. Moreover, given the sparsity of available light field training sets, these methods tend to overfit to specific scenes or capture scenarios. Algorithmic solutions that do not involve learning are typically more computationally general and flexible.

We present a first principles light field depth estimation method with no learned priors, which is comparable to the best-performing methods in terms of accuracy while being almost an order-of-magnitude (8–11 $\times$) faster than state-of-the-art non-learning-based methods. We achieve these results by estimating

disparity at a sparse set of pixels identified as most important for the final result. These estimates are then propagated to all pixels using an occlusion-aware diffusion process. Traditionally, diffusion pipelines for depth completion follow three steps [9][26]: 1) obtain sparse depth labels, 2) determine diffusion direction at each labeled point, and 3) perform dense depth diffusion. Step 2 is critical yet difficult: determining the diffusion direction requires us to know the depth at pixels around each label, but for efficiency we only have a sparse set of labeled points. Holynski and Kopf [9] deal with this by assuming that sparse labels do not lie on depth edges so that neighboring pixels have a similar label. Yucer et al. [26] handle labels on depth edges, but their method is designed for light fields with a large number (3000+) of views.

Our novel contribution is that we determine diffusion direction from other sparse labels in non-local neighborhoods via a novel backward-forward diffusion process. This also optimizes the data weight constraint allowing us to differentiate between texture and depth edges. Our method is robust to noise and low-gradient edges, allowing it to work on light fields with few views (e.g., 7x7 from a Lytro), and work whether the sparse labels lie on edges or not.

The source code for our method is available as open source software.¹

2 Related Work

Many light field depth estimation methods seek to exploit the regular structure of an epipolar-plane image (EPI). Khan et al. [16] use a set of large Prewitt filters to reliably detect oriented lines in EPIs. However, while their approach is efficient and works for their purpose of superpixel segmentation, it lacks the sub-pixel precision required for accurate depth prediction. Zhang et al.’s [27] spinning parallelogram operator works in a similar fashion on EPIs, but has a larger support than a Prewitt filter, and provides more accurate estimates. Their approach is similar to Tošić and Berkner’s [22] convolution with a set of specially adapted kernels to create light field scale-depth spaces. Wang et al. [23] [24] build on this by proposing a photo-consistency measure to address occlusion. Tao et al.’s [21] work considers higher dimensional representations of EPIs which allows them to use both correspondence and defocus for depth estimation.

The relation between defocus and depth is also exploited by the sub-pixel cost volume of Jeon et al. [13], who also present a method for dealing with the distortion induced by micro-lens arrays. An efficient and accurate method for wide-baseline light fields was proposed by Chuchwara et al. [6]. They use an oversegmentation of each view to get initial depth proposals, which are iteratively improved using PatchMatch [3]. As such, their strategy of progressively refining a coarse depth estimate is similar in some respects to our own work. Also similar to our method is the work of Holynski and Kopf [9] who present an efficient method for depth densification from a sparse set of points for augmented reality applications. However, they assume that the set of sparse points, and their depth

¹ github.com/brownvc/lightfielddepth

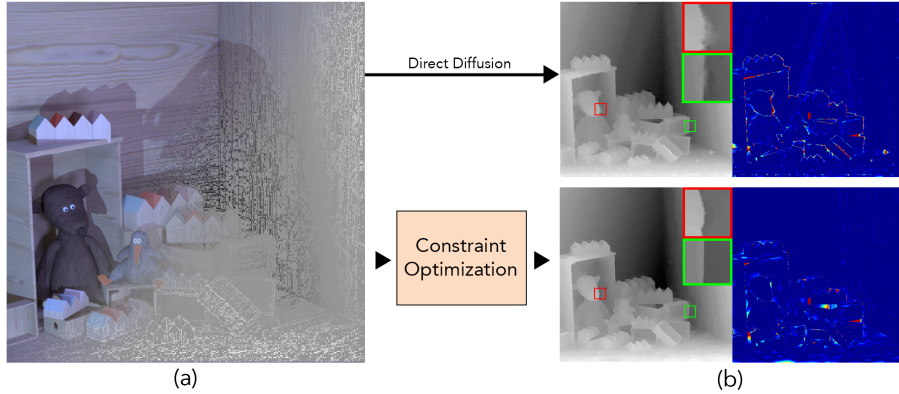


Fig. 1: **(a)** We find sparse sub-pixel disparity labels at image edges using EPI-space edge detection. **(b,top)** Direct diffusion of the sparse labels leads to bleeding around low gradient edges. **(b,bottom)** Our proposed constraint optimization strategy leads to stronger disparity gradients, lower MSE, and fewer outliers.

value is known beforehand. Our method does not make this assumption, and seeks to identify both the points and their depth, as well as performing dense diffusion. Yucer et al. [26] present a diffusion-based method that uses image gradients to estimate a sparse label set. However, their method is designed to work for light fields with thousands of views. Chen et al. [5] estimate accurate occlusion boundaries from superpixels to regularize the depth estimation process.

