Occlusion-Aware Cost Constructor for Light Field Depth Estimation

Yingqian Wang[1], Longguang Wang[1], Zhengyu Liang[1], Jungang Yang[1]✉, Wei An[1], Yulan Guo[1]
[1]National University of Defense Technology

https://github.com/YingqianWang/OACC-Net

Abstract

Matching cost construction is a key step in light field (LF) depth estimation, but was rarely studied in the deep learning era. Recent deep learning-based LF depth estimation methods construct matching cost by sequentially shifting each sub-aperture image (SAI) with a series of predefined offsets, which is complex and time-consuming. In this paper, we propose a simple and fast cost constructor to construct matching cost for LF depth estimation. Our cost constructor is composed by a series of convolutions with specifically designed dilation rates. By applying our cost constructor to SAI arrays, pixels under predefined disparities can be integrated and matching cost can be constructed without using any shifting operation. More importantly, the proposed cost constructor is occlusion-aware and can handle occlusions by dynamically modulating pixels from different views. Based on the proposed cost constructor, we develop a deep network for LF depth estimation. Our network ranks first on the commonly used 4D LF benchmark in terms of the mean square error (MSE), and achieves a faster running time than other state-of-the-art methods.

1. Introduction

Light field (LF) cameras can encode 3D scenes into 4D LF images. By using the abundant spatial and angular information in the LF images, the scene depth can be obtained by performing LF depth estimation. As a fundamental task in LF image processing, depth estimation benefits many subsequent applications such as refocusing [33], view synthesis [3, 13], 3D reconstruction [14], and virtual reality [37].

With the advances of deep neural networks, many deep learning-based methods [2, 6, 7, 9, 17, 19, 26, 29] have been proposed and boosted the performance of LF depth estimation. Recent deep learning-based methods achieve LF depth estimation in a four-step pipeline including feature extraction, cost construction, cost aggregation, and depth regression. To achieve higher accuracy, these methods designed different modules for feature extraction [29] and cost aggregation [2, 9]. However, as a key step in LF depth estimation, matching cost construction was rarely studied.

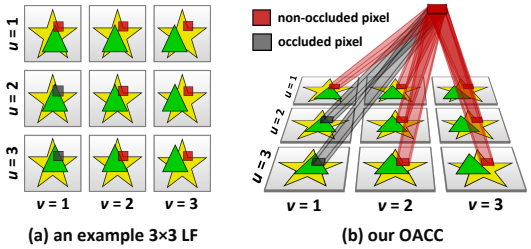

Figure 1. An illustration of the proposed occlusion-aware cost constructor (OACC). (a) A toy example of a 3×3 LF, in which the yellow star is partially occluded by the green triangle. (b) Our OACC constructs matching cost via convolutions and can handle occlusions by assigning smaller weights to occluded pixels.

To construct matching costs for LF depth estimation, existing methods [2, 29] shift each sub-aperture image (SAI) with a series of predefined offsets, and then concatenate the shifted SAIs to form a cost volume. Although this *shift-and-concat* scheme is easy to implement, the large number of shifting operation[1] reduces the efficiency of these methods. Moreover, during matching cost construction, pixels at different spatial locations are processed equally, which cannot handle the spatially-varying occlusions where some views are less informative and can even deteriorate the estimation results.

To handle the aforementioned challenges, in this paper, we propose an occlusion-aware cost constructor (OACC) for LF depth estimation. Our cost constructor is composed by a series of convolutions with specifically designed dilation rates. By applying our OACC to SAI arrays, pixels under predefined disparities can be integrated without performing shifting operation. More importantly, our OACC can handle occlusions by dynamically modulating pixels from different views, as shown in Fig. 1. Based on the proposed OACC, we develop a deep network for LF depth estimation. Our network achieves state-of-the-art depth estimation accuracy with a significant acceleration.

The contributions of this paper can be summarized as:

[1]For example, in *LFAttNet* [29], totally 80 views are shifted by 8 disparity levels, resulting in 640 sequential shifting operations.

- We propose a cost constructor to replace the *shift-and-concat* approach for matching cost construction.
- We make our cost constructor to be occlusion-aware by modulating pixels from different views in a fine-grained manner.
- We develope an OACC-Net for LF depth estimation. Our method achieves top accuracy with significant acceleration as compared to other state-of-the-art methods on the 4D LF benchmark [8].

2. Related Works

In this section, we review the major works in LF depth estimation. We classify the existing methods into traditional methods and deep learning-based methods.

2.1. Traditional Methods

Early works on LF depth estimation follow the traditional paradigm and use different approaches to measure the consistency among different views. Tao et al. [27] proposed to combine the correspondence cue and the defocus cue for LF depth estimation. Subsequently, Tao et al. [28] introduced a shading-based refinement approach to improve the depth estimation accuracy. Jeon et al. [11] proposed a phase-based multi-view stereo matching method and achieved depth estimation in the Fourier domain. Wang et al. [31] considered occlusions in LF depth estimation and proposed an occlusion-aware algorithm based on the partial angular consistency. Williem et al. [36] proposed angular entropy cost and adaptive defocus cost to handle the noise and occlusion issues for depth estimation. More recently, Han et al. [4] proposed an occlusion-aware vote cost to preserve edges in depth maps.

Since an epipolar plane image (EPI) contains patterns of oriented lines and the slope of these lines is related to the depth values, many methods achieve depth estimation by analyzing the slope of each line on EPIs. Wanner et al. [34] proposed a structure tensor to estimate the slope of lines in horizontal and vertical EPIs, and refined the initial results by global optimization. Zhang et al. [38] proposed a spinning parallelogram operator (SPO) to estimate the slopes for depth estimation. Sheng et al. [25] proposed to estimate slopes using multi-orientation EPIs and achieved improved results over SPO. Schilling et al. [24] proposed an inline occlusion handling scheme operated on EPIs to achieve state-of-the-art depth estimation performance among traditional methods.

