View-consistent 4D Light Field Depth Estimation

Numair Khan[1]
numair_khan@brown.edu
Min H. Kim[2]
minhkim@vclab.kaist.ac.kr
James Tompkin[1]
james_tompkin@brown.edu

[1] Brown University, USA
[2] KAIST, South Korea

Abstract

We propose a method to compute depth maps for every sub-aperture image in a light field in a view consistent way. Previous light field depth estimation methods typically estimate a depth map only for the central sub-aperture view, and struggle with view consistent estimation. Our method precisely defines depth edges via EPIs, then we diffuse these edges spatially within the central view. These depth estimates are then propagated to all other views in an occlusion-aware way. Finally, disoccluded regions are completed by diffusion in EPI space. Our method runs efficiently with respect to both other classical and deep learning-based approaches, and achieves competitive quantitative metrics and qualitative performance on both synthetic and real-world light fields.

1 Introduction

Light fields allows high-quality depth estimation of fine detail by aggregating disparity information across many sub-aperture views. This typically results in a depth map for the central view of a light field only. However, in principle we can estimate depth for every pixel in the light field. Some applications require this ability, such as when editing a light field photograph when every output view will be seen on a light field display. Estimating depth reliably for disoccluded regions is difficult because it requires aggregating information from fewer samples. Few existing methods estimate depth for every light field pixel. These are typically computationally expensive [25, 27] or not strictly occlusion aware. Jiang et al. [12, 13] presented the first practical view consistent method based on deep learning.

We present a counterpart first principles method with no learned priors, which produces comparable or better accuracy and view consistency than the current state of the art while being 2–4× faster. Our method is based around estimating accurate view-consistent disparity at edges, and then completing an occlusion-aware diffusion process to fill in missing regions.

Depth diffusion is a long-standing problem in which it is difficult to ensure both consistency and correctness in disoccluded regions. Our key contribution is an angular inpainting method that ensures depth consistency by design, while accounting for the visibility of points in disoccluded regions. In this way, we avoid the problem of trying to constrain or regularize view consistency after estimating depth spatially, and so can maintain efficiency.

Our code will be released as open source software at visual.cs.brown.edu/lightfielddepth.

2 Related Work

The regular structure of an Epipolar Plane Image (EPI) obviates the need for extensive angular regularization and, thus, many light field operations seek to exploit it. Khan et al.'s [14] light field superpixel algorithm operates in EPI space to ensure a view consistent segmentation. The depth information implicit within an EPI is useful for disparity estimation algorithms. Zhang et al. [27] propose an EPI spinning parallelogram operator for this purpose. This operator is similar in respects to the large Prewitt filters of Khan et al. [14] but has a larger support, and provides more accurate estimates. A related method is presented by Tošić and Berkner [22] who create light field scale-depth spaces through convolution with a set of specially adapted kernels. Wang et al. [23, 24] exploit the angular view presented by an EPI to address the problem of occlusion. Tao et al.'s [21] work uses both correspondence and defocus in a higher-dimensional EPI space for depth estimation.

Beyond EPIs, Jeon et al.'s [11] method exploits the relation between defocus and depth too. They shift light field images by small amounts to build a subpixel cost volume. Chuchwara et al. [6] present an efficient light-field depth estimation method based on superpixels and PatchMatch [3] that works well for wide-baseline views. Efficient computation is also addressed by the work of Holynski and Kopf [7]. Their method estimates disparity maps for augmented reality applications in real-time by densifying a sparse set of point depths obtained using a SLAM algorithm. Chen et al. [5] estimate occlusion boundaries with superpixels in the central view of a light field to regularize the depth estimation process.

With deep learning, methods have sought to bypass the large number of images in a light field by learning 'priors' that guide the depth estimation process. Huang et al.'s [9] work can handle an arbitrary number of uncalibrated views. Alperovich et al. [2] showed that an encoder-decoder architecture can be used to perform light field operations like intrinsic decomposition and depth estimation for the central cross-hair of views. As one of the few depth estimation methods that operates on every pixel in a light field, Jiang et al. [12, 13] generate disparity maps for the entire light field and enforce view consistency. However, as their method uses low-rank inpainting to complete disoccluded regions, it fails to account for occluding surfaces in reprojected depth maps. Our method uses occlusion-aware edges to guide the inpainting process and so captures occluding surfaces in off-center views.

3 Occlusion-aware Depth Diffusion

A naive solution to estimate per-view disparity might be to attempt to compute a disparity map for each sub-aperture view separately. However, this is typically challenging for edge views and is highly inefficient, not only in terms of redundant computation but also due to the spatial domain constraints or regularization that must be added to ensure that depth maps are mutually consistent across views. Another simple approach might be to calculate a disparity map for a single view, and then reproject it into all other views. However, this approach fails to handle scene points that are not visible in the single source view. Such points cause holes in the case of disocclusions, or lead to inaccurate disparity estimates when the points lie on an occluding surface. While most methods try to deal with the former case through inpainting, for instance via diffusion, the latter scenario is more difficult to deal with as the occluding surface may have a depth label not seen in the original view. Thus, techniques like diffusion are insufficient on their own without additional guidance.