In recent years, many methods have sought to use data-driven methods to learn priors to avoid the cost of dealing with a large number of images, and to overcome the loss of spatial information induced by the spatio-angular tradeoff in lenslet images. Alperovich et al. [2] use an encoder-decoder architecture to perform an intrinsic decomposition of a light field, and also recover disparity for the central cross-hair of views. Huang et al. [11] provide a network-based solution that is able to handle an arbitrary number of uncalibrated views. Jiang et al. [14][15] fuse the disparity estimates at four corner views estimated using a deep learning optical-flow method. Shi et al. [19] build on this by adding a refinement network to the fusion pipeline.

3 Our Approach

Our goal is to estimate disparity at a sparse set of points such that their labels can be accurately and efficiently diffused to all pixels. Any arbitrary sparse set of points does not prove very useful. Instead we want to select points that are most representative of the actual disparity. Past work on image reconstruction has shown that edges are sufficient for recovering a perceptually accurate representation of the original image [7]. Hence, we populate our sparse set for diffusion by selecting points around light field edges.

3.1 Sparse Depth Labeling

An EPI (Epipolar-Plane Image) provides an angular slice through a 4D light field, and has a very regular linear structure resulting from epipolar geometry constraints: points in world space become lines in an EPI, with the slope of each line corresponding to the depth of the point. The regularity of an EPI makes it easy to identify salient edges and their depth at the same time.

Thus, for each EPI I in the central cross-hair of light field views we use a set of large Prewitt filters [16] to recover a set $\mathcal{L}$ of parametric lines representing edge points in 4D space. This process, while fast, tends to generate many false-positives. To filter these out we use a gradient-based alignment scheme: each line $l \in \mathcal{L}$ is sampled at n locations to generate the set of samples $S_l = \{(x_i, y_i)\}$. The line l is considered a false-positive if the local image gradient of I does not align with the line direction at a minimum k number of samples:

$$\sum_{\mathbf{s} \in S_l} \mathbb{1} \left(\frac{\nabla I(\mathbf{s})(\nabla l)^T}{\|\nabla I(\mathbf{s})\| \|\nabla l\|} > \cos(\tau_f) \right) < k \quad (1)$$

where $\mathbb{1}(\cdot)$ is the characteristic function of the set of aligned samples, ∇I is the first-order image gradient approximated using a 3x3 Sobel filter, and ∇l is perpendicular to the line. The parameters τ_f and k are constants with $\tau_f = \pi/13$ and $k = (\text{EPI height})/c$, with $1 \leq c \leq \text{EPI height}$. To determine the constant value c , we consider two factors: 1) the accuracy of EPI line fitting, and 2) the expected minimum number of views a point is visible in. In the case of perfect alignment between the line and EPI gradients, $c = 1$. This means that a line with even a single misaligned sample is rejected. However, if a point is occluded in some views, the corresponding EPI line will be hidden and misalignment of samples in those views is inevitable. If we set $c = 1$ we risk discarding such lines. We determine empirically that $c = 4$ provides good results for all light fields.

The parametric definition of EPI lines does not carry any visibility information for a point across light field views. We determine the visibility $v(l)$ of a point $l \in \mathcal{L}$ in the central light field view as:

$$v(l) = \mathbb{1} \left(\frac{\nabla I(\mathbf{s}_c) \nabla l}{\sqrt{\nabla I(\mathbf{s}_c)^2 + \nabla l^2}} > \cos(\tau_v) \right) \quad (2)$$

where $\mathbf{s}_c$ is the EPI sample corresponding to the central view and $\tau_v = \pi/10$.

Notice that the number of discrete disparity values of points in $\mathcal{L}$ is bounded by the number of large Prewitt filters used for EPI line fitting. Computational efficiency considerations prevent this number from becoming too large. Moreover, numerical precision and sampling errors result in the granularity of depth estimates plateauing beyond a certain number of filters. Thus, to enable the calculation of sub-pixel disparity values we fine-tune the initial estimates through random search and trilateral filtering. Let $\mathcal{L}_c = \{l \in \mathcal{L} \mid v(l) = 1\}$. Then for each $l \in \mathcal{L}_c$ and image samples $S_l = \{(x_i, y_i)\}$ along the line we minimize the energy

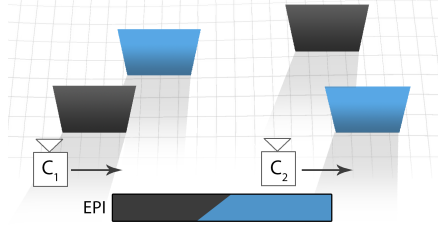


Fig. 2: The slope of an EPI edge alone does not disambiguate the depth of surfaces on either side of the edge. In the example shown above, both cameras C_1 and C_2 move across the scene to the right and generate the same EPI even though the depth ordering of the scene viewed by each camera is different.

function defined by the entropy of normalized intensity values:

$$E(l) = \sum_{\mathbf{s} \in S_l} -P(I(\mathbf{s})) \log_2(P(I(\mathbf{s}))) \quad (3)$$

where $I(\mathbf{s})$ is the intensity value at $\mathbf{s}$ and $P(\mathbf{s})$ is estimated from a histogram.

We minimize $E(l)$ by performing a random search in the 2D parameter space defined by the x-intercepts of l on the top and bottom edge of the EPI, $l = (x_t, x_b)$: at the j th iteration of the search we generate uniform random numbers $(o_t, o_b) \sim U(-1, 1)(\alpha t^j)$, to generate a proposal $l_j = (x_t + o_t, x_b + o_b)$. This is accepted with probability one if $E(l_j) < E(l_{j-1})$. We use $t = 0.88$, $\alpha = 0.15$ and run the search for a maximum of 10 iterations.

