A Lightweight Depth Estimation Network for Wide-Baseline Light Fields

Yan Li, Qiong Wang, Lu Zhang, Gauthier Lafruit

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

A Lightweight Depth Estimation Network for Wide-baseline Light Fields

Yan Li, Qiong Wang, Lu Zhang, Gauthier Lafruit

***Abstract*—Existing traditional and ConvNet-based methods for light field depth estimation mainly work on the narrow-baseline scenario. This paper explores the feasibility and capability of ConvNets to estimate depth in another promising scenario: wide-baseline light fields. Due to the deficiency of training samples, a *large-scale* and *diverse* synthetic wide-baseline dataset with labelled data is introduced for depth prediction tasks. Considering the practical goal for real-world applications, we design an end-to-end trained lightweight convolutional network to infer depths from light fields, called *LLF-Net*. The proposed *LLF-Net* is built by incorporating a cost volume which allows variable angular light field inputs and an attention module that enables to recover details at occlusion areas. Evaluations are made on the synthetic and real-world wide-baseline light fields, and experimental results show that the proposed network achieves the best performance when compared to recent state-of-the-art methods. We also evaluate our *LLF-Net* on narrow-baseline datasets, and it consequently improves the performance of previous methods.**

***Index Terms*—Light field, depth estimation, convolutional neural network, lightweight, wide-baseline, narrow-baseline, synthetic dataset.**

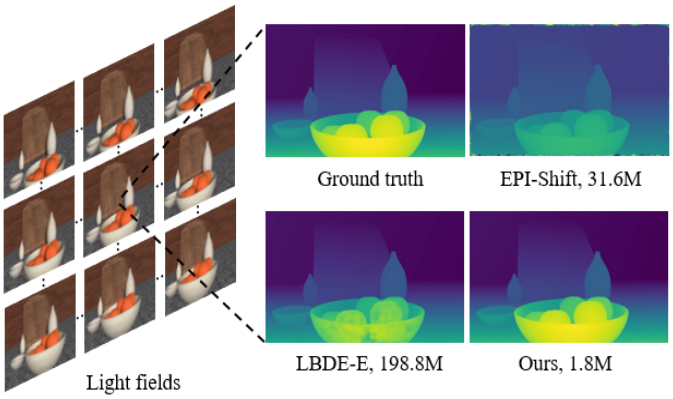

Fig. 1. An example of the proposed wide-baseline light fields. Our network *LLF-Net* has the fewer parameters (1.8 million) but is capable of performing more accurate depth estimation when compared to LBDE-E [25] with 198.8 million parameters and EPI-Shift [26] with 31.6 million parameters.

I. Introduction

In practical research areas such as 3D reconstruction, view synthesis and autonomous driving, accurate depth estimation is crucially needed. The light field, referred to as 4D computational photography technology, has been active in reconstructing depth in the real-world scenes. Different from 2D photography, light fields record the radiance of the lights from diverse directions, enhancing possibilities of perceiving depth. In this work, we focus on depth estimation for light fields.

Existing light field datasets can be divided into narrow-baseline and wide-baseline. Narrow-baseline light fields are typically captured by a plenoptic camera where a grid of micro-lenses are placed between the main lens and the image sensor, e.g., Lytro ILLUM camera [1]. A light field image from the camera is usually separated into the so-called sub-aperture images, and the baseline between sub-aperture images is very narrow. To date, traditional [2–13] and ConvNet-based [14–20] methods have been well studied for high performance in narrow-baseline light fields, and achieved a low percentage of errors, e.g., EPINET [19]. For wide-baseline light fields, they are usually captured by a camera array or gantry (i.e., a conventional camera is placed onto a gantry, and then uniformly moved by a motor in a plane). The baseline between the recorded wide-baseline light-field images is large and the spatial resolution of images is usually high. Compared with the narrow-baseline scenario, the wide-baseline is more capable of improving depth accuracy due to its large baseline [21]. Up to now, considerable efforts have been also made by traditional methods [3, 6, 10, 13, 22–24] to solve the problem of depth estimation in the wide-baseline scenario. However, ConvNet-based approaches are rarely studied in this scenario. Our objective is to explore and apply ConvNets into depth estimation for wide-baseline light fields.

Training ConvNets requires a large amount of labelled data. Unfortunately, there were no large-scale public wide-baseline light field datasets to the extent that limits the development of ConvNet-based methods. For a new dataset creation, a straightforward way is to collect real data and label them through physical depth sensing devices (e.g., structure light sensor or LiDAR). However, it is difficult, tedious or even unsuitable: structured light sensor is cheap but usually produces inaccurate depth which may cause performance degradation in ConvNets models, while LiDAR offers accurate depth but is unaffordable. Similar to the narrow-baseline scenario, we put effort into creating a synthetic wide-baseline dataset with accurate (ground truth) depths, aiming at training and evaluating ConvNet models, inferring depth for real-world datasets, and serving to the research community for future promising researches. We use a 3D computer graphics software to create a large-scale, wide-baseline synthetic dataset with diversities

Manuscript received 2020. The paper is supported by the China Scholarship Council funding and Van Buuren-Jaumotte-Demoulin doctoral funding.

Yan Li and Gauthier Lafruit are with the LISA Department, Université Libre De Bruxelles, Brussels, Belgium. Lu Zhang is with the IETR lab, INSA Rennes, Rennes, France. Qiong Wang is with Zhejiang University of Technology, Hangzhou, China (e-mail: wangqiong819@gmail.com).

(around 0.4K light fields), called *WLF*. It consists of two subsets: photo-realistic subset (*Hand-designed*), and nearly photo-realistic subset (*Flying-objects*). Fig. 1 shows an example scene that contains light fields and ground truth.

With respect to the ConvNet model for wide-baseline scenario, we consider to develop a *lightweight* model toward more practical goals, e.g., applications in mobile devices. To date, Shi *et al.* [25] present a divide-and-train deep model and Leistner *et al.* [26] present an end-to-end trained deep model for wide-baseline light field depth estimation. However, the two models have more than 198 million parameters and 31 million parameters respectively, which are so heavyweight that they are less suitable for practical applications. Since EPINET [19] achieves the state-of-the-art performance in the narrow-baseline scenario with much fewer parameters (5.1 million), we made an attempt to test and re-train (denoted as EPINET_T) the proposed *WLF* dataset, but the performances are too poor (cf. Fig. 10). Thus, we put forward a novel end-to-end trained lightweight and effective network *LLF-Net* by taking knowledge from stereo-based ConvNet models. As shown in Fig. 1, our model is lightweight and the quality of our reconstructed depth map is superior to the state-of-the-art methods in the wide-baseline scenario.