Our proposed method deals with this issue of depth consistency in subviews of light fields via an occlusion-aware diffusion process. We estimate sparse depth labels at edges that are explicitly defined across views, and then efficiently determine their visibility in each sub-aperture view. Given occlusion-aware edges which persist across views, these edge depth labels can be used as more reliable guides for filling any holes in reprojected views. Since the edge depth labels are not restricted to the source view, we capture any occluding surfaces not visible in the source view. This avoids the aforementioned problem of unseen depth labels. In addition, by performing our inpainting step in the angular rather than the spatial domain of the light field, we improve cross view consistency and occlusion awareness.

Edge Depth & Visibility Estimation To begin, we estimate depth labels at all edges in the light field using the EPI edge detection algorithm proposed by Khan et al. [14]. This filters each EPI with a set of large Prewitt filters, and then estimates a representation of each EPI edge as a parametric line. Lines in EPI space correspond to scene points, with the slope of the line being proportional to the depth of the point. Therefore, this line representation captures both position and disparity information of scene edges.

However, the parametric definition implies the line exists in all views, and, hence, the representation does not contain any visibility information for the point across light field views. We can address this by looking at local structure around the edge for point samples along the line: the point is visible if the local gradient matches the global line direction. We define a gradient-alignment-based visibility score for each sample, then threshold this score to decide which part of the line is actually visible in a given view [15].

Given an EPI line l, we sample it at n locations to obtain a set of point samples $S_l = \{x_i, y_i\}$ on the EPI I. Each sample s_i corresponds to a projection of the original point in light field view i, whose visibility can be determined as:

$$v_l(i) = \mathbb{1}\left(\frac{\nabla I(s_i)(\nabla l)^T}{\|\nabla I(s_i)\|\|\nabla l\|} > cos(\tau_v) \right), \tag{1}$$

where $\mathbb{1}$ denotes the characteristic function of a set beyond the visibility threshold, ∇I is the image gradient, ∇l is the direction perpendicular to the line l, and $\tau = \pi/13$.

The characteristic function was proposed by [15] to generate a central view disparity map, and so only central view points were kept. By extending this idea to the entire light field, we retain all points along with their depth and visibility information, and use them to guide the disparity propagation into disoccluded and occluded regions. Since disparity directly at edges can often be ambiguous due to image resolution limits, the points are offset along the image gradient so that they lie on an actual surface. This is completed using the two-way propagation scheme of [15], which also allows a disparity map for the central view to be generated using dense diffusion of the sparse point depths.

Cross-hair View Projection The EPI line-fitting algorithm works on EPIs in the central cross-hair views—that is, the central row and column of light field images. While it is possible to run it on other rows and columns, this can become expensive, and the central set is usually sufficient to detect visible surfaces in the light field [26]. Hence, we project the estimated disparity map from the center view into all views along the cross-hair. Since gradients at depth edges in the estimated disparity map are not completely sharp, this leads to some edges being projected onto multiple pixels in the target view. We deal with this by sharpening the edges of the disparity map before projection, as shown in Shih et al. [19], using a weighted median filter [16] with parameters $r = 7$ and $\varepsilon = 10^{-6}$. Omitting this step can cause inaccurate estimates around strong depth edges. The result is not very sensitive to parameters r and ε since most parameter settings will target the error-prone strong edges.

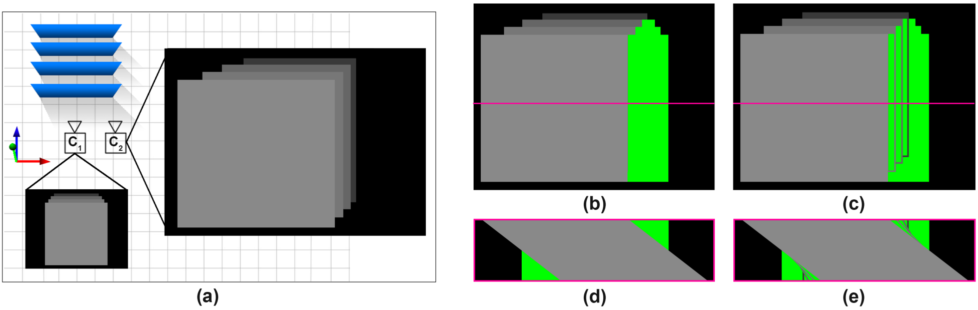


Figure 1: **(a)** Ground-truth disparity maps for a scene from two different camera positions C_1 and C_2. **(b)** Naively attempting to generate the output of C_2 by reprojecting C_1 results in large holes, shown here in green. **(c)** Our method uses depth edges to guide disparity propagation in such disoccluded regions. The EPIs corresponding to the highlighted row are shown in **(d)** and **(e)**. The EPI in **(e)** constitutes our depth EPI D_o.