The resulting disparity estimates are then refined by filtering jointly in the spatial, disparity, and LAB color space. Let $\mathcal{P}$ represent the spatial projection of $\mathcal{L}_c$ into the central view, and let p_s, p_d , and p_c be the spatial position, disparity, and color of a point $p \in \mathcal{P}$. Then the refined disparity estimate $f(p_d)$ is calculated using a spatial neighborhood $\mathcal{S}$ around p as

$$f(p_d) = \frac{1}{W} \sum_{q \in \mathcal{S}} \mathcal{N}_{\sigma_s}(\|p_s - q_s\|) \mathcal{N}_{\sigma_d}(p_d - q_d) \mathcal{N}_{\sigma_c}(\|p_c - q_c\|) p_d \quad (4)$$

where the normalization factor W is given by

$$W = \sum_{q \in \mathcal{S}} \mathcal{N}_{\sigma_s}(\|p_s - q_s\|) \mathcal{N}_{\sigma_d}(p_d - q_d) \mathcal{N}_{\sigma_c}(\|p_c - q_c\|) \quad (5)$$

We found that $\sigma_s = 10$, $\sigma_d = 0.1$ and $\sigma_c = 0.5$ work for all scenes.

3.2 Diffusion Constraint Optimization

In order to propagate the sparse set of disparity labels in $\mathcal{P}$ to a dense grid of pixels, we need to determine the diffusion direction at each $p \in \mathcal{P}$. As Figure 2 shows, the disparity for a point lying on an EPI edge alone is not sufficient for determining the surface direction in which to perform diffusion. Yucer et al.[26]

solve this problem by considering the image gradients on either side of an EPI edge. This strategy has the added advantage of enabling them to differentiate depth edges from texture edges as the latter will have similar gradients on either side of the edge. Such differentiation of edges is desirable as texture edges may cause undesirable discontinuities in smooth depth regions. Unfortunately, their approach fails to generate good results for light fields with a small number of views such as those captured using a lenslet array or a camera rig (Yucer et al. consider light fields with 3000+ images).

Our proposed solution to the diffusion direction problem works for light fields with few views (e.g., 7x7 from a Lytro). Moreover, it allows the differentiation of depth and texture edges by providing a measure of how much each $p \in \mathcal{P}$ contributes to the final disparity map: points lying on texture edges will have a low contribution to disparity gradients in the final result.

We use $S[\mathcal{A}]$ to represent the image created by splatting sparse points in a set $\mathcal{A}$ onto a $w \times h$ raster grid, and D to be a dense $w \times h$ disparity map. Diffusion is formulated as a constrained quadratic optimization problem:

$$\hat{D}[\mathcal{A}] = \underset{D}{\operatorname{argmin}} \sum_{p \in \mathcal{A}} E_d(p) + \sum_{(p,q) \in \mathcal{S}} E_s(p,q) \quad (6)$$

where $\hat{D}[\mathcal{A}]$ is the optimal disparity map given the sparsely labeled image $S[\mathcal{A}]$ and $\mathcal{S}$ is the set of all four-connected neighbors in D . The data term $E_d(p)$ and smoothness term $E_s(p,q)$ are defined as:

$$E_d(p) = \lambda_d(p) \|S[\mathcal{A}](p) - D(p)\|, \quad (7)$$

$$E_s(p,q) = \lambda_s(p) \|D(p) - D(q)\| \quad (8)$$

with $\lambda_d(\cdot)$ and $\lambda_s(\cdot)$ being the spatially-varying data and smoothness weights.

Equation 6 represents a standard Poisson problem, and we solve it using an implementation of the LAHBPCG solver [20] by posing the constraints in the gradient domain as proposed by Bhat et al. [4]. We begin by defining the sets

$$\mathcal{P}_f = \{p + \nabla I(p) \mid \forall p \in \mathcal{P}\} \quad (9)$$

$$\mathcal{P}_b = \{p - \nabla I(p) \mid \forall p \in \mathcal{P}\} \quad (10)$$

where $\nabla I(p)$ is the gradient of the central light field view at point p . Then we solve Equation 6 for $\hat{D}[\mathcal{P}_f]$ and $\hat{D}[\mathcal{P}_b]$ using weights

$$\lambda_s(p) = \frac{1}{\|\nabla I(p)\| + \epsilon} \quad (11)$$

$$\lambda_d(p) = \begin{cases} 10^6 & \text{if } p \in \mathcal{A}, \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

We compare the normalized depth profile around each point $p \in P$ along $\nabla I(p)$ in $\hat{D}[\mathcal{P}_f]$ and $\hat{D}[\mathcal{P}_b]$. As shown in Figure 3, the profile for the correct offset direction ($\nabla I(p)$ or $-\nabla I(p)$) more closely resembles a step function around p