2.2. Deep Learning-based Methods

Recently, deep networks have been widely used for depth estimation and achieved significant performance gain over traditional methods. Heber et al. [6] proposed the first end-to-end network to learn the mapping between a 4D LF and its corresponding depths. Subsequently, Heber et al. [7] proposed a U-shaped network with 3D convolutions to extract geometric information from LFs for depth estimation. Shin et al. [26] proposed a multi-stream network and a series of data augmentation approaches for fast and accurate LF depth estimation. Tsai et al. [29] proposed an attention-based view selection network to adaptively incorporate all angular views for depth estimation. Peng et al. [21] proposed an unsupervised LF depth estimation method that can be trained without using the ground-truth depth maps. Subsequently, Peng et al. [22] proposed a zero-shot learning-based method that can perform unsupervised depth estimation without using external datasets. More recently, Chen et al. [2] proposed an attention-based multi-level fusion network to handle the occlusion problem for depth estimation. Huang et al. [9] proposed a multi-disparity-scale cost aggregation approach to achieve fast LF depth estimation.

Different from existing methods which focus on designing advanced modules for feature extraction [29] or cost aggregation [2,9], in this paper, we study the cost construction stage and propose a simple and efficient module to achieve occlusion-aware cost construction.

3. Method

In this section, we first describe the LF structure and analyze the influence of occlusions to angular consistency. Then, we introduce the proposed occlusion-aware cost constructor. Finally, we introduce our network for LF depth estimation.

3.1. LF Structure and Occlusion Analysis

We use the two-plane model [18] to parameterize LFs. As shown in Fig. 2, a light ray originated from point P can be uniquely determined by its intersections across the camera plane $\Omega = \{(u, v)\}$ and the image plane $\Pi = \{(h, w)\}$. Consequently, an LF can be formulated as a 4D tensor according to the mapping function $\mathcal{L}(u, v, h, w) : \Omega \times \Pi \mapsto \mathbb{R}$. In this paper, we denote the 4D LF as $\mathcal{L} \in \mathbb{R}^{U \times V \times H \times W}$, where U and V represent the angular dimensions, and H and W represent the spatial dimensions.

Note that, a scene point (e.g., point P in Fig. 2) will be projected to different locations on the images captured by different cameras. As shown in Fig. 2(b), there is a disparity (denoted as d) between projections P_1 and P_2, and the depth value of P can be calculated according to $\gamma = fB/d$, where B and f represent the baseline length and the focal length of the LF camera. Consequently, depth estimation can be achieved by estimating the per-pixel disparities of the LF images. In this paper, we follow [2,9,26,29] to estimate the disparity map of the center view.

Since the projections of a scene point on different views should have identical intensity under Lambertian and non-occlusion assumptions, depth estimation can be achieved by

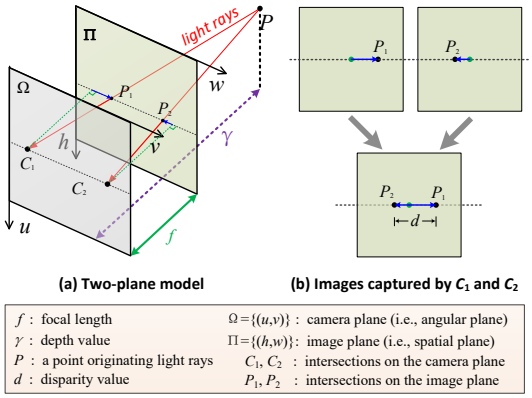

Figure 2. An illustration of the two-plane model.

choosing the disparity candidate with highest angular consistency. To compare the angular consistency under different disparities, we construct angular patches according to

$$\mathcal{A}_d^p(u,v) = \mathcal{L}\left(u, v, h + (u_c - u)d, w + (v_c - v)d\right), \quad (1)$$

where A_d^p is the angular patch at pixel $p(h,w)$ with disparity d, and u_c and v_c represent the angular coordinates of the center view.

Here, we use an example for illustration. As shown in Fig. 3(a), we select two pixels from scene *sideboard* [8], and construct angular patches under different disparities. The angular patches of pixel **A** are shown in Fig. 3(c), from which we can see that the color of the pixels in the angular patch is more consistent near groundtruth disparity value (i.e., $d = 1.13$). We further calculate the standard deviation of these angular patches to evaluate their consistency. As shown in Fig. 3(e), the curve reaches its minimum at the groundtruth disparity. That is, the intensity of pixels in an angular patch is most consistent under correct disparities.

However, this theory does not hold in occluded regions. As shown in Fig. 3(d), we construct angular patches of pixel **B** under different disparities. It can be observed that pixels on the top-left corner of the angular patch have different color from other pixels even under the groundtruth disparity (i.e., $d = 0.31$). That is because, the corresponding pixels in the top-left views are occluded by the foreground basketball. Consequently, occlusions can deteriorate the angular consistency under correct disparities, making the standard deviation curve (the red curve in Fig. 3(f)) not reach its minimum at the groundtruth disparity.

It is interesting that when we mask the top-left pixels in each angular patch of pixel **B** and calculate the standard deviation of the remaining pixels, the modified curve (the pink curve in Fig. 3(f)) reaches its minimum near the groundtruth disparity. It demonstrates that the intensity of non-occluded pixels in an angular patch are most consistent at correct disparities. Motivated by this observation, we design an occlusion-aware cost constructor to handle the occlusion issue for matching cost construction.

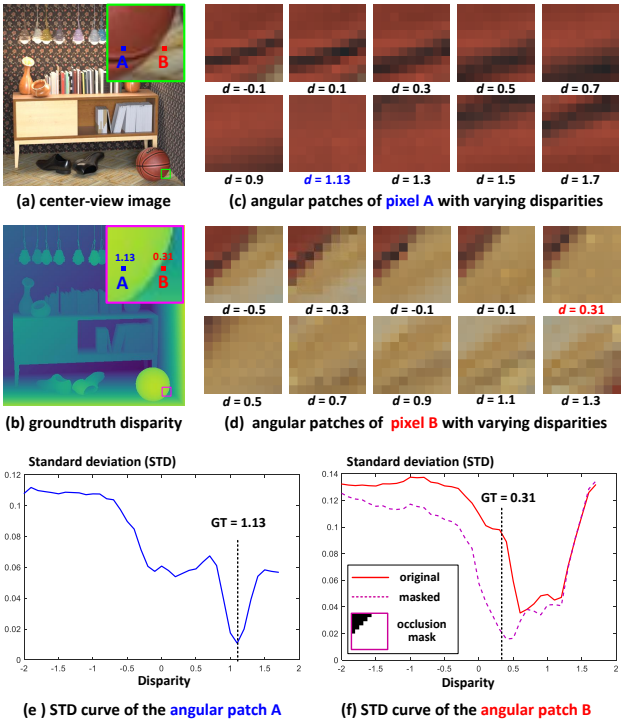

Figure 3. Comparison of the angular consistency in occluded and non-occluded regions. Standard deviation is used to evaluate the consistency of pixels in an angular patch. For the occluded region, an occlusion mask is needed to make the standard deviation reach its minimum at the correct disparity.