In our network, features are extracted for each view of the horizontal and vertical stream of light field views, and then the cost volume is obtained by a sequence of operations, i.e., shift-interpolation, cost calculation and fusion operations. With respect to the fusion, a divide-concatenate-sum operation is adopted, allowing flexible light field inputs while maintaining depth accuracy. An attention mechanism is proposed used in the cost aggregation module to adaptively assign weights to the feature maps of image views in each stream (view attention) and to the two streams (stream attention), which helps depth estimation at occlusion regions and preserves depth discontinuities. We make evaluations of the proposed network on the *WLF* test set and real-world datasets, and experimental results show that our network outperforms state-of-the-art methods in both quantitative and qualitative evaluations.

The main contributions of our work are summarized as follows:

1) We introduce a large-scale and diverse synthetic dataset with ground truth labels in the wide-baseline scenario for the first time (to the best of our knowledge), offering possibilities to further exploit and validate the learning-based methods.

2) We design a novel end-to-end trainable lightweight network for light field depth estimation.

3) Our network is built with the novel cost volume module that allows flexible light field inputs, and the attention module for better handling occlusions and preserving depth discontinuities.

4) The performance of the proposed network outperforms state-of-the-art light field depth estimation methods in the wide-baseline scenario, and even the narrow-baseline scenario.

We have released the training sets, comprising of part I at https://zenodo.org/record/3931237.X1_B_RT7SaF and part II at https://zenodo.org/record/3934712.X1_CoxT7SaF. The rest of the proposed *WLF* dataset, the trained models and the source code of the proposed model are released at https://sites.google.com/site/yanliresearch/llf-net after publication, which can be used for reproducing our results, comparisons and further improvements.

II. Related Work

A. Deep learning-based methods

Recently, deep learning methods have gained much attention in estimating depth from light fields. Heber et al. [14], Luo et al. [15], Heber et al. [16] and Feng et al. [17] feed the input of Epipolar Plane Image (EPI) to the ConvNet where the network learns the proportional relation between the slope of the epipolar line and depth. However, this relation is hard to learn in wide-baseline light fields due to the absence of the epipolar line on the EPI. Shin *et al.* [19] present the EPINET where the geometric characteristic of epipolar images is used for depth estimation. EPINET achieves top-performing performance for the narrow-baseline scenario, but poor performance for the wide-baseline scenario even with re-training on wide-baseline datasets. Shi *et al.* [25, 27] propose to fine-tune FlowNet 2.0 [28] for the initial depth prediction, followed by a refinement network to improve depth quality. Leistner *et al.* [26] present an EPI-Shift network, where light field stacks are shifted from wide-baseline to narrow-baseline, and then used to predict depths by trained models from narrow-baseline datasets. Though these two methods can work onto wide-baseline light fields, both models are heavyweight and not suitable for practical applications. In contrast, our proposed network is a lightweight network, and able to perform well on the wide-baseline light field inputs.

B. Light Field Datasets

We review the publicly available synthetic light field dataset, which are frequently used for training or comparing the performance of competing ConvNet-based methods in depth estimation.

Most of these available datasets belong to the narrow-baseline, which are composed of a grid of 9x9 light field image views and with the small disparity range [-4, 4] (HCI [29], CVIA-HCI [30], and DLFD [31]). HCI [29] and CVIA-HCI [30] are the two most frequently-used datasets for the narrow-baseline scenario in literature. The HCI includes 7 frames/scenes with available full ground truth depths, which are usually used for evaluation. The CVIA-HCI includes 16 frames with available ground truth depths that are provided for training, and 8 frames with available ground truth provided for evaluating the methods. Nevertheless, models trained on the CVIA-HCI and/or even HCI and DLFD datasets are not able to infer depth well on wide-baseline datasets due to the mismatch between the source and target disparity range. Hence we propose to build a large-scale light field dataset to train the deep models and stimulate more future developments for the wide-baseline scenario.

III. Methodology

The goal of the proposed method is to estimate the dense depth map for the center view from light fields. The light field

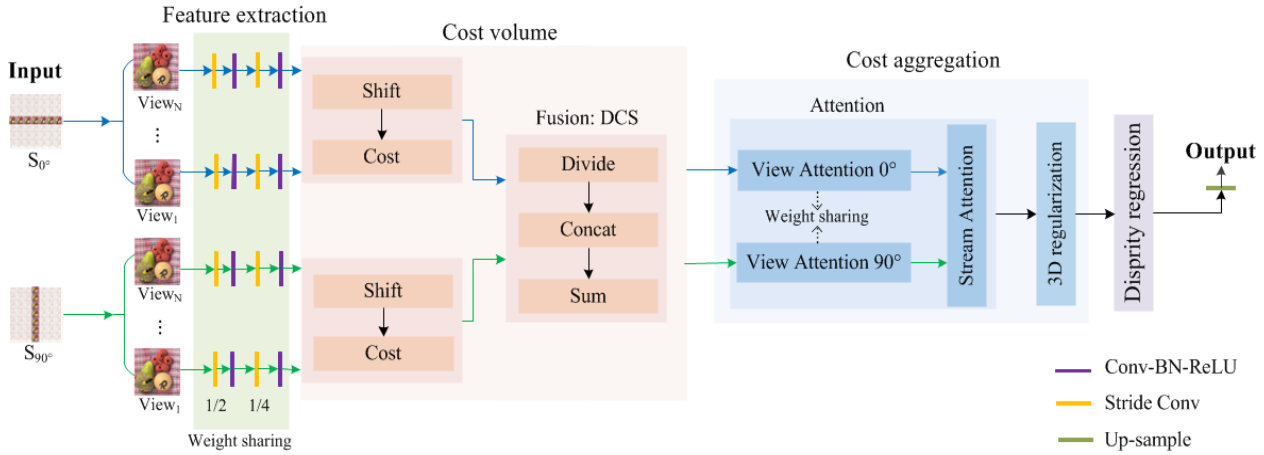

Fig. 2. Overview of the proposed network architecture.

inputs are parameterized by the two-plane parametrization $L(x, y, u, v)$, where (u, v) are the camera plane coordinates and (x, y) are the image plane coordinates. The relationship between the other views and center view are represented as:

$$L(x, y, u, v) = L(x + (u^* - u)d(x, y), y + (v^* - v)d(x, y), u^*, v^*), (u \in [1, N], v \in [1, N]) \quad (1)$$

where N denotes the number of (angular) views along the horizontal and vertical directions in light fields, (u^*, v^*) represent the coordinates of the central view, and $d(x, y)$ is the disparity of the pixel (x, y) in the central view (i.e. the offset between the central view and its adjacent right view). Estimating the disparities (or depths) of the central view corresponds to searching the offset of corresponding points in other views.