Angular Inpainting After depth reprojection, we must deal with the two problems highlighted in the overview: inpainting holes, and accounting for occluding surfaces in off-center sub-aperture views. We tackle this by using the edges from Section 3 to guide a dense diffusion process. Moreover, we ensure view consistency by performing diffusion in EPI space.

The EPI lines from the first stage constitute a set L of cross-view edge features. L is robust to occlusions in a single view as it exists in EPI space. As such, L provides occlusion-aware sparse depth labels to guide dense diffusion in EPI space. Diffusion in EPI space has the added advantage of ensuring view consistency.

Let D_o represent an angular slice of the disparity maps with values reprojected from the center view and with propagation guides (Figure 1). Then, we formulate diffusion as a constrained quadratic optimization problem:

$$\hat{D} = \underset{D}{\operatorname{argmin}} \sum_{p \in D} E_d(p) + \sum_{(p,q) \in \mathcal{S}} E_s(p,q), \tag{2}$$

where $\hat{D}$ is the optimal depth labeling of the EPI, and $\mathcal{S}$ is the set of four-connected neighboring pixels. The data $E_d(p)$ and smoothness terms $E_s(p,q)$ are defined as:

$$E_d(p) = \lambda_d(p) \left\| D(p) - D_o(p) \right\|_2^2, \tag{3}$$

$$E_s(p,q) = \lambda_s(p,q) \left\| D(p) - D(q) \right\|_2^2, \tag{4}$$

We take the weight for the smoothness term from the EPI intensity image I:

$$\lambda_s(p,q) = \frac{c}{\|\nabla I(p)\| + \varepsilon}, \tag{5}$$

where $c = 0.1$. We define the weight for the data term as:

$$\lambda_d(p) = \begin{cases} 15 & \text{if } p \in \mathcal{C}, \\ \omega_e(p) & \text{if } p \in \mathcal{L}, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where $\omega_e(p)$ is the edge-importance weight proposed by [15], and $\mathcal{C}$ and $\mathcal{L}$ are the set of pixels coming from the reprojected center view disparity map and EPI line guides, respectively.

Equation (2) defines the optimal disparity map $\hat{D}$ as one that minimizes divergence from the labeled data (Eq. (3)) while being as smooth as possible. Equation (4) measures smoothness as the similarity between disparities of neighboring pixels. We wish to relax the smoothness constraint for edges, so smoothness weight is chosen as the inverse of the image gradient (Eq. (5)). This allows pixels across edges to have a disparity difference without being penalized. The data weight (Eq. (6)) is determined empirically and works for all datasets.

Optimizing Equation (2) is a standard Poisson optimization problem. We solve this using the Locally Adaptive Hierarchical Basis Preconditioning Conjugate Gradient (LAHBPCG) solver [20] by posing the data and smoothness constraints in the gradient domain [4].

Non-cross-hair View Reprojection We now have view-consistent disparity estimates for every pixel in the central cross-hair of light field views: $(u_c, \cdot)$, and $(\cdot, v_c)$. As noted, this set is usually large enough to cover every visible surface in the scene. Hence, all target views (u_i, v_i) outside the cross-hair can be simply computed as the mean of the reprojection of the closest horizontal and vertical cross-hair view $((u_c, v_i)$ and (u_i, v_c), respectively).

4 Experiments

Baseline Methods We compare our results to the state-of-the-art depth estimation methods of Jiang et al. [12] and Shi et al. [18]. Both methods use the deep-learning-based Flownet2.0 [10] network to estimate optical flow between the four corner views of a light field, then use the result to warp a set of anchor views. In addition, Shi et al. further refine the edges of their depth maps using a second neural network trained on synthetic light fields. While Shi et al.'s method generates high-quality depth maps for each sub-aperture view, they do not have any explicit cross-view consistency constraint (unlike Jiang et al.).

Datasets For our evaluation, we used both synthetic and real world light fields with a variety of disparity ranges. For the synthetic light fields, we used the HCI Light Field Benchmark Dataset [8]. This dataset consists of a set of four 9 × 9, 512 × 512 pixels light fields: *Dino*, *Sideboard*, *Cotton*, and *Boxes*. Each has a high-resolution ground-truth disparity map for the central view only. As such, we use this dataset to evaluate the accuracy of depth maps generated by our method and the baseline methods.