3.2. Occlusion-Aware Cost Constructor

Given a 5D LF feature $\mathcal{L} \in \mathbb{R}^{U\times V\times H\times W\times C}$ and a candidate disparity $d \in \{d_{min}, \cdots, d_{max}\}$, existing methods construct matching costs using the *shift-and-concat* approach. Specifically, they shift the feature of each view according to its angular coordinate (u,v) and the given disparity d, then concatenate all the shifted features to generate the cost tensor. That is,

$$\mathcal{F}_{u,v}^d(h,w,:) = \mathcal{F}_{u,v}\left(h + (u_c - u)d, w + (v_c - v)d, :\right), \quad (2)$$

$$\mathcal{C}_d = Concat\left(\mathcal{F}_{0,0}^d, \cdots, \mathcal{F}_{U,V}^d\right), \quad (3)$$

where $\mathcal{F}_{u,v}^d \in \mathbb{R}^{H\times W\times C}$ denotes the shifted feature of view (u,v) under disparity d, and $\mathcal{C}_d \in \mathbb{R}^{H\times W\times UVC}$ denotes the cost tensor under disparity d. Finally, the cost volume is generated by stacking all the cost tensors (i.e., $\mathcal{C}_{d_{min}}, \cdots, \mathcal{C}_{d_{max}}$) along the disparity dimension.

Instead of using the *shift-and-concat* approach, in this paper, we propose an occlusion-aware cost constructor for matching cost construction. The main ideas of our OACC

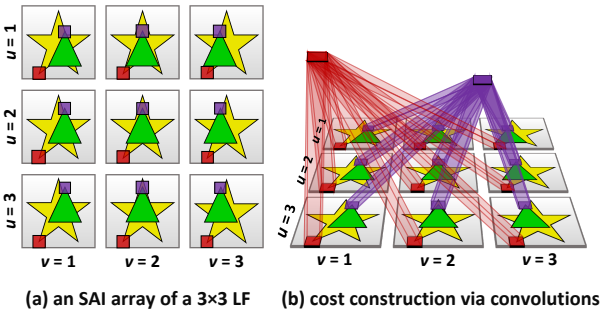

Figure 4. An illustration of our cost construction process.

are: 1) using convolutions to integrate pixels of each view under specific disparities; 2) modulating the input pixels to handle occlusions during cost construction.

3.2.1 Cost Construction via Convolutions

To construct matching cost, pixels in the angular patch under each candidate disparity should be integrated respectively. However, how to efficiently find the corresponding pixels to form an angular patch remains challenging. Since LF images have a regular spatial-angular structure [32], cost construction can be achieved by performing convolutions on the SAI array.

Here, we use a toy example for illustration. As shown in Fig. 4(a), a 3×3 LF is organized in an array of SAIs. In this scenario, the yellow star is in the background with zero disparity, and the green triangle is in the foreground and has a positive disparity[2]. The bottom-left vertex of the yellow star and the top vertex of the green triangle are marked by a red box and a purple box, respectively. It can be observed that both red and purple boxes are evenly distributed in a square region, which can be easily integrated via convolutions. Consequently, we design our cost constructor as a series of convolutions with a kernel size of $U \times V$ and different dilation rates to integrate angular patches under different disparities. The dilation rate is closely related to the preset disparity d and can be calculated according to

$$dila(d) = [H - d, W - d], \tag{4}$$

where H and W denote the height and width of each SAI, respectively. From Eq. 4, we can conclude that object with a larger disparity value (e.g., the purple box) has a smaller dilation rate, which is consistent with the toy example in Fig. 4. Using our proposed cost constructor, angular patches under different disparities can be integrated without performing any shifting operation, and the matching cost can be efficiently constructed by convolving all the pixels in the angular patch, as shown in Fig. 4(b).

[2]Under a positive disparity, object in the left/upper views locates at right/lower positions.

In our implementation, zero-padding is performed on each SAI to avoid aliasing among different views at boundaries. Moreover, the boundary of the resulting features are cropped to ensure the output features have a spatial resolution of $H \times W$. Details of padding and cropping strategies can be referred to the supplemental material.

3.2.2 Occlusion Handling via Pixel Modulation

As analyzed in Section 3.1, occlusions can deteriorate the angular consistency and should be masked out during cost construction. Inspired by *Deformable ConvNet V2* [40], we introduce a modulation mechanism to dynamically adjust the amplitude of pixels from different views for occlusion handling. Specifically, given an angular patch[3] $\mathcal{A}_d^p \in \mathbb{R}^{U \times V}$ at spatial location $p{=}(h, w)$ under disparity d, the modulated convolution can be formulated as

$$y(p, d) = \frac{\sum_{k=1}^{UV} \omega_k \cdot \mathcal{A}_d^p(k) \cdot \Delta m_k^p}{\sum_{k=1}^{UV} \Delta m_k^p}, \tag{5}$$

where $y(p)$ denotes the resulting matching cost at spatial location p under disparity d, ω_k denotes the weight of our cost constructor at the k^{th} sampling point, and $\Delta m_k^p \in [0, 1]$ is the modulation scalar. Note that, the weights of our cost constructor are shared across different spatial locations and disparity values, while the modulation scalar is spatially varying and only shared among different disparity values.

With the modulation mechanism, our cost constructor can adjust the contributions of each view at each location to achieve occlusion-aware cost construction. For example, if a scene point is occluded in some views, our OACC will assign small modulation scalars to the occluded pixels to reduce their impact on the matching cost. It is demonstrated in Sec. 4.2 that the modulation mechanism is crucial for accurate depth estimation.