An overview of the proposed network architecture is illustrated in Fig. 2 and detailed in Table I. The cross-hair of light field images are used and fed into the proposed network. To make image correspondence features distinguishable, deep feature descriptors are extracted for each view from cross-hair views in the *Feature Extraction* (Section III-A). Next, the discriminative cost volume [3, 4] is constructed by operating all extracted features in the *Cost Volume Generation* (Section III-B). Afterwards, an attention mechanism is introduced to remove disparity errors caused by the occlusion, and a 3D encoder-decoder network is applied to regularize the disparity space in the *Cost Aggregation* (Section III-C). Finally, the disparity map is produced in *Disparity Regression* (Section III-D), and the smooth L1 loss (Section III-E) is used for training our network.

TABLE I
THE DETAILS OF THE PROPOSED NETWORK ARCHITECTURE.

Layers	Output size	Input layer	Output layer
Feature extraction (for each X $\in$ S_{0° $\cup$ S_{90°)			
Conv_K2S2	$H/2 \times W/2 \times C$	X	C1_1
ConvBnR_K2S1	$H/2 \times W/2 \times C$	C1_1	C1_2
Conv_K2S2	$H/4 \times W/4 \times C$	C1_2	C2_1
ConvBnR_K2S1	$H/4 \times W/4 \times C$	C2_1	C2_2
Cost volume (C2_2S_{0° = $\{C2\_2\}_1^N$, C2_2S_{90° = $\{C2\_2\}_1^N$)			
Shift_Cost	L/4×H/4×W/4× NC	C2_2S_{0°	SIC1
Shift_Cost	L/4×H/4×W/4× NC	C2_2S_{90°	SIC12
Div_Concat	L/4×H/4×W/4× $6C$	SIC1, SIC2	$\{DC\}_1^{3(N-1)/2}$
Sum	L/4×H/4×W/4× $6C$	$\{DC\}_1^{3(N-1)/2}$	[CV_{0°, CV_{90°]
View and Stream attention			
View Attention 0°	L/4 × H/4 × W/4 × 3C	CV_{0°	CV_v1
View Attention 90°	L/4 × H/4 × W/4 × 3C	CV_{90°	CV_v2
Stream Attention	L/4 × H/4 × W/4 × 6C	CV_v1, CV_v2	CV_s
Cost regularization			
3DConvBnR_K3S1	L/4 × H/4 × W/4 × 2C	CV_s	3Cbr1
3DConvBnR_K3S1	L/4 × H/4 × W/4 × 2C	3Cbr1	3Cbr2
3DConvBnR_K3S2	L/8 × H/8 × W/8 × 4C	3Cbr2	3Cbr3
3DeConvBnR_K3S2	L/4 × H/4 × W/4 × 2C	3Cbr3	3DCbr1
3DeConvBnR_K3S1	L/4 × H/4 × W/4 × 1	3DCbr1	3DCbr2
Upsampling	L × H × W × 1	3DCbr2	Up1
SoftArg	H × W × 1	Up1	D

A. Feature Extraction

Our network takes as input the horizontal and vertical streams of image views with the dimension $H \times W \times N$ from light fields, where H and W represent the height and width of image (spatial resolution). We apply a 2D plane convolution network to extract distinct features. It is firstly constructed by one convolution layer and one Conv-Bn-Relu block (a convolution layer followed by a batch normalization layer, and a ReLU layer), in which the stride of the former convolution layer is set to 2 for down-sampling inputs and the latter is set to 1. Then the same structure is repeated to produce sub-scale features. Finally, the output feature maps are downsized to a quarter spatial resolution. The kernel of the convolution filters is 2x2. We adopt the shared 2D network on both streams of views since we found sharing parameters is better than the non-sharing case in terms of disparity accuracy and efficiency.

B. Cost Volume Generation

Given two streams of feature maps, a sequence of operations, i.e., shift-interpolation, cost calculation and fusion are used to generate the cost volume. Note that building the cost volume does not introduce any parameters to train.

1) Shift-Interpolation: The feature maps of the central view F_r are regarded as the reference, and the others along the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

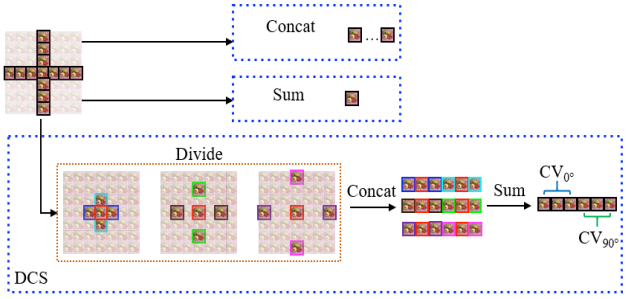

Fig. 3. Variants for cost fusion (best viewed in color).

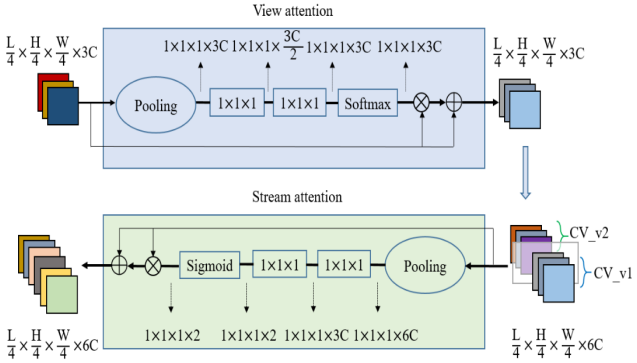

Fig. 4. Epipolar view and stream residual attention. Global max-pooling is used in pooling to downsize inputs, and each first 1x1x1 convolution is followed by a ReLU activation.

stream are the target feature maps F_t. The feature maps of target views F_t are shifted toward the reference view by each hypothesis disparity $\hat{d}$ within the disparity range (see Eq. 2). Then the bilinear interpolation is employed to calculate appropriate values for each pixel at sub-pixel position.

$$\hat{d} = d_{min} + n(d_{max} - d_{min})/L, (n \in \{0, 1, ..., L-1\}) \quad (2)$$

where d_{min}, d_{max} and L represent the minimum and maximum disparity in the range and the number of labels (or discrete disparities), respectively. The dimension of (warped) feature maps for each target view and the feature maps for the reference view is herein $(L \times H^l \times W^l \times C^l)$, where l denotes the scale level and C indicates the channels.