For real-world light field data, we use the EPFL MMSPG Light-Field Image Dataset [17] and the New Stanford Light Field Archive [1]. The EPFL light fields are captured with a Lytro Illum and consist of 15 × 15 views of 434 × 625 pixels each. However, as the edge views tend to be noisy, we only use the central 7 × 7 views in our experiments. We show results for the *Bikes* and *Sphynx* scenes. The light fields in the Stanford Archive are captured with a moving camera and have a larger baseline than the Lytro and synthetic scenes. Each scene consists of 17 × 17 views with varying spatial resolution. We use all views from the *Lego* and *Bunny* scenes, scaled down to a spatial resolution of 512 × 512 pixels.

Metrics We evaluate both the accuracy and consistency of the depth maps. For accuracy, we use the mean-squared error (MSE) multiplied by a hundred, and the percentage of *bad pixels*. The latter metric represents the percentage of pixels with an error above a certain threshold. For our experiments we use the error thresholds 0.01, 0.03, and 0.07. The unavailability of ground truth depth maps for the EPFL and Stanford light fields prevents us from presenting accuracy metrics for the real world light fields.

To evaluate view consistency, we reproject the depth maps onto a reference view and compute the variance. Let $\rho_0, \rho_1, \dots, \rho_n$ represent the depth maps for the n light field views warped onto a target view (u,v). The view consistency at pixel s in view (u,v) is given by:

$$\mathcal{C}_{(u,v)}(s) = \frac{1}{n}\sum_{i=0}^{n}(\rho_i(s) - \mu_s)^2, \qquad \text{where } \mu_s = \frac{1}{n}\sum_{i=0}^{n}\rho_i(s), \tag{7}$$

and overall light field consistency is given as the mean over all pixels s in the target view S:

$$\mathcal{C}_{(u,v)} = \frac{1}{S}\sum_{s=0}^{S}\mathcal{C}_{(u,v)}(s). \tag{8}$$

This formulation allows for the consistency to be evaluated quantitatively for both the synthetic and real world light fields.

We also compare computational running time. Our method is implemented in MATLAB except for the C++ Poisson solver. Both baseline methods use the authors' implementations. The learning-based components of both baselines uses TensorFlow, with other components of Jiang et al. in MATLAB. All CPU code ran on an AMD Ryzen ThreadRipper 2950X 16-Core Processor, and GPU code ran on an NVIDIA GeForce RTX 2080Ti.

4.1 Evaluation

Accuracy Table 1 presents quantitative results for the central view of all light fields in accuracy comparisons against ground truth depth. Our method is competitive or better on the MSE metric against the baseline methods, reducing error on average by 20% across the four light fields. However, our method produces more bad pixels than the baseline methods. For baseline techniques to have higher MSE but fewer bad pixels means that they must have larger outliers. This can be confirmed by looking at the error plots in Figure 5.

View Consistency Figure 2 presents results for view consistency across all three datasets. The box plots at the top show that our method has competitive or better view consistency than the baseline methods. As expected, Shi et al.'s method without an explicit view consistency term has significantly larger consistency error. At the bottom of the figure, we visualize how this error is distributed spatially across the views in the light field. Both our method and Jiang et al.'s method produce relatively even distributions of error across views. In our supplemental video, we show view consistency error spatially for each light field view.

Computational Resources Figure 3 presents a scatter plot of runtime versus view consistency across our three datasets. Our method produces comparable or better consistency at a faster runtime, being 2–4× faster than Jiang et al.'s methods per view for equivalent error.

Qualitative Figures 5 and 6 present qualitative single-view depth map results. To assess view consistency, we refer the reader to our supplemental video, which also includes accuracy error and view inconsistency heat map visualizations for all scenes. Overall, all methods produce broadly comparable results, though each method has different characteristics. The learning-based methods tend to produce smoother depths across flat regions. All methods struggle with thin features; our approach fares better with correct surrounding disocclusions (*Boxes* crate; see video). On the *Bunny* scene, our approach introduces fewer background errors and shows fewer 'edging' artifacts than Jiang et al. Shi et al. produces cleaner depth map appearance for *Lego*, but is view inconsistent. Jiang et al. is view consistent, but introduces artifacts on *Lego*. One limitation of our method is on *Sphynx*, where a distant scene and narrow baseline cause noise in our EPI line reconstruction.