3.2.3 Occlusion Mask Generation

To achieve occlusion-aware cost construction, the occlusion mask of each view need to be calculated to generate reasonable modulation scalars in Eq. 5. However, accurate occlusion estimation is a non-trivial task. Inspired by the unsupervised LF depth estimation methods [12, 21, 22, 39], in this paper, we propose a parameter-free approach to deduce occlusion mask of each view. Specifically, for regions with occlusions, a scene point that is available in the center view can be unavailable in the surrounding views, and the occluded pixels in these surrounding views cannot find their corresponding pixels in the center view. Consequently, the

[3]Our modulated convolution is applied to the SAI arrays as in Fig. 4. For simplicity, we use the angular patch to denote the pixels from each view under a specific disparity, and ignore the dilation rates while performing our modulated convolution.

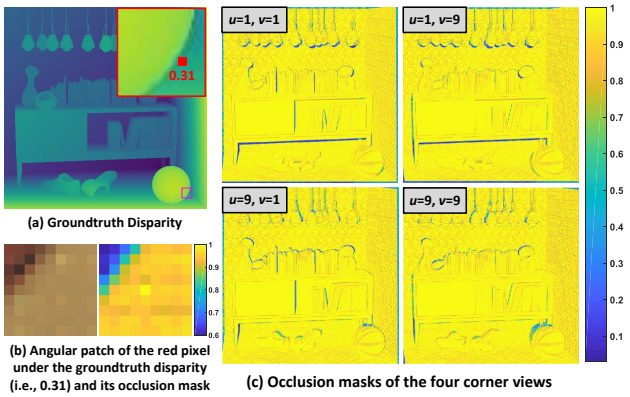

Figure 5. Visualization of the generated occlusion masks on scene *sideboard*. Lower values represent heavier occlusions.

fine-grained occlusion mask can be calculated based on the photometric consistency prior.

Denote the disparity map of the center view as $\mathcal{D}_c$, the surrounding views are firstly warped to the center view, i.e.,

$$\mathcal{I}_{k\to c} = W_{k\to c}^{\mathcal{D}_c}(\mathcal{I}_k), \quad k = 1, 2, \cdots, UV, \tag{6}$$

where $W_{k\to c}^{\mathcal{D}_c}$ denotes the warping operation that projects the k^{th} view $\mathcal{I}_k$ to the center view $\mathcal{I}_c$. Assume that the disparity map $\mathcal{D}_c$ is accurate, the projected view $\mathcal{I}_{k\to c}$ should have identical values to the center view $\mathcal{I}_c$ at non-occluded regions. Therefore, we use the absolute residuals between $\mathcal{I}_{k\to c}$ and $\mathcal{I}_c$ to measure the photometric consistency, i.e.,

$$\mathcal{I}_{k\to c}^{res} = |\mathcal{I}_{k\to c} - \mathcal{I}_c|. \tag{7}$$

Finally, the occlusion mask of the k^{th} view is obtained by re-mapping $\mathcal{I}_{k\to c}^{res}$ to $[0, 1]$, i.e.,

$$\mathcal{M}_k = |1 - \mathcal{I}_{k\to c}^{res}|^q, \tag{8}$$

where q is a scalar that controls the decaying rate. A larger q can enhance the sensitivity to occlusions but degrade the robustness to noise (see Sec 4.2). In our implementation, we empirically set q=2 to achieve a good trade-off between occlusion awareness and noise robustness.

Using the aforementioned approach, the occlusion mask (i.e., the modulation scalars in Eq. 5) of each view can be obtained. As shown in Fig. 5, the generated occlusion masks are reasonable and consistent to the real case.

3.3. Network Design

Based on the proposed OACC, we develop a deep network called OACC-Net for LF depth estimation. As shown in Fig. 6, our network takes a $U\times V$ LF as its input and sequentially performs feature extraction, OACC-based cost construction, cost aggregation, and depth regression.

3.3.1 Feature Extraction

As shown in Fig. 6(b), in our feature extraction module, a 3×3 convolution is first used to extract initial features. Then, eight residual blocks [5] are cascaded for deep feature extraction. These residual blocks are built in a "Conv-BN-LeakyReLU-Conv-BN" structure with skip connections for local residual learning. Features generated by the last residual block are further fed to three cascaded 3×3 convolutions to generate features for cost construction. Note that, the weights of all the convolutions in our feature extraction module are shared among different views.

3.3.2 Cost Construction

We use the proposed OACC for matching cost construction. As described in Sec. 3.2.1, features generated by the feature extraction module are organized into SAI arrays to form the input of our OACC. Besides, the occlusion mask of each view is generated for occlusion-aware pixel modulation. However, there is a "chicken-and-egg" problem between occlusion mask generation and depth estimation. That is, occlusion mask generation requires a disparity map as its input, but the disparity information is unavailable at this stage.

Here, we propose an iterative scheme to solve this problem. In the testing phase, we generate an initial occlusion mask by setting all its elements to one (i.e., non-occlusion assumption), and use this initial mask for depth estimation. After obtaining the initial disparity map, we update the occlusion mask and use the updated mask to generate more accurate disparity maps. It is demonstrated in Sec. 4.2 that the proposed iterative scheme works well and can make occlusion prediction and depth estimation mutually boost. In the training stage, we directly use the groundtruth disparity map to generate occlusion masks to avoid training collapse.

3.3.3 Cost Aggregation and Regression

Given the cost volume generated by our OACC, we first use a 1×1 convolution to reduce its channel depth from 512 to 160. Then we cascade eight 3D convolutions with a kernel size of 3×3×3 for cost aggregation. The third to the sixth 3D convolutions are organized into two residual blocks, and channel attention layers are adopted after each residual block to highlight contributive channels. Finally, a 3D tensor $\mathcal{F}_{final} \in \mathbb{R}^{D\times H\times W}$ is generated by the last 3D convolution, and the disparity is regressed according to

$$\hat{\mathcal{D}}_c = \sum_{d_k=d_{min}}^{d_{max}} d_k \times Softmax(\mathcal{F}_{final}), \tag{9}$$

where $\hat{\mathcal{D}}_c$ denotes the estimated center-view disparity, $Softmax(\cdot)$ denotes the softmax normalization which is performed along the disparity axis of $\mathcal{F}_{final}$.