2) Cost Calculation: After obtaining the warped feature maps from streams of target views at the scale level l, we calculate the matching cost by using the concatenation [32] between the reference and target feature maps to form a 4D cost volume, as shown in Eq. (2) and Eq. (3).

$$C^l(\hat{d}, y, x, c) = \mathrm{C}\{F^l(y + \hat{d}(v^* - v), x + \hat{d}(u^* - u), c_i) , (i = 1, ..., N)\} \quad (3)$$

Herein, c_i denotes all the feature channels of view i.

3) Fusion: At this step, we make a fusion of calculated costs across views and streams such that neighboring views or streams enhance capabilities of solving ambiguity problems in correspondence matching. Actually, there exists different strategies to perform cost fusions. From the perspective of input sizes, strategies vary from fixed, to non-fixed or near-fixed inputs. Hereafter we discuss fusion variants in details, as are demonstrated in Fig. 3.

Concat is employed to concatenate all reference-target pairs of costs or horizontal and vertical groups of costs along the channel dimension, where the stacked size of the former is $L^l \times H^l \times W^l \times 4NC$, and the latter is $L^l \times H^l \times W^l \times 2NC$. Since the number of stacked feature channels is equal to that of convolution input filters, the networks then require the fixed inputs (i.e. the number of input views during the test should be the same with that during the training).

Sum computes the sum of all reference-target costs in which each cost is calculated by the absolute difference between the reference and target view. The sum fusion produces the fixed-length output $L^l \times H^l \times W^l \times C$ regardless of the input size.

Divide, Concat, Sum (DCS) is designed to fuse costs across multiple-baseline cost volumes. The DCS fusion is presented as the combination of the Concat fusion and the Sum fusion. Given the cross-hair light fields, all the target views are positioned along the four directions (i.e. left, right, top, and bottom direction) of the reference view, and there exists a large set of target views along each direction. Actually, the target views in each direction are located in different camera baselines, ranging from 1 to N-1; thus there might exist redundancy among these baselines. An intuitive idea for eliminating this redundancy is to make fusions across the baselines. Specifically, we propose to divide the cross-hair light field inputs into (N-1)/2 groups with the same baseline of views; and each group contains five views, including one reference view and its four directional neighboring target views. For each group, all feature maps are firstly concatenated across the channel dimension (Concat fusion). Then we take the sum over all groups (Sum fusion). Note that the vertical views are not rotated by 90 degrees. Compared with the pure Concat fusion, this fusion requires any number of groups of the above-mentioned five input views, which is not limited to the fixed number of input views. Compared with the Sum fusion, this fusion maintains absolute directional information, which might contribute to high depth accuracy. The fused output, comprised of the horizontal part $\mathrm{CV}_0\circ$ and the vertical part $\mathrm{CV}_{90}\circ$ (see Fig. 3), has the dimension $L^l \times H^l \times W^l \times 6C$. DCS is flexible in the number of input angular views N. Note that when N is set to 3, DCS will be the same with the Concat fusion.

C. Cost Aggregation

Cost aggregation is leveraged to refine the fused cost volume since we did not take into account the potential occlusion issues before. With respect to the occlusion, we know that the same 3D real-world point that is visible in the reference view might be occluded by foreground objects in the target view, which leads to difficulties in finding correspondences. This issue is alleviated in our input cross-hair light fields since points might be visible in some angular views. Moreover, we are aware that points are occluded in the horizontal stream of views but might be less or not occluded in the vertical stream of views, and vice versa (cf. Fig. 8). Likewise, for

any stream that consists of multiple views, the point is not seen in some target views, but might be visible in other views. Thus, if all views are equally used in searching the correspondence, the occlusion issue appearing in some views or stream can cause trouble to find the correct disparity due to the mismatches. This implicitly indicates that the cost features calculated from the *Cost Volume Generation* are not equally important. Note that 2D attention modules (e.g. used to change the weights of the features on each stage of the ResNet to enhance the consistency [33]) are increasingly used in deep neural networks to focus on what we want. To make our model focus on more informative features (e.g. that are more helpful for alleviating the occlusion issue and preserving depth discontinuity), we extend the 2D attention network to two types of 3D attention networks (view attention and stream attention). The view attention module is designed to exploit the interdependencies among feature channels with the same angular direction (0° or 90°), while the stream attention module is designed to exploit the interdependencies among feature channels with different angular directions (cross-direction). Thus, the input of "View Attention 0°" is the horizontal part of the output of "Sum" (CV_{0°) and the input of "View Attention 90°" is the vertical part of the output of "Sum" (CV_{90°), cf. Table I or Fig. 3; and the input of the "Stream attention" is the concatenation of the output CV_v1 from "View Attention 0°" and the output CV_v2 from "View Attention 90°" along the channel dimension. For the attention block, cf. Fig. 4, the global average pooling is applied to capture the channel-wise statistics of the cost volume, followed by a simple gating mechanism capturing the channel-wise dependencies of the cost volume. Within this gating mechanism, the softmax operator in the view attention block is responsible for the output of the attention map, containing multiple different weights for the features in each stream; and the sigmoid operator in the stream attention block assigns the two weights to the features from the two streams respectively. The interdependent features of the cost volume are thus assigned to a large weight, and vice versa.

Followed by the attention network, a 3D encoder-decoder network is used to regularize the output of the attention network across the disparity dimension. This naturally involves large context information, which enforces the smoothness at low texture regions. This network is built by three 3D convolutions and two transposed convolutions. Last, the bilinear interpolation is used to resize back to the same spatial resolution of inputs, and the output has the dimension $L \times H \times W \times 1$.

D. Disparity Regression

The differential soft argmin operator proposed by [32] is employed to obtain the final disparity map. The soft argmin operator regresses continuous disparities $\widetilde{D}$ by calculating the expectation of weighted disparities, as given in Eq. 4,

$$\widetilde{D} = \sum_{\hat{d}=d_{min}}^{d_{max}} \hat{d} * P(\hat{d}) \tag{4}$$

where $P(\hat{d})$ is the weight probability of the pixel at disparity $\hat{d}$.