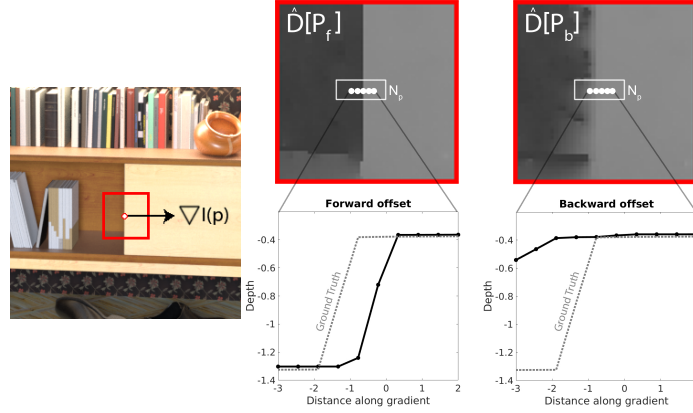


Fig. 3: Given an edge point p with image gradient $\nabla I(p)$ and depth label p_d we would like to determine which side of the edge to diffuse p_d . We generate images $\hat{D}[\mathcal{P}_f]$ and $\hat{D}[\mathcal{P}_b]$ by solving a Poisson optimization problem with diffusion direction $p + \nabla I(p)$ and $p - \nabla I(p)$ respectively. The correct diffusion direction generates an intensity profile resembling a step function. In the example shown, p_d corresponds to the surface on the right of the edge as $p + \nabla I(p)$ generates a profile more closely resembling a step function.

due to a strong depth gradient. This happens because neighboring points in the correct offset direction will have a disparity value similar to p . The high data term together with the global smoothness constraint results in a small gradient around p when the incorrect offset pushes it to the wrong side of the edge. We estimate the profile around p in each of $\hat{D}[\mathcal{P}_f]$ and $\hat{D}[\mathcal{P}_b]$ by convolving the normalized value of a set N_p of pixels around p with the step filter $\mathbf{F} = [-1 \ -1 \ 1 \ 1]$.

The two-way diffusion scheme allows us to differentiate between depth and texture edges by providing a measure of how much a point contributes to disparity gradients in the final result. We define the “edgeness” at point p as

$$\lambda_e(p) = \max_{\{\hat{D}[\mathcal{P}_f], \hat{D}[\mathcal{P}_b]\}} \|N_p \otimes \mathbf{F}\| \quad (13)$$

Points with a $\lambda_e(\cdot)$ score in the 10th percentile are discarded as texture edges. We refine the disparity label at each remaining point $p \in P$ by creating a sweeping depth volume and minimizing the pixel-wise variance in a window around p . Using the existing disparity estimates to select a small set of sweeping planes allows us to perform this step with very low computational overhead.

3.3 Dense Diffusion

The final dense disparity map is generated by optimizing Equation 6 to obtain $\hat{D}[\mathcal{Q}]$ where $\mathcal{Q} = \{p \pm \nabla I(p) \mid p \in \mathcal{P}\}$ is a sparse set of points offset in the

Table 1: Quantitative comparison of our method with two baseline non-learning-based methods on the central view of four 9×9 synthetic light fields. The running time shown is the cost to estimate depth for a single view of the light field. ‘BP(x)’ is the number of *bad pixels* above threshold x in error. Our approach has less error than competing methods while being $\sim 8\text{--}11\times$ faster.

Light Field	Run time (s)			MSE $\times 100$			BP(0.01)			BP(0.03)			BP(0.07)		
	[13]	[27]	Ours	[13]	[27]	Ours	[13]	[27]	Ours	[13]	[27]	Ours	[13]	[27]	Ours
<i>Sideboard</i>	754	537	62.1	3.21	1.02	0.89	63.7	78.0	73.8	40.6	42.0	37.4	26.7	14.4	16.2
<i>Dino</i>	805	531	62.4	1.73	0.36	0.45	76.4	77.3	69.4	48.6	40.4	30.8	22.7	6.90	10.4
<i>Cotton</i>	748	530	59.4	12.5	1.81	0.68	52.6	75.3	56.2	21.4	34.7	18.0	9.05	4.40	4.90
<i>Boxes</i>	736	541	58.5	16.0	7.90	6.70	69.3	84.7	76.8	45.5	55.3	47.9	26.8	18.9	28.3
Average	761	535	60.6	8.37	2.77	2.18	65.5	78.8	69.0	39.1	40.2	33.5	21.3	11.2	14.9

diffusion direction previously determined. The smoothness and data weights are

$$\lambda_d(p) = \omega \exp(a\lambda_e(p)) \quad (14)$$

$$\lambda_s(p) = \frac{1}{\|\nabla I(p)\| \|\nabla H(p)\|} \quad (15)$$

$$H(p) = \mathbf{G} \circledast (\hat{D}[\mathcal{P}_f] + \hat{D}[\mathcal{P}_b]) \quad (16)$$

where $\mathbf{G}$ is the Gaussian filter and $H(\cdot)$ provides a heuristic estimate of disparity to guide the dense diffusion process. The parameters ω and a in Equation 14 are set to 1.5×10^2 and 3 respectively.