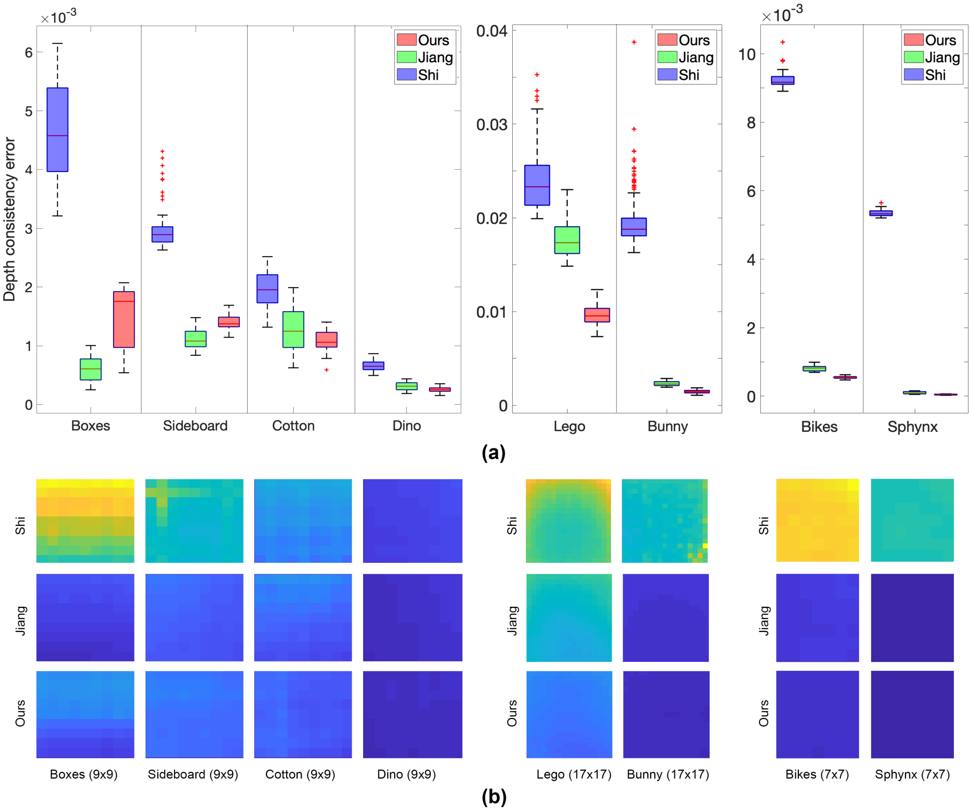


Figure 2: Quantitative view consistency comparison of our method and Jiang et al. [12] and Shi et al. [18]. While the method of Jiang et al. enforces cross view consistency, Shi et al. operates on each view individually and has no explicit consistency constraint. **(a)** For each light field, we plot summary statistics over $\mathcal{C}_{(u,v)}$ for all views (u,v) in the light field (Equation (8)). **(b)** The angular distribution of the error over all views.

Figure 3: Average depth consistency error and runtimes for the three assessed datasets. Our method runs consistently faster than the baselines, while having comparative or better depth consistency. Note that errors across datasets are shown in absolute terms.

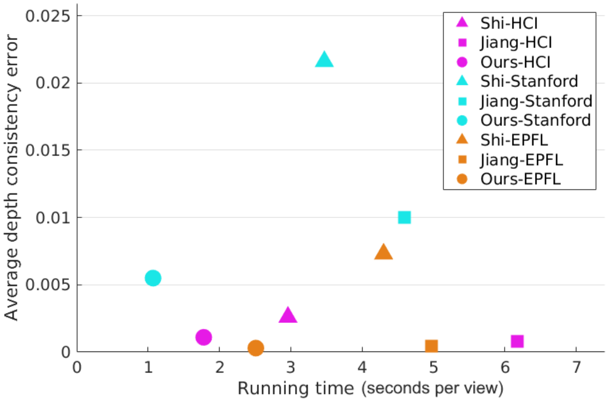
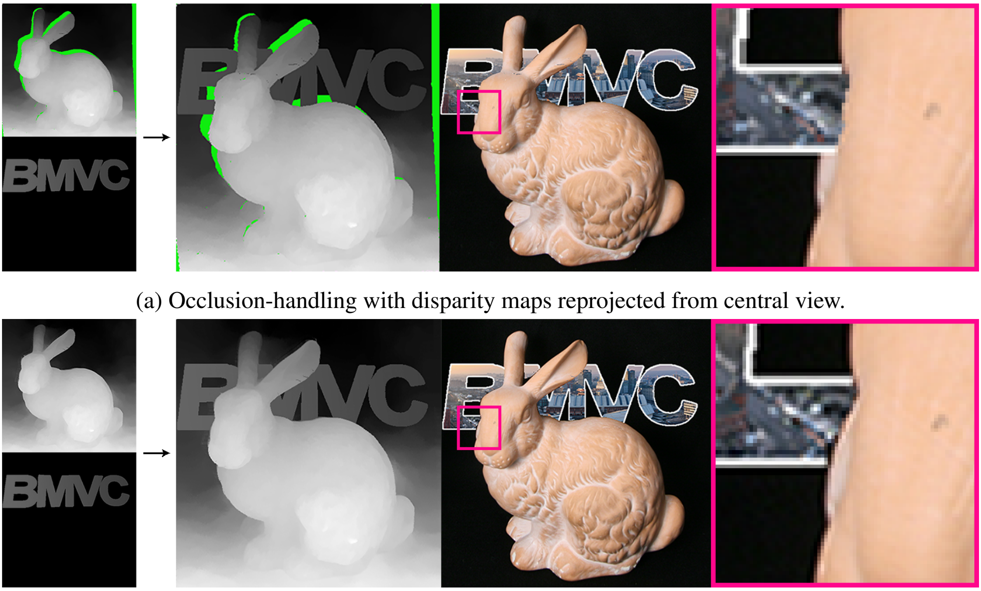