E. Training Loss

We use the smooth L1 loss for the training process, which is computed between the predicted disparity $\hat{d}$ and the ground truth g in patch p as in Eq. 5 and Eq. 6,

$$L = \sum_{i \in p} Smooth_{L1}(\hat{d}_i - g_i) \tag{5}$$

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| \leq 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{6}$$

IV. WLF Dataset

We have designed a large-scale, wide-baseline synthetic light field camera array dataset *WLF*. The quantity of the light fields is 381, which is around 14 times larger than that of the popularly-used dataset CVIA-HCI. Each light field provides 9x9 angular (RGB) images and ground truth disparities as similar to the CVIA-HCI dataset. The light fields cover high resolution (1920x1080) and low resolution (512x512) images.

A. Dataset Construction

To enrich the dataset diversity, the *WLF* dataset is constructed in two scenarios: Hand-designed and Flying-objects. The scenes in Hand-designed and Flying-objects scenarios are rendered by the Cycle engine in the open source software Blender [1]. Fig. 5 shows the rendered samples from these two scenarios, and the statistics of the *WLF* dataset are given in Table II.

Hand-designed Scenario: We carefully collect free 3D models from different websites [2] with free licenses and elaborately assemble them to create physically plausible and meaningful scenes. Each scene contains more than two challenges in depth estimation: fine structure, repetitive pattern, occlusion, shading, and/or glossy appearance. Hand-designed scenario counts the aesthetic impression, but the manual design of 3D scenes is tedious and expensive, which causes difficulties to generate a large size dataset. This subset includes 36 scenes, and is split into 24 training scenes and 12 test scenes.

Flying-objects Scenario The richness of the dataset content is significant, therefore we attempt to render new scenes with flying objects in a faster way, which is inspired by recent advances of synthetic scenes with flying objects [34–36] in deep learning methods. Specifically, we carefully collect a large number of 3D models from the websites[2] and [37], and collect the texture images and environmental maps from Google Image. We then make a 3D cube in 3D space of Blender software, and the surfaces of cube are randomly textured. Next, a number of objects, which vary from 2 to 20, are randomly and automatically put in the cube, including 1-15 static objects and 1-5 random moving objects. The objects are randomly scaled, rotated and translated. Moreover, the light intensity is random, and the light field cameras are randomly and slightly translated. This subset includes 345 scenes, and is provided for training models.

[1] https://www.blender.org/ [2] https://chocofur.com, https://sketchfab.com, https://free3d.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

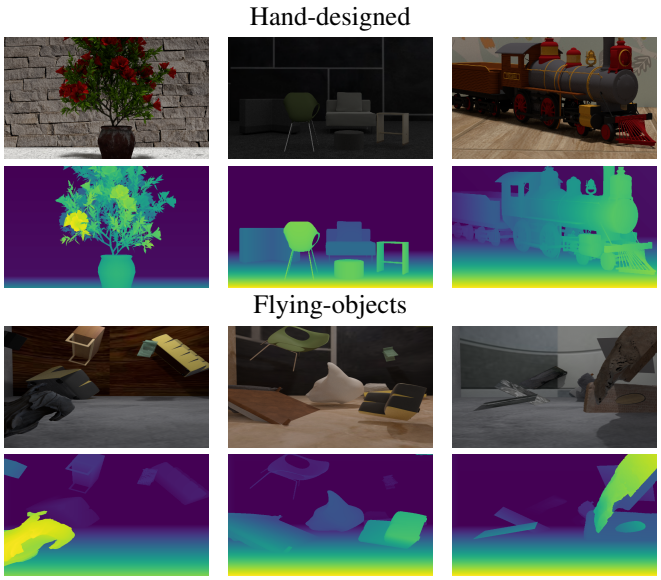

Fig. 5. Examples of *WLF* dataset: the central view and colored ground truth disparity map are shown.

TABLE II
DATASETS STATICS OF *WLF*

Dataset	#train	#test	spatial resolution	disparity range
Hand-designed	24	12	1920 ×1080	[0, 50]
Flying-objects	345		512 ×512	[0, 50]

B. Evaluation metrics

To measure the accuracy of reconstructed disparities from wide-baseline light fields, we adopt the widely-used metrics in depth estimation [30, 38, 39], i.e., mean square error (MSE) and bad pixels. The bad pixels are computed as the percentage of errors between the predicted disparity and ground truth disparity that are larger than a threshold. Considering thresholds in [30, 38, 39], the thresholds of the bad pixels are set to 0.15, 0.3, 0.6 and 1 in the paper. Note that a lower MSE or bad pixel percentage means a better performance.

V. EXPERIMENTS

A. Implementation details

We use Tensorflow [40] to implement the proposed network. The training and inference are both run on a Windows PC equipped with a Nvidia GTX 1080Ti GPU with 11GB memory and Intel i7 3.6Ghz CPU with 32GB memory. We use randomly cropped patches of size 128x128 for wide-baseline training set *WLF* and a smaller size 64x64 for narrow-baseline training set CVIA-HCI (due to its smaller quantity). Color scaling, 90, 180 and 270 degree rotation, etc are used for increasing the number of the data samples to the order of millions. We use the rmsprop optimizer [41], and start at the learning rate 1e-4, and then divide it by two after 80k iterations for *WLF* and after 150k iterations for CVIA-HCI. For each iteration the mini-batch size is 8 for *WLF* and 16 for CVIA-HCI respectively. The d_{min} and d_{max} in Eq. (2) are set to 0 and 50 for *WLF*, -4 and 4 for CVIA-HCI respectively. The number of labels L is set to 128.

TABLE III
COMPARISONS OF THE BAD-0.3, BAD-0.6 AND PARAMETERS FOR THREE FUSIONS IN *Cost volume generation*. THE BEST PERFORMANCE IS IN **BOLD**.

Fusion	Parameters	Adaptive	Hand-designed bad-0.3	Hand-designed bad-0.6
Concat	2.5M	✗	**7.00**	3.37
Sum	1.5M	✓	12.72	5.33
DCS	1.8M	✓	7.01	**3.04**

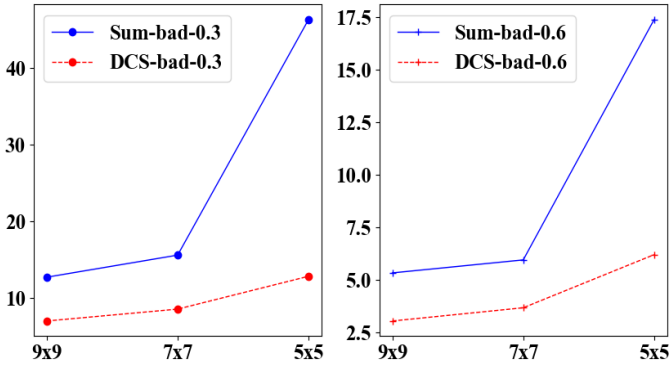

Fig. 6. Comparisons of the DCS fusion and Sum fusion on flexible angular inputs.