4 Experiments

4.1 Datasets

We use both synthetic and real world light fields for evaluation. The HCI Light Field Dataset [10] is a collection of synthetic scenes provided with high-resolution ground truth disparity maps for the central view. We use the four scenes *Dino*, *Sideboard*, *Cotton*, and *Boxes* from this dataset. Each scene has 9×9 views of 512×512 pixels. For real world data, we use four scenes from the EPFL MMSPG Light-Field Image Dataset [18]: *Grid*, *Bikes*, *Silos*, and *Sphynx*. These light fields were captured with a Lytro Illum camera and have 15×15 views with 434×625 pixels each. However, as the edge views tend to be noisy, we only use the central 7×7 views in our experiments. Finally, we demonstrate results on nine 17×17 light fields from the New Stanford Light Field Archive [1]. These are captured with a camera rig and have a wider baseline than the EPFL and HCI light fields. We scale their spatial resolution to 512×512 pixels.

Table 2: Quantitative comparison of our method with two baseline learning-based methods on the central view of a set of four 9×9 synthetic light fields. The running time shown is the cost to estimate depth for a single view of the light field. ‘BP(x)’ is the number of *bad pixels* which fall above threshold x in error. Our approach has lower error than the method of Jiang et al. [14] and $\sim 8\times$ faster runtime; our method has competitive error to the state of the art method of Shi et al. [19], without requiring any learning. As baseline methods have higher MSE but fewer bad pixels, the errors they do have must be larger. We confirm this by visualizing error plots in Figure 7

Light Field	Run time (s)			MSE * 100			BP(0.01)			BP(0.03)			BP(0.07)		
	[19]	[14]	Ours	[19]	[14]	Ours	[19]	[14]	Ours	[19]	[14]	Ours	[19]	[14]	Ours
<i>Sideboard</i>	72.3	507	62.1	1.12	1.96	0.89	53.0	47.4	73.8	20.4	18.3	37.4	2.70	9.30	16.2
<i>Dino</i>	59.3	500	62.4	0.43	0.47	0.45	43.0	29.8	69.4	13.1	8.80	30.8	4.31	3.60	10.4
<i>Cotton</i>	79.8	500	59.4	0.88	0.97	0.68	38.8	25.4	56.2	9.60	6.30	18.0	2.80	2.00	4.9
<i>Boxes</i>	56.2	491	58.5	8.48	11.6	6.70	66.5	51.8	76.8	37.1	27.0	47.9	21.9	18.3	28.3
Average	66.9	500	60.6	2.72	3.75	2.18	50.3	38.6	69.0	20.1	15.1	33.5	7.92	8.30	14.9

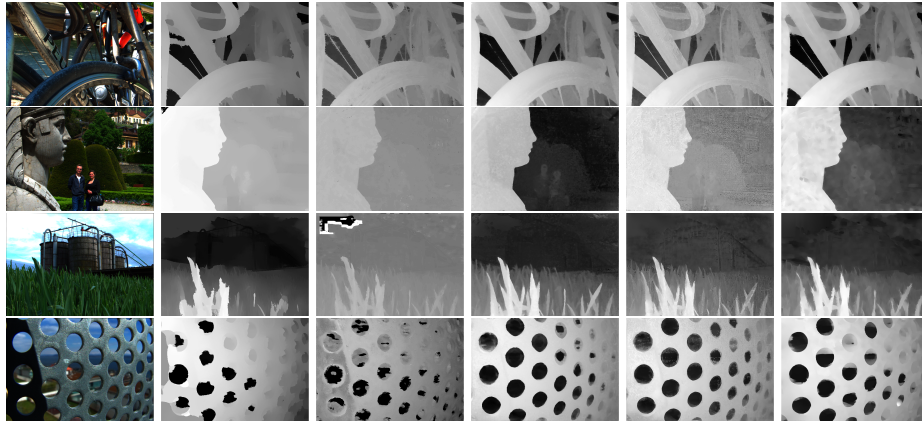


Fig. 4: Real-world EPFL database lightfields, captured using a Lytro camera. From left to right: central view, non-learning-based estimations of Jeon et al. [13] and Zhang et al. [27], learning-based methods of Jiang et al. [14] and Shi et al. [19], and our non-learning-based method.

4.2 Baselines

We compare our results to the state-of-the-art methods of Jeon et al. [13], and Zhang et al. [27]. For reference, we also compare with the learning-based methods of Jiang et al. [14], and Shi et al. [19]. Both methods use a neural network to estimate optical flow between the four corner views of a light field, which is then used to warp a set of anchor views. Shi et al. refine their final result by passing

Table 3: The average reprojection error onto the four corner views. We show results for real world light fields from the EPFL (top) and Stanford datasets (bottom). The error for each view is computed as the mean of the absolute difference between the real and reprojected view times 10^{-2} . Our method performs consistently better than all baselines on the real-world light fields of the EPFL dataset.

Light Field	Reprojection Error				
	[19]	[14]	[27]	[13]	Ours
<i>Bikes</i>	5.2	5.3	5.3	5.2	2.5
<i>Grid</i>	6.9	6.8	6.6	6.6	4.5
<i>Silos</i>	3.3	4.2	2.5	3.5	2.3
<i>Sphinx</i>	5.4	5.8	5.6	5.7	2.6
<i>Average</i>	5.2	5.5	5.0	5.3	3.0
<i>Bunny</i>	1.2	1.6	1.3	2.1	1.1
<i>Bulldozer</i>	2.0	3.4	2.1	8.9	2.4
<i>Chess</i>	1.7	1.7	1.7	6.6	1.8
<i>Eucalyptus</i>	1.2	1.6	1.3	3.5	1.1
<i>Jelly-Beans</i>	1.5	1.7	1.4	3.3	1.7
<i>Jewels</i>	2.3	2.5	2.7	6.6	2.3
<i>Lego</i>	2.5	4.4	3.1	7.6	3.0
<i>Tarot</i>	9.9	17	15	20	17
<i>Truck</i>	1.4	4.9	1.7	4.1	1.4
<i>Average</i>	2.6	4.3	3.4	7.0	3.6

it through an additional network trained on synthetic light field data. We show results of our evaluation on all datasets in Figures 4, 5, and 6.