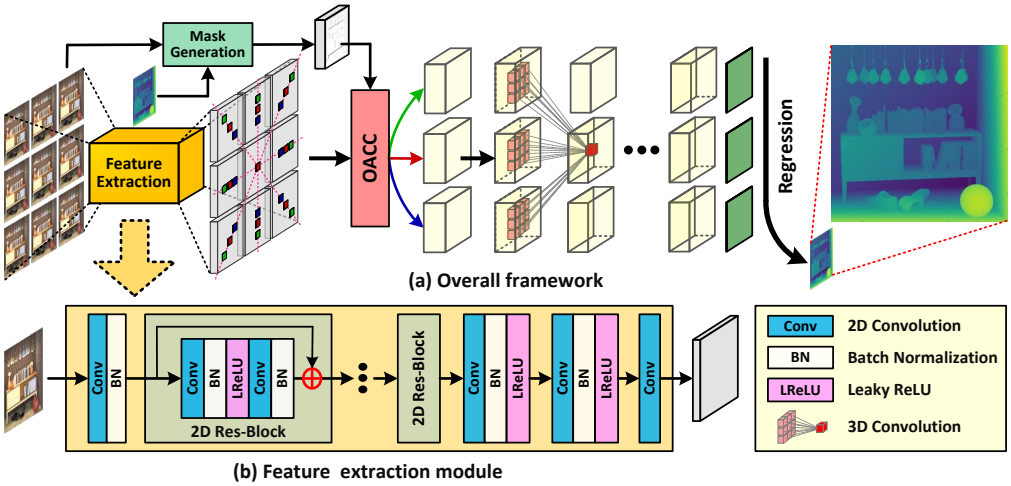

Figure 6. An overview of our OACC-Net. Here, a 3×3 LF is used as an example for illustration.

4. Experiments

In this section, we first introduce the datasets and implementation details, then conduct experiments to investigate our models. Finally, we compare our OACC-Net to several state-of-the-art LF depth estimation methods.

4.1. Datasets and Implementation Details

We used the 4D LF benchmark [8] to validate the effectiveness of our method. All LFs in this benchmark have an angular resolution of 9×9 and a spatial resolution of 512×512. All the 9×9 views are used by our method for depth estimation. We followed [2, 9, 26, 29] to use 16 scenes in the "Additional" category for training, 8 scenes in the "Stratified" and "Training" categories for validation, and 4 scenes in the "Test" category for test. We also used other LF datasets [16, 23, 30, 35] to test the generalization capability of our method (see Fig. 10 and the supplemental material).

During the training phase, we randomly cropped SAIs into patches of size 48×48, and converted them into gray-scale images. We performed a large variety of data augmentation, including random flipping and rotation, brightness and contrast adjustment, noise injection, refocusing, and downsampling. Our OACC-Net was trained in a supervised manner with an L1 loss, and was optimized using the Adam method [15] with β_1=0.9, β_2=0.999. The batch size was set to 16 and the learning rate was set to 1×10^{-3}. The training was stopped after 3×10^5 iterations and takes about 7 days. Our model was implemented in PyTorch and trained on a PC with two Nvidia RTX 2080Ti GPUs.

We used the mean square error (MSE) and bad pixel ratio (BadPix(ϵ)) as quantitative metrics for performance evaluation. BadPix(ϵ) measures the percentage of incorrectly estimated pixels whose absolute errors exceeding a predefined threshold (e.g., ϵ =0.07, 0.03, 0.01).

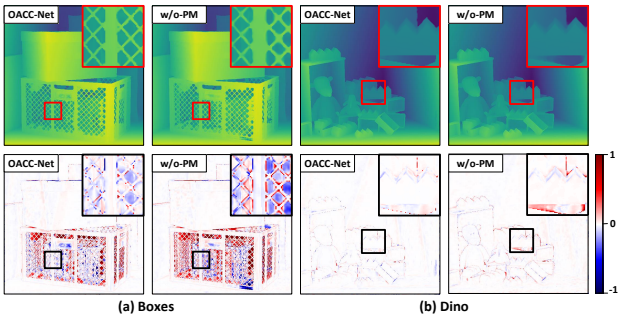

Figure 7. Visual comparisons of our method on scenes (a) *boxes* and (b) *dino* with/without using pixel modulation mechanism. Top-row figures show the estimated disparity $\hat{\mathcal{D}}$ while the bottom-row figures show the corresponding error maps ($\hat{\mathcal{D}} - \mathcal{D}_{gt}$).

4.2. Model Analyses

We first conduct experiment to validate the effectiveness of our pixel modulation mechanism. Then, we test the performance of our method with different number of iterations and decaying rates. Finally, we demonstrate the efficiency of our OACC.

Pixel modulation mechanism. We replaced the modulated convolution with a vanilla convolution and retrained this variant from scratch. As shown in Table 1, compared to our OACC-Net (i.e., "iter_2"), model "w/o-PM" suffers a 1.513 and 0.336 increase in BadPix0.07 and MSE, respectively. That is because, without using the pixel modulation mechanism, pixels from each view and at each spatial location are processed equally thus the occlusion issue cannot be handled. The qualitative results in Fig. 7 also demonstrate that the disparities predicted by our OACC-Net are more accurate at regions with heavy occlusions.

Number of iterations. We compare the performance of our method with different number of iterations (see Sec 3.3.2). As shown in Table 1, directly using an all-one ten-

Table 1. BadPix0.07 (BP07) and mean square error (MSE) achieved by different variants of our OACC-Net on the 4D LF benchmark [8]. "*w/o-PM*" denotes the model trained without using the pixel modulation mechanism, "iter_k" denotes the model performing k iterations in the inference stage, and "*gt-mask*" denotes the model using occlusion masks generated by the groundtruth disparities. The main model of our OACC-Net (i.e., *iter_2*) is highlighted. The best results are in red and the second best results are in blue.