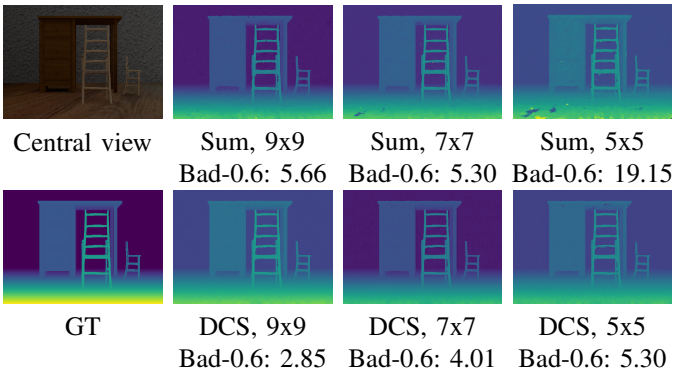

Fig. 7. Visual comparisons of the DCS fusion and Sum fusion on flexible angular inputs.

B. Ablation study

To validate the effectiveness of two proposed components in the *LLF-Net*, i.e. the fusion in *Cost volume generation* and the attention in *Cost aggregation*, the ablation studies are conducted on the Hand-designed validation set that consists of 8 scenes split from the training set. The bad-0.3 and bad-0.6 metrics are used for measuring the depth accuracy.

1) Fusion in Cost volume generation: Firstly, we make quantitative comparisons of different variants of fusions in *Cost volume generation* on aspects of depth accuracy and model size. Table III shows the evaluation results, and compares their adaptive ability of testing various angular resolutions without retraining the new angular resolution inputs. The proposed DCS fusion gets the best performance by bad-0.6 metric with considerable parameters. Besides, it is not limited by fixed angular resolution inputs.

Fig. 6 compares the performance results between two fusion

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

TABLE IV
COMPARISONS OF THE DEPTH ACCURACY AND PARAMETERS WITH AND WITHOUT ATTENTION NETWORKS IN *Cost aggregation*.

Dataset	Parameters	Hand-designed	
		bad-0.3	bad-0.6
w/o Attention	1.79 M	7.01	3.04
Attention	1.82 M	**6.25**	**2.72**

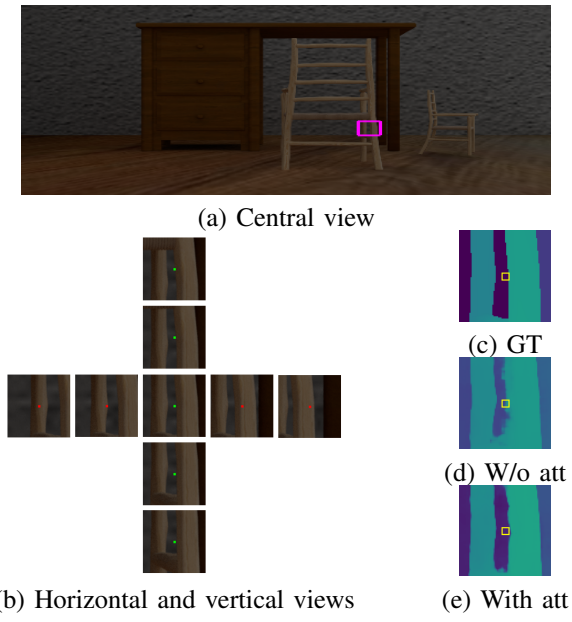

Fig. 8. Visual comparisons of depth estimation results without and with attention block. (a) central view with a selected patch P in pink bounding box, (b) the patch P (the intersection) and the corresponding patches in the horizontal and vertical views, where the red point indicates the pixel in the central view is occluded in the current view, and the green point means visible in the current view, (c) the ground truth disparity of patch P, (d) the estimated disparity map without attention block and (e) the estimated disparity map with attention block (best viewed in color).

ways (Sum and DCS) from testing variable angular resolutions. The DCS fusion always produces more accurate depths than the Sum fusion. When limiting the angular resolution, the DCS achieves much better performance, which means that it is more adaptive to limited input views. Fig. 7 illustrates visual comparison results from these two fusions. The DCS fusion undergoes a degradation in performance, but this is much less than that from the Sum fusion where artifacts occur in the disparity map.

2) Attention networks in Cost aggregation: To test the necessity of the proposed attention networks, Table IV compares the quantitative evaluation results with and without using them. We find that using the attention networks considerably improves the quality of estimated disparity maps.

Fig. 8 shows a visual comparison of disparity estimation without and with using attention networks. For a pixel in the selected patch P (see Fig. 8 (b)), it is occluded in all horizontal views, but it is visible in all vertical views. With the attention networks for selecting more meaningful views, the disparities of pixels around occlusion regions are better estimated and the sharp boundaries at depth discontinuities are better preserved, as shown in Fig. 8 (c-e).

TABLE V
TRAINING DATASET SCHEDULING.

Dataset	Hand-designed	
	bad-0.3	bad-0.6
Hand-designed	10.18	5.53
Hand-designed+Flying-objects	**6.25**	**2.72**

TABLE VI
COMPARISONS OF THE DEPTH ACCURACY AND PARAMETERS WITH AND WITHOUT WEIGHT SHARING IN *Feature extraction*.

Dataset	Parameters	Hand-designed	
		bad-0.3	bad-0.6
No sharing	2.02 M	11.25	5.12
Sharing	1.82 M	**6.25**	**2.72**

3) Dataset scheduling: To check the necessity of the Flying-Object subset in the proposed *WLF* dataset, we performed ablation experiments under two different training set scheduling schemes. As shown in Table V, the qualitative performance with Flying-objects (with large-scale training frames) in training is improved by a large margin.

4) Weight sharing in Feature extraction: We trained the *LLF-Net* with and without the use of the weight sharing, respectively. The two cases are compared in terms of the number of parameters and the bad pixels with two threshold values 0.3 and 0.6, cf. Table VI. We see that the weight sharing case has two advantages. Firstly, the weight sharing helps to reduce the number of parameters of the model (about 0.2 million less parameters compared with the non-sharing case). Secondly, the weight sharing helps to improve the depth estimation performance (the bad pixels are approximately reduced by half). The weight sharing scheme, employed at the low-level layers, learns a general feature representation model for all input views, whereas the non-sharing scheme learns distinct feature representation models for different input views. The latter may lead to a situation where the corresponding pixels are performed by different mathematical operations so that the extracted features of all input views may not be well-matched, which is not beneficial to the subsequent correspondence searching.