4.3 Metrics

For evaluation on the synthetic light fields, where we do have ground truth, we use two standard evaluation metrics: mean-squared-error (MSE) times a hundred, and the percentage of pixels with an error higher than a specified threshold. The latter metric, referred to as *bad pixels*, is evaluated at the thresholds 0.01, 0.03, and 0.07. No ground truth is available for the real world scenes. Nevertheless, we evaluate the performance of our method quantitatively on these light fields as the reprojection error induced by warping the central view of the light field onto the corner views using the estimated disparity map. We also evaluate the computational efficiency of each method by reporting its running time. All three non-learning-based methods, and parts of Jiang et al., were implemented in MATLAB. The network code of Jiang et al. and Shi et al. was implemented

Table 4: Evaluating the key steps of sparse depth labeling (Section 3.1) and diffusion direction estimation (Section 3.2) in our approach by replacing them with the algorithms of Khan et al. [16] and Yucer et al. [26] respectively.

Light Field	MSE $\times$ 100			BP(0.01)			BP(0.03)			BP(0.07)		
	[16]	[26]	Ours	[16]	[26]	Ours	[16]	[26]	Ours	[16]	[26]	Ours
<i>Sideboard</i>	7.48	1.26	0.89	96.5	73.4	73.8	89.5	45.2	37.4	74.7	21.4	16.2
<i>Dino</i>	6.77	1.26	0.45	96.9	67.7	69.4	90.5	46.1	30.8	78.0	28.0	10.4
<i>Cotton</i>	10.7	1.81	0.68	97.9	76.2	56.2	93.9	46.1	18.0	82.9	24.4	4.90
<i>Boxes</i>	16.1	11.9	6.70	97.2	85.5	76.8	91.2	68.1	47.9	78.9	30.7	28.3
<i>Average</i>	10.3	4.06	2.18	97.1	75.7	69.0	91.3	51.4	33.5	78.9	30.7	14.9

in Tensorflow. All CPU code was run on an AMD Ryzen Threadripper 2950X 16-Core Processor, and GPU code on an NVIDIA GeForce RTX 2080Ti.

5 Discussion

We evaluate our approach to sparse depth labeling (Section 3.1) and diffusion direction estimation (Section 3.2) by replacing these key steps in our algorithm with the methods of Khan et al. [16] and Yucer et al. [26] respectively. Table 4 shows that our approach provides significantly higher accuracy on all metrics. As mentioned earlier, [16] does not provide sub-pixel accuracy, and [26] assumes a light field with a very large number (3000+) of views.

In terms of accuracy we observe that our method has consistently lower MSE on both synthetic and real world light fields, but suffers a higher number of bad pixels. This is indicative of more outliers in the baseline methods as the errors they do have must be considerably large. This intuition is confirmed by visualizing the MSE in Figure 7 where we see relatively fewer regions of red for our method compared to the baselines.

6 Conclusion

We have presented a light field depth estimation method that performs faster than both learning and non-learning based methods while providing comparable, or better accuracy. This is done by estimating accurate sub-pixel disparity labels at a sparse set of points, and then using labels in non-local neighborhoods to optimize constraints for fast diffusion-based densification. The effectiveness of this strategy is demonstrated by qualitative and quantitative results on both synthetic and real world light fields.



Fig. 5: HCI lightfield scenes. From top to bottom: central view, ground truth depth, non-learning-based estimations of Jeon et al. [13] and Zhang et al. [27], learning-based methods of Jiang et al. [14] and Shi et al. [19], and our non-learning-based method. Our method is more accurate than [13], [27] and [14], while being significantly faster ($8\text{--}11\times$). Compared to [19], we have fewer outliers.