Models	*Backgammon*		*Dots*		*Pyramids*		*Stripes*		*Boxes*		*Cotton*		*Dino*		*Sideboard*		**Average**	
	BP07	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*	*BP07*	*MSE*
w/o-PM	7.142	5.097	2.419	1.382	0.269	0.008	5.909	1.084	14.17	4.023	0.550	0.174	1.649	0.133	3.846	0.674	4.494	1.572
iter_1	4.128	4.059	1.762	1.395	0.156	0.005	3.444	0.879	10.84	3.182	0.352	0.172	1.099	0.091	3.366	0.562	3.143	1.293
iter_2	3.931	3.938	1.510	1.418	0.157	0.004	2.920	0.845	10.70	2.892	0.312	0.162	0.967	0.083	3.350	0.542	2.981	1.236
iter_3	3.928	3.858	1.578	1.421	0.158	0.005	2.920	0.847	10.56	2.968	0.319	0.161	0.989	0.082	3.314	0.581	2.970	1.240
iter_4	3.918	3.824	1.572	1.424	0.158	0.005	2.909	0.846	10.60	2.966	0.310	0.159	0.999	0.082	3.337	0.591	2.975	1.237
gt-mask	3.910	3.593	1.515	1.297	0.156	0.005	2.876	0.840	10.01	2.278	0.300	0.118	0.909	0.072	3.057	0.502	2.842	1.088

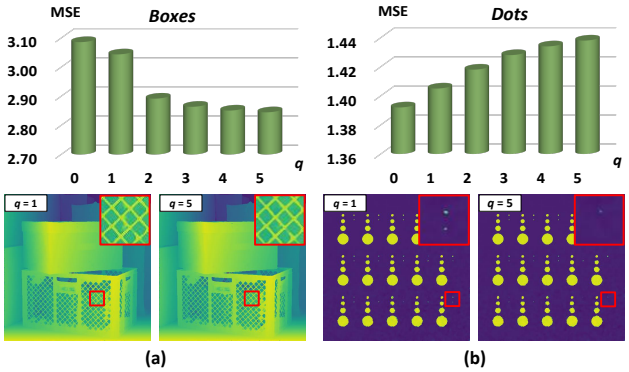

Figure 8. Results achieved by our method on scenes (a) *boxes* and (b) *dots* with different decaying rates q. Top figures show the MSE *w.r.t.* different decaying rates, and the bottom figures show the disparity maps with q=1 and q=5.

sor as an occlusion mask (i.e., "iter_1") achieves 3.143 and 1.293 in terms of average BadPixel0.07 and average MSE, respectively. It demonstrates the importance of occlusion mask generation in our method. When using the initial estimated disparity to generate occlusion masks, the average BadPixel0.07 and MSE values are reduced to 2.981 and 1.236, respectively, which demonstrates the effectiveness of our iterative scheme. Note that, performing more iterations (i.e., "iter_3" and "iter_4") cannot introduce significant improvements. Therefore, we set the iteration number to 2 in our OACC-Net for a good trade-off between accuracy and efficiency. Moreover, we explore the upper bound of our method by using the occlusion mask generated by groundtruth disparity. As shown in Table 1, by using "groundtruth" masks, occlusions can be well located and the accuracy is improved. It demonstrates that accurate occlusion masks are important for LF depth estimation.

Effect of decaying rate. We compare our method with different decaying rates in Eq. 8. As shown in Fig. 8, larger decaying rates can make our method perform better on scenes with heavy occlusions (see Fig. 8(a)) but reduce the robustness to large noise (see Fig. 8(b)). That is because, under large noise, the photometric consistency can be broken and some pixels can be mis-classified as occluded pixels with large decaying rates. Consequently, we set q=2 in our method for a good trade-off between occlusion-awareness and noise-robustness.

Table 2. Model size and running time of our OACC-Net at different stages. Here, a 9×9 LF with a spatial resolution of 512×512 is used as input. The proposed OACC achieves a very fast inference speed and significantly accelerates our OACC-Net.

Stages	*Shift-and-Concat*	*OACC-Net* (ours)
feature extraction	0.04 M / 0.004 s	0.04 M / 0.004 s
cost construction	0 M / **2.741** s	0.04 M / **0.004** s
aggregation & regression	4.93 M / 0.003 s	4.93 M / 0.003 s
mask generation	-	0 M / 0.015 s
cost construction	-	shared / 0.004 s
aggregation & regression	-	shared / 0.004 s
Total	4.97 M / 2.748 s	5.01 M / 0.034 s

Efficiency. We investigate the efficiency of our method by listing the model size and running time of our OACC-Net at each stage. Here, we introduce a variant by using the *shift-and-concat* approach for cost construction, where 80 views are shifted by 8 disparity levels. As shown in Table 2, the *shift-and-concat* approach spends 2.741 seconds on cost construction while our OACC spends only 4 milliseconds. Since our OACC can directly convolve pixels under specific disparities and avoids the repetitive shifting operation, the inference speed of our OACC-Net is significantly accelerated at the cost of a 0.04 M increase in model size.

4.3. Comparison to the State-of-the-art Methods

We compare our method to 13 state-of-the-art methods, including 7 traditional methods [4, 10, 24, 25, 31, 36, 38] and 6 deep learning-based methods [9, 17, 19, 20, 26, 29].

1) Quantitative Results: Table 3 shows the MSE and average running time achieved by different methods. It can be observed that our method achieves the lowest MSE (i.e., highest accuracy) on 9 scenes and the second lowest MSE on 2 scenes. We submitted our results to the 4D LF benchmark [8] for a comprehensive evaluation. Among all the 84 submissions, our method achieves the first and the second place in terms of average MSE and average BadPix0.07, respectively. Readers can refer to our supplemental material for additional results. Note that, our method only spends 0.034 seconds on each scene, which is faster than FastLFnet [9] by an order of magnitude. The high accuracy and efficiency demonstrate the superiority of our OACC.

Table 3. Mean square error (MSE) and average running time achieved by different methods on the 4D LF benchmark [8]. The best results are in red and the second best results are in blue.