5) Training patch size: The size of patches used in the patch-wise training may influence the performance of ConvNets [42]. We thus investigated this influence on the performance of the proposed *LLF-Net*. Specifically, four different sizes, i.e. 96x96, 128x128, 160x160 and 192x192, were selected for the experiments. From Fig. 9, we empirically observe that the model with patch size 128x128 has the lowest bad-0.3 and bad-0.6 errors. The patch size smaller than this size will degrade the depth accuracy, whereas the patch size larger than this size will make little difference, but requires a larger GPU memory footprint.

C. Performance on wide-baseline datasets

To verify the effectiveness of our network *LLF-Net* on wide-baseline light field datasets, we conducted experiments on

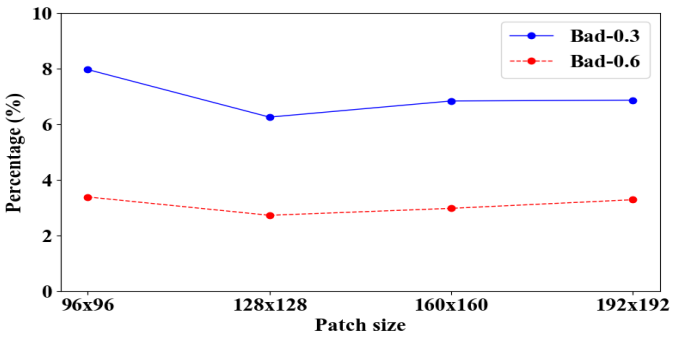

Fig. 9. Comparisons of the depth accuracy using different size patches during training.

the synthetic *WLF* dataset and real-world Google [43] and ULB [44] datasets. The proposed *LLF-Net* is compared with recent state-of-the-art depth estimation methods, comprising of traditional light field depth estimation methods LF_OCC [3] and RPRF [13], ConvNet-based methods HSM [21], EPINET [19] and LBDE-E [25]. With respect to ConvNet-based methods, HSM [21] is designed for stereo-based depth estimation, EPINET [19] achieves top performance among published methods in narrow-baseline light field datasets and LBDE-E [25] achieves considerable performance on both narrow- and wide-baseline light field datasets. Note that EPINET [19] is originally trained on narrow-baseline datasets and fails to infer depth in wide-baseline datasets, therefore we re-trained it using their public source code, and denote it as EPINET_T.

1) Synthetic dataset: Table VII shows the quantitative comparisons on the four exemplar scenes from Hand-designed test set of the *WLF* dataset. Comparing to state-of-the-art methods, our proposed *LLF-Net* achieves the lowest bad pixel errors in all four scenes. In Fig. 10, visual comparisons of these scenes are given, in which the first column for each scene displays the ground truth or estimated disparity map and the second column shows the central view and bad pixel error map. It is clear that our estimated disparity maps are all closer to the ground truth and have the fewest number of bad pixels. In contrast, the estimated disparity maps from LF_OCC are noisy, those from RPRF look over-smoothed and have quantification errors, both EPINET and EPINET_T fail to predict disparities, HSM [21] is disturbed by the ambiguous background, EPI-Shift [26] and LBDE-E [25] both seem not able to handle the foreground well.

Moreover, we use all test scenes (12 in total) of *WLF* for further comparisons by mse, bad-0.15, bad-0.3, bad-0.6 and bad-1 metrics. As shown in Table VIII, our end-to-end trained model produces the lowest average errors in all metrics even with the fewest parameters (110 times fewer than LBDE-E [25]). We finally compare our performance to the only two ConvNet based methods that allow adaptive angular light field inputs (9x9, 7x7 and 5x5 light fields) during inference. Fig. 11 shows that when the angular resolution of light fields is lower, the performance of our model that is trained from 9x9 light field inputs gradually degrades but is still much better than HSM [21] and LBDE-E [25].

2) Real-world dataset: Fig. 12 demonstrates visual comparisons on Google (5x5 light fields) and ULB (9x9 light fields) test scenes respectively. We exclude [26] for this comparison since the models that it provided only allowed 9x9 light field inputs. Though the proposed model is a lightweight model, it is capable of producing the more accurate disparity maps in real-world scenes when compared to the other ConvNet-based methods. Specifically, for both scenes, EPINET [19] is still not able to predict disparities, similar to their results from synthetic datasets. Ours have few noticeable artifacts in the foreground than that in LBDE-E [25], and in both foreground and background than that in HSM [13]. When compared to the traditional methods, ours have fewer artifacts than LF_OCC [3] and have fewer over-smoothness issues at occlusion regions in RPRF [13]. Besides we can clearly see from the background of the "Path" scene, our model is able to capture more details than the other methods, e.g., more details are recovered on persons.

D. Performance on narrow-baseline datasets

Though the focus of our work is on wide-baseline datasets, we also evaluate the performance of the proposed model on narrow-baseline datasets. The quantitative and qualitative comparison are further made between the proposed model and the recent state-of-the-art methods.

1) Quantitative evaluation: For evaluations, we test frequently-used datasets used in previous works, i.e. seven HCI and eight CVIA-HCI test scenes, and assess by metrics (mse, bad-0.1 and bad-0.07) defined in [29, 30, 45]. We compare our method with state-of-the-art traditional and ConvNet-based methods, and the results are summarized in Table IX. Our model improves the mse and makes a decrease by a large percentage in bad-0.1, when compared with the best published model EPINET [19]. Our model achieves similar accuracy with EPINET in bad-0.07 metric, however our model is end-to-end trained in less than two days on the same training set with EPINET that is trained more than five days. Furthermore, our model has the smallest capacity among ConvNet models, and runs the fastest (less than 0.5s per frame) among all top-performing methods in the narrow-baseline scenario.

2) Qualitative evaluation: Fig. 13 shows visual comparison results of our method with top-performing methods in Table IX on the scenes from synthetic datasets (HCI and CVIA-HCI) and real-world dataset (EPFL [46]). Clearly, the proposed method achieves the highest quality of disparity maps, especially on the challenging real-world scenes. Our model not only can recover disparities of the smooth surface with less noise, but is capable of capturing more details at occlusion areas, e.g., chain link fences or empty circles in real-world scenes.