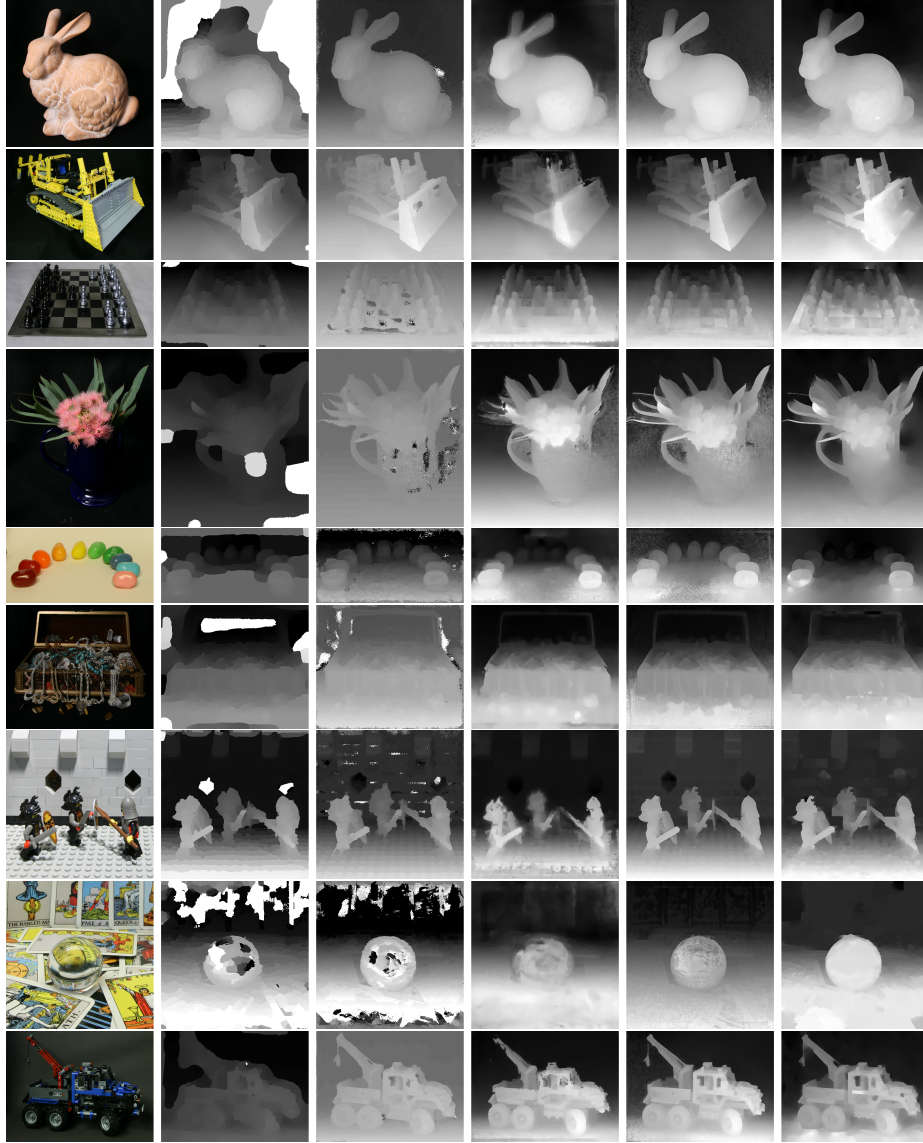


Fig. 6: Stanford lightfield scenes. From left to right: central view, non-learning-based estimations of Jeon et al. [13] and Zhang et al. [27], learning-based methods of Jiang et al. [14] and Shi et al. [19], and our non-learning-based method.