Method	*Backgm*	*Dots*	*Pyramids*	*Strips*	*Boxes*	*Cotton*	*Dino*	*Sideboard*	*Bedroom*	*Bicycle*	*Herbs*	*Origami*	*Average*	*Time* (s)
LF_OCC [31]	22.78	3.185	0.077	7.942	9.593	1.074	0.944	2.073	0.530	7.673	22.96	2.223	6.755	519.9
CAE [36]	6.074	5.082	0.048	3.556	8.424	1.506	0.382	0.876	0.234	5.135	11.67	1.778	3.730	832.1
PS-RF [10]	6.892	8.338	0.043	1.382	9.043	1.161	0.751	1.945	0.288	7.926	15.25	2.393	4.617	1413
SPO [38]	4.587	5.238	0.043	6.955	9.107	1.313	0.310	1.024	0.209	5.570	11.23	2.032	3.968	2115
SPO-MO [25]	4.133	3.763	0.009	1.934	10.37	1.329	0.254	0.932	0.152	5.617	12.05	1.667	3.518	4304
OBER-cross-ANP [24]	4.700	1.757	0.008	1.435	4.750	0.555	0.336	0.941	0.185	4.314	10.44	1.493	2.584	183.0
OAVC [4]	3.835	16.58	0.040	1.316	6.988	0.598	0.267	1.047	0.212	4.886	10.36	1.478	3.968	19.41
EPN+OS+GC [20]	3.699	22.37	0.018	8.731	9.314	1.406	0.565	1.744	1.188	6.411	11.58	10.09	6.426	274.7
Epinet-fcn [26]	3.629	1.635	0.008	0.950	6.240	0.191	0.167	0.827	0.213	4.682	9.700	1.466	2.476	1.976
Epinet-fcn-m [26]	3.705	1.475	0.007	0.932	5.968	0.197	0.157	0.798	0.204	4.603	9.491	1.478	2.418	10.66
Epinet-fcn-9x9 [26]	3.909	1.980	0.007	0.915	6.036	0.223	0.151	0.806	0.231	4.929	9.423	1.646	2.521	2.041
EPI-Shift [17]	12.79	13.15	0.037	1.686	9.790	0.475	0.392	1.261	0.284	6.920	17.01	1.690	5.458	22.57
EPI_ORM [19]	3.411	14.48	0.016	1.744	4.189	0.287	0.336	0.778	0.298	3.489	4.468	1.826	2.944	76.61
LFAttNet [29]	3.648	1.425	0.004	0.892	3.996	0.209	0.093	0.530	0.366	3.350	6.605	1.733	1.904	5.862
FastLFnet [9]	3.986	3.407	0.018	0.984	4.395	0.322	0.189	0.747	0.202	4.715	8.285	2.228	2.456	0.624
OACC-Net (ours)	3.938	1.418	0.004	0.845	2.892	0.162	0.083	0.542	0.148	2.907	6.561	0.878	1.698	0.034

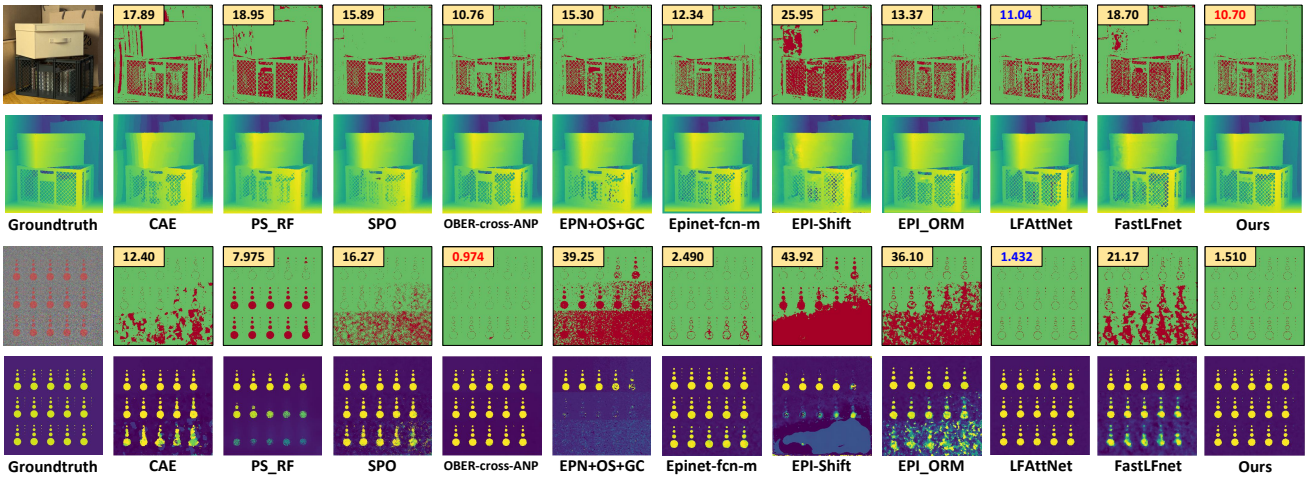

Figure 9. Visual comparisons among different LF depth estimation methods on the 4D LF benchmark [8]. For each scene, the bottom row shows the estimated disparity maps and the top row shows the corresponding BadPix0.07 maps (pixels with absolute error larger than 0.07 are marked in red).

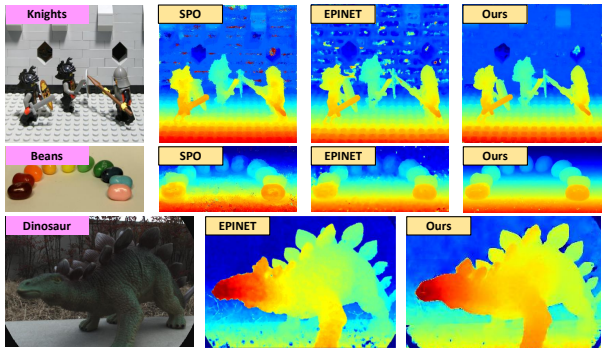

Figure 10. Visual results achieved by SPO [38], EPINET [26], and our method on real LFs.

2) Visual Comparison: Figure 9 shows the estimated disparities and corresponding BadPix0.07 maps. Since the proposed OACC can handle occlusions in a fine-grained manner, our OACC-Net performs well on scenes with heavy and complex occlusions (e.g., the nested structures in scene *boxes*). Besides, our method is also robust to noise and outperforms many state-of-the-art methods [9, 17, 19, 26] on scenes with large noise (e.g., the bottom dots in scene *dots*).

3) Performance on real LFs. We test the performance of our OACC-Net on real LFs captured by a moving camera [30] and a Lytro camera [1]. Since groundtruth depths are unavailable, we used the model trained on the synthetic LFs [8] for inference and compare the visual performance of our method to SPO [38] and EPINET [26]. As shown in Fig. 10, the depth maps produced by our method are more reasonable with fewer artifacts. It demonstrates that our OACC-Net can well generalize to real LFs. Please refer to our supplemental material for additional comparisons.

5. Conclusion

In this paper, we proposed an occlusion-aware cost constructor for LF depth estimation. Our OACC can efficiently construct matching cost and handle occlusions by modulating input pixels. Based on OACC, we developed a deep network called OACC-Net for depth estimation. Our method is highly efficient and achieves better performance than many state-of-the-art methods on different scenarios.