Table 1: Quantitative comparison of the accuracy of our method with two baseline learning-based methods on the central view of four 9 × 9 synthetic light fields. 'BP(x)' is the number of *bad pixels* which fall above threshold x in error. Our method has lower mean-squared error (MSE) than the method of Jiang et al. [12]; our method has competitive MSE to the method of Shi et al. [18] without requiring any learning. As baseline methods have higher MSE but fewer bad pixels, the outliers they do have must be larger. We demonstrate this by visualizing error plots in Figure 5.

Light	MSE * 100			BP1(0.01)			BP2(0.03)			BP3(0.07)		
Field	[18]	[12]	Ours	[18]	[12]	Ours	[18]	[12]	Ours	[18]	[12]	Ours
Sideboard	1.12	1.96	**0.89**	53.0	**47.4**	73.8	20.4	**18.3**	37.36	**2.70**	9.3	16.2
Dino	**0.43**	0.47	0.45	43.0	**29.8**	69.4	13.1	**8.8**	30.8	4.3	**3.6**	10.4
Cotton	0.88	0.97	**0.68**	38.8	**25.4**	56.2	9.6	**6.3**	18.0	2.8	**2.0**	4.9
Boxes	8.48	11.60	**6.70**	66.5	**51.8**	76.8	37.1	**27.0**	47.9	21.9	**18.3**	28.3
Mean	2.72	3.75	**2.18**	50.3	**38.6**	69.0	20.1	**15.1**	33.5	**7.9**	8.3	14.9

Applications Consistent per-view disparity estimates are vital for practical applications such as light field editing: knowing the disparity of regions occluded in the central view allows view-consistent decals or objects to be inserted into the 3D scene (Figure 4). Moreover, without per-view disparity, users can only edit the central view as changes in other views cannot be propagated across views. This limits editing flexibility.



(a) Occlusion-handling with disparity maps reprojected from central view.

(b) Occlusion-handling with per-view disparity estimates.

Figure 4: With consistent per-view disparity estimates, objects can be inserted into the 3D scene with accurate occlusions. **(a)** Since the right cheek of the bunny is not visible in the central view, a simple reprojection of the disparity from the central view fails to handle occlusion correctly for the inserted object. The green regions denote holes in the reprojection. **(b)** With per-view disparity, the object can be placed correctly in all views.