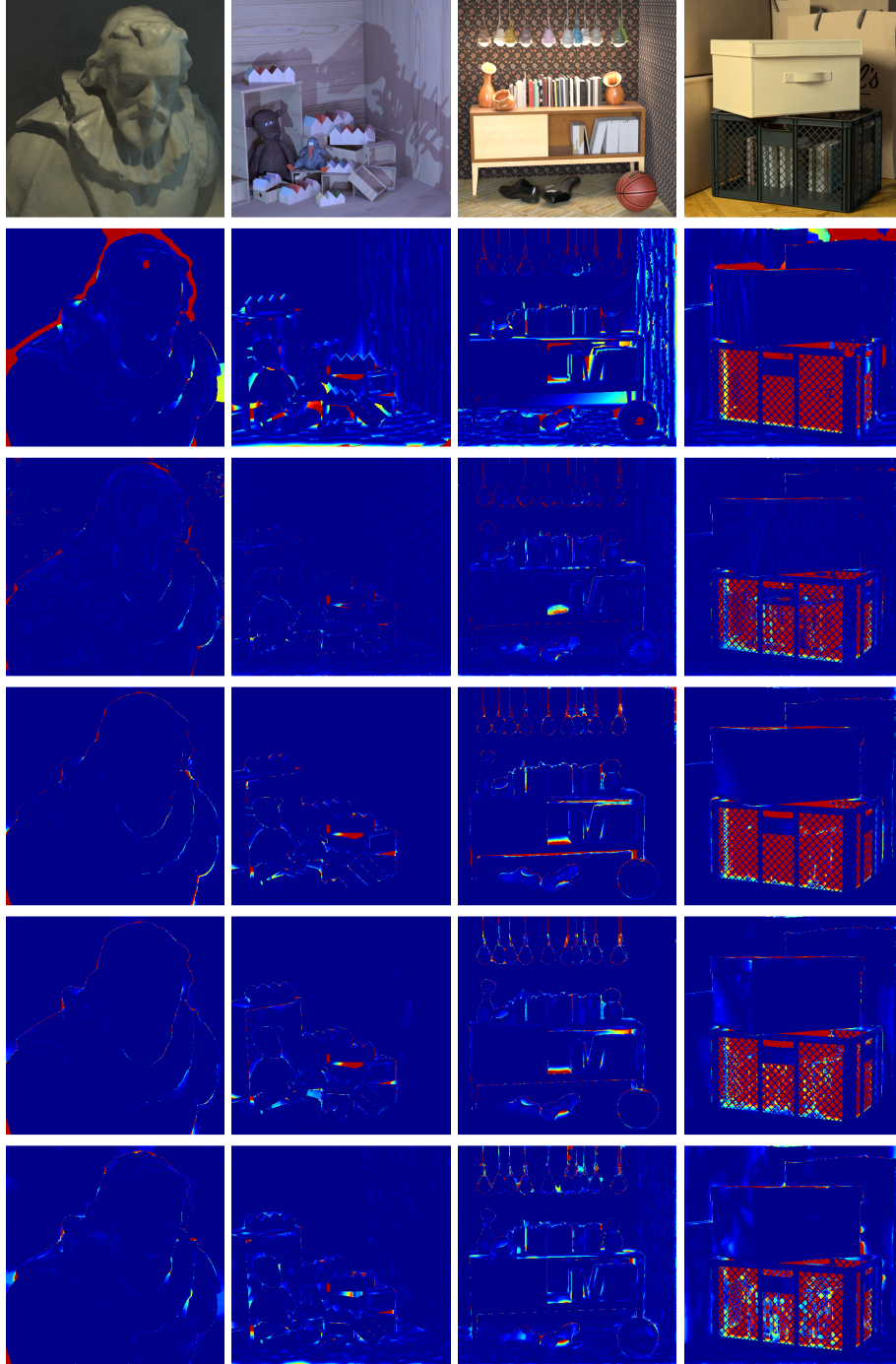


Fig. 7: The baseline methods have higher MSE but fewer bad pixels indicating that the errors they do have must be very large. Visualizing MSE confirms that overall our method has fewer regions with large errors, indicated in red.

References

1. The new stanford light field archive (2008), <http://lightfield.stanford.edu/>
2. Alperovich, A., Johannsen, O., Goldluecke, B.: Intrinsic light field decomposition and disparity estimation with a deep encoder-decoder network (2018)
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* **28**(3) (Aug 2009)
4. Bhat, P., Zitnick, L., Cohen, M., Curless, B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. In: *ACM Transactions on Graphics (TOG)* (2009)
5. Chen, J., Hou, J., Ni, Y., Chau, L.P.: Accurate light field depth estimation with superpixel regularization over partially occluded regions. *IEEE Transactions on Image Processing* **27**(10), 4889–4900 (2018)
6. Chuchvara, A., Barsi, A., Gotchev, A.: Fast and accurate depth estimation from sparse light fields. *IEEE Transactions on Image Processing* **29**, 2492–2506 (2020)
7. Elder, J.H.: Are edges incomplete? *International Journal of Computer Vision* **34**(2–3), 97–122 (1999)
8. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. pp. 43–54 (1996)
9. Holynski, A., Kopf, J.: Fast depth densification for occlusion-aware augmented reality **37**(6) (2018)
10. Honauer, K., Johannsen, O., Kondermann, D., Goldluecke, B.: A dataset and evaluation methodology for depth estimation on 4d light fields. In: *Asian Conference on Computer Vision*. pp. 19–34. Springer (2016)
11. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: DeepMVS: Learning multi-view stereopsis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2821–2830 (2018)
12. Isaksen, A., McMillan, L., Gortler, S.J.: Dynamically reparameterized light fields. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. pp. 297–306 (2000)
13. Jeon, H.G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.W., So Kweon, I.: Accurate depth map estimation from a lenslet light field camera. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1547–1555 (2015)
14. Jiang, X., Le Pendu, M., Guillemot, C.: Depth estimation with occlusion handling from a sparse set of light field views. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. pp. 634–638. IEEE (2018)
15. Jiang, X., Shi, J., Guillemot, C.: A learning based depth estimation framework for 4d densely and sparsely sampled light fields. In: *Proceedings of the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2019)
16. Khan, N., Zhang, Q., Kasser, L., Stone, H., Kim, M.H., Tompkin, J.: View-consistent 4d light field superpixel segmentation. In: *International Conference on Computer Vision (ICCV)* 2019. IEEE (2019)
17. Levoy, M., Hanrahan, P.: Light field rendering. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. pp. 31–42 (1996)
18. Rerabek, M., Ebrahimi, T.: New light field image dataset. In: *8th International Conference on Quality of Multimedia Experience (QoMEX)*. No. CONF (2016)
19. Shi, J., Jiang, X., Guillemot, C.: A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Transactions on Image Processing* **28**(12), 5867–5880 (2019)

20. Szeliski, R.: Locally adapted hierarchical basis preconditioning. In: ACM SIGGRAPH 2006 Papers, pp. 1135–1143 (2006)
21. Tao, M.W., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 673–680 (2013)
22. Tosić, I., Berkner, K.: Light field scale-depth space transform for dense depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 435–442 (2014)
23. Wang, T.C., Efros, A.A., Ramamoorthi, R.: Occlusion-aware depth estimation using light-field cameras. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3487–3495 (2015)
24. Wang, T.C., Efros, A.A., Ramamoorthi, R.: Depth estimation with occlusion modeling using light-field cameras. *IEEE transactions on pattern analysis and machine intelligence* **38**(11), 2170–2181 (2016)
25. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. In: ACM SIGGRAPH 2005 Papers, pp. 765–776 (2005)
26. Yucer, K., Kim, C., Sorkine-Hornung, A., Sorkine-Hornung, O.: In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 249–257. IEEE (2016)
27. Zhang, S., Sheng, H., Li, C., Zhang, J., Xiong, Z.: Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding* **145**, 148–159 (2016)