References

[1] Yunsu Bok, Hae-Gon Jeon, and In So Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):287–300, 2016. 8

[2] Jiaxin Chen, Shuo Zhang, and Youfang Lin. Attention-based multi-level fusion network for light field depth estimation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1, 2, 6

[3] Mantang Guo, Jing Jin, Hui Liu, and Junhui Hou. Learning dynamic interpolation for extremely sparse light fields with wide baselines. In *International Conference on Computer Vision (ICCV)*, pages 2450–2459, 2021. 1

[4] Kang Han, Wei Xiang, Eric Wang, and Tao Huang. A novel occlusion-aware vote cost for light field depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 7, 8

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[6] Stefan Heber and Thomas Pock. Convolutional networks for shape from light field. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3746–3754, 2016. 1, 2

[7] Stefan Heber, Wei Yu, and Thomas Pock. Neural epi-volume networks for shape from light field. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2252–2260, 2017. 1, 2

[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 (ACCV)*, pages 19–34, 2016. 2, 3, 6, 7, 8

[9] Zhicong Huang, Xuemei Hu, Zhou Xue, Weizhu Xu, and Tao Yue. Fast light-field disparity estimation with multi-disparity-scale cost aggregation. In *International Conference on Computer Vision (ICCV)*, pages 6320–6329, 2021. 1, 2, 6, 7, 8

[10] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Depth from a light field image with learning-based matching costs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):297–310, 2018. 7, 8

[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. 2

[12] Jing Jin and Junhui Hou. Occlusion-aware unsupervised learning of depth from 4-d light fields. *IEEE Transactions on Image Processing*, 2022. 4

[13] Jing Jin, Junhui Hou, Jie Chen, Huanqiang Zeng, Sam Kwong, and Jingyi Yu. Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[14] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics*, 32(4):73–1, 2013. 1

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning and Representation (ICLR)*, 2015. 6

[16] Mikael Le Pendu, Xiaoran Jiang, and Christine Guillemot. Light field inpainting propagation via low rank matrix completion. *IEEE Transactions on Image Processing*, 27(4):1981–1993, 2018. 6

[17] Titus Leistner, Hendrik Schilling, Radek Mackowiak, Stefan Gumhold, and Carsten Rother. Learning to think outside the box: Wide-baseline light field depth estimation with epi-shift. In *International Conference on 3D Vision (3DV)*, pages 249–257, 2019. 1, 7, 8

[18] Marc Levoy and Pat Hanrahan. Light field rendering. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996. 2

[19] Kunyuan Li, Jun Zhang, Rui Sun, Xudong Zhang, and Jun Gao. Epi-based oriented relation networks for light field depth estimation. In *British Machine Vision Conference (BMVC)*, 2020. 1, 7, 8

[20] Yaoxiang Luo, Wenhui Zhou, Junpeng Fang, Linkai Liang, Hua Zhang, and Guojun Dai. Epi-patch based convolutional neural network for depth estimation on 4d light field. In *International Conference on Neural Information Processing (ICNIP)*, pages 642–652, 2017. 7, 8

[21] Jiayong Peng, Zhiwei Xiong, Dong Liu, and Xuejin Chen. Unsupervised depth estimation from light field using a convolutional neural network. In *International Conference on 3D Vision (3DV)*, pages 295–303, 2018. 2, 4

[22] Jiayong Peng, Zhiwei Xiong, Yicheng Wang, Yueyi Zhang, and Dong Liu. Zero-shot depth estimation from light field using a convolutional neural network. *IEEE Transactions on Computational Imaging*, 2020. 2, 4

[23] Martin Rerabek and Touradj Ebrahimi. New light field image dataset. In *International Conference on Quality of Multimedia Experience (QoMEX)*, 2016. 6

[24] Hendrik Schilling, Maximilian Diebold, Carsten Rother, and Bernd Jähne. Trust your model: Light field depth estimation with inline occlusion handling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4530–4538, 2018. 2, 7, 8

[25] Hao Sheng, Pan Zhao, Shuo Zhang, Jun Zhang, and Da Yang. Occlusion-aware depth estimation for light field using multi-orientation epis. *Pattern Recognition*, 74:587–599, 2018. 2, 7, 8

[26] Changha Shin, Hae-Gon Jeon, Youngjin Yoon, In So Kweon, and Seon Joo Kim. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4748–4757, 2018. 1, 2, 6, 7, 8

[27] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *International Conference on Computer Vision (ICCV)*, pages 673–680, 2013. 2

[28] Michael W Tao, Jong-Chyi Su, Ting-Chun Wang, Jitendra Malik, and Ravi Ramamoorthi. Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1155–1169, 2015. 2

[29] Yu-Ju Tsai, Yu-Lun Liu, Ming Ouhyoung, and Yung-Yu Chuang. Attention-based view selection networks for light-field disparity estimation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 12095–12103, 2020. 1, 2, 6, 7, 8

[30] Vaibhav Vaish and Andrew Adams. The (new) stanford light field archive. *Computer Graphics Laboratory, Stanford University*, 6(7), 2008. 6, 8

[31] 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. 2, 7, 8

[32] Yingqian Wang, Longguang Wang, Jungang Yang, Wei An, Jingyi Yu, and Yulan Guo. Spatial-angular interaction for light field image super-resolution. In *European Conference on Computer Vision (ECCV)*, 2020. 4

[33] Yingqian Wang, Jungang Yang, Yulan Guo, Chao Xiao, and Wei An. Selective light field refocusing for camera arrays using bokeh rendering and superresolution. *IEEE Signal Processing Letters*, 26(1):204–208, 2018. 1

[34] 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. 2

[35] Sven Wanner, Stephan Meister, and Bastian Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *Vision, Modelling and Visualization (VMV)*, volume 13, pages 225–226, 2013. 6

[36] Williem, In Kyu Park, and Kyoung Mu Lee. Robust light field depth estimation using occlusion-noise aware data costs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2484–2497, 2018. 2, 7, 8

[37] Jingyi Yu. A light-field journey to virtual reality. *IEEE MultiMedia*, 24(2):104–112, 2017. 1

[38] 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. 2, 7, 8

[39] Wenhui Zhou, Enci Zhou, Gaomin Liu, Lili Lin, and Andrew Lumsdaine. Unsupervised monocular depth estimation from light field image. *IEEE Transactions on Image Processing*, 29:1606–1617, 2019. 4

[40] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9308–9316, 2019. 4