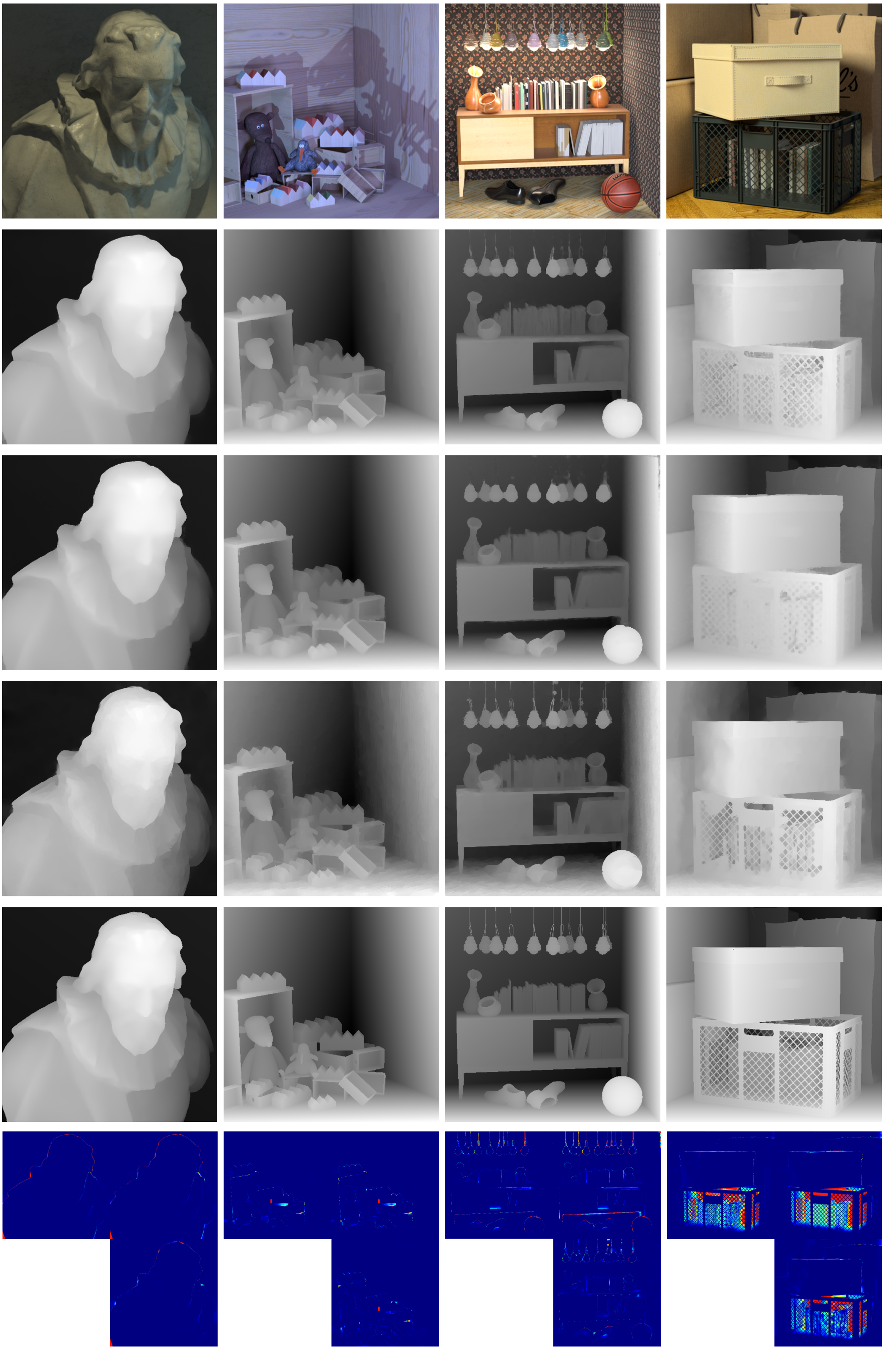


Figure 5: *HCI* dataset. Top to bottom: light field central view, Shi et al. [18], Jiang et al. [12], our method, ground truth depth, error maps in clockwise order (Shi, Jiang, Ours). In general, our method has a lower mean squared error (MSE) with fewer large outliers (please zoom into error maps), captures thin features better, and generates more view-consistent depth maps. However, their depth maps are more geometrically accurate more often (lower bad pixel percentages) and less sensitive to variations in image texture.

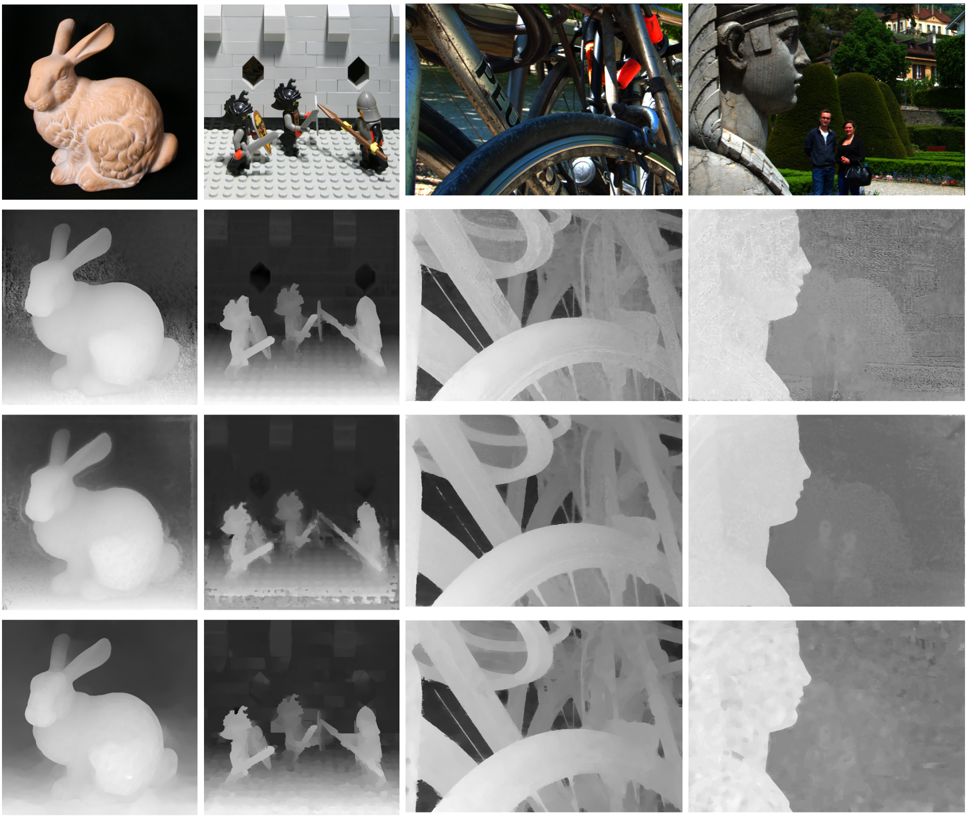


Figure 6: Real-world light fields from the Stanford (*left pair*) and EPFL (*right pair*) datasets. *Top to bottom:* central RGB view, Shi et al. [18], Jiang et al. [12], and our method. While our method has more bad pixels and can be sensitive in narrow baseline cases (*far right:* limitation Sphynx case), in general our method has equivalent or lower view consistency error, runs faster, and has no training data or pre-trained network dependency.

5 Discussion and Conclusion

Our work demonstrates that careful handling of depth edge estimation, occlusions, and view consistency can produce per-view disparity maps with comparable performance to state of the art learning-based methods in terms of average accuracy and view consistency. This can also lead to computation time performance gains.

Nonetheless, our method does have limitations; one area where our lack of explicit (u,v) regularization is sometimes a factor is in spatial edge consistency, e.g., for diagonal angles in non-cross-hair views. Adding additional regularization begins another trade-off between smoothness, accuracy, and computational cost. Further, our method is limited when an EPI contains an area enclosed by high gradient boundaries with no data term depth value in it, and disocclusion propagation can be attributed either to the foreground occluder or revealed background (e.g., *Lego* scene, arm of right-most character).

Accurate edge estimation and occlusion reasoning is still a core problem, and greater context may help. Future work might apply small targeted deep-learned priors for efficiency.

References

[1] The New Stanford Light Field Archive, 2008. URL http://lightfield.stanford.edu/.

[2] A. Alperovich, O. Johannsen, and B. Goldluecke. Intrinsic light field decomposition and disparity estimation with a deep encoder-decoder network. In *26th European Signal Processing Conference (EUSIPCO)*, 2018.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[4] Pravin Bhat, Larry Zitnick, Michael Cohen, and Brian Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. In *ACM Transactions on Graphics (TOG)*, 2009.

[5] Jie Chen, Junhui Hou, Yun Ni, and Lap-Pui Chau. Accurate light field depth estimation with superpixel regularization over partially occluded regions. *IEEE Transactions on Image Processing*, 27(10):4889–4900, 2018.

[6] A. Chuchvara, A. Barsi, and A. Gotchev. Fast and accurate depth estimation from sparse light fields. *IEEE Transactions on Image Processing (TIP)*, 29:2492–2506, 2020.

[7] Aleksander Holynski and Johannes Kopf. Fast depth densification for occlusion-aware augmented reality. In *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, volume 37. ACM, 2018.

[8] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*, pages 19–34. Springer, 2016.

[9] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018.

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[11] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1555, 2015.

[12] Xiaoran Jiang, Mikaël Le Pendu, and Christine Guillemot. Depth estimation with occlusion handling from a sparse set of light field views. In *25th IEEE International Conference on Image Processing (ICIP)*, pages 634–638. IEEE, 2018.

[13] Xiaoran Jiang, Jinglei Shi, and Christine Guillemot. 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.

[14] Numair Khan, Qian Zhang, Lucas Kasser, Henry Stone, Min H. Kim, and James Tompkin. View-consistent 4D light field superpixel segmentation. In *International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[15] Numair Khan, Min H. Kim, and James Tompkin. Fast and accurate 4D light field depth estimation. Technical Report CS-20-01, Brown University, 2020.

[16] Ziyang Ma, Kaiming He, Yichen Wei, Jian Sun, and Enhua Wu. Constant time weighted median filtering for stereo matching and beyond. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 49–56, 2013.

[17] Martin Rerabek and Touradj Ebrahimi. New light field image dataset. In *8th International Conference on Quality of Multimedia Experience (QoMEX)*, number CONF, 2016.

[18] Jinglei Shi, Xiaoran Jiang, and Christine Guillemot. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Transactions on Image Processing (TIP)*, 28(12):5867–5880, 2019.

[19] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Richard Szeliski. Locally adapted hierarchical basis preconditioning. In *ACM SIGGRAPH 2006 Papers*, pages 1135–1143. 2006.

[21] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 673–680, 2013.

[22] Ivana Tosic and Kathrin Berkner. Light field scale-depth space transform for dense depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 435–442, 2014.

[23] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3487–3495, 2015.

[24] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Depth estimation with occlusion modeling using light-field cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(11):2170–2181, 2016.

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

[26] Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619, 2013.

[27] Shuo Zhang, Hao Sheng, Chao Li, Jun Zhang, and Zhang Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159, 2016.