E. Limitations and future work

The performance of the proposed *LLF-Net* is limited to some challenging issues, i.e. large textureless regions, heavily occluded (textureless or view-dependent) regions and non-Lambertian regions. In fact, it is very difficult to find the correspondences in the large textureless region since there are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

TABLE VII
BAD PIXEL ERROR PERCENTAGES OF THE FOUR EXEMPLAR SCENES OF THE *WLF* DATASET AGAINST THE GROUND TRUTH.

Scene	Buddha2		Furniture2		Perikles		Sideboards	
	bad-0.3	bad-0.6	bad-0.3	bad-0.6	bad-0.3	bad-0.6	bad-0.3	bad-0.6
LF_OCC [3]	98.41	88.64	97.96	49.75	98.49	92.93	97.01	73.67
RPRF [13]	11.10	0.86	17.11	1.16	14.82	1.26	31.15	25.46
HSM [21]	32.32	7.27	21.98	7.96	19.09	3.48	53.27	39.90
EPINET [19]	100	100	100	100	100	100	100	100
EPINET_T [19]	97.65	92.05	96.40	88.27	97.39	94.79	95.79	91.63
EPI-Shift [26]	22.22	4.51	22.09	6.57	48.81	9.36	50.67	38.16
LBDE-E [25]	14.01	7.72	16.92	8.18	46.20	32.78	45.89	39.23
Ours	**1.44**	**0.74**	**1.93**	**1.01**	**3.30**	**0.58**	**21.17**	**14.82**

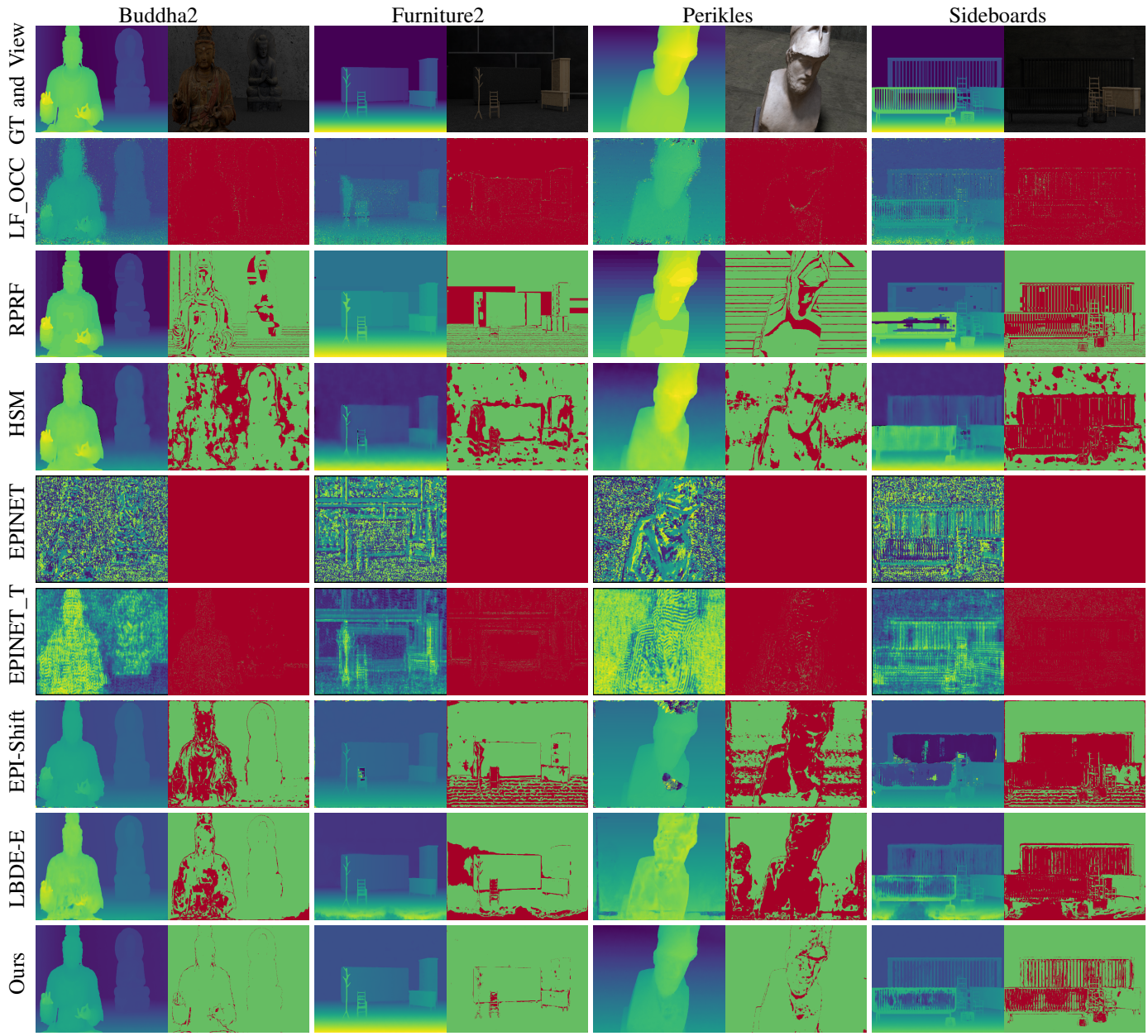

Fig. 10. Visual comparison results of the scenes from the *WLF* dataset: the central view and ground truth disparity map are shown in the first row, and the other rows show the predicted disparity maps and bad pixel error maps from state-of-the-arts respectively (Best viewed in color).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

TABLE VIII
PERFORMANCE COMPARISON RESULTS ON THE *WLF* TEST SET. THE AVERAGE ERRORS OF ALL FRAMES ARE LISTED AND THE BEST PERFORMANCE IS IN BOLD. THE QUANTITY OF PARAMETERS OF CONVNET-BASED METHODS IS IN MILLION (M).

Method	Parameters	End-to-end trained	Hand-designed				
			mse	bad-0.15	bad-0.3	bad-0.6	bad-1
LF_OCC [3]	-	-	13.56	98.86	97.54	78.63	40.86
RPRF [13]	-	-	1.70	40.43	16.01	5.43	4.70
HSM [21]	3.1	✓	1.58	62.22	36.08	14.22	8.52
EPINET [19]	5.1	✓	458.13	100	100	100	100
EPINET_T [19]	5.1	✓	86.89	98.56	97.10	94.11	89.92
EPI-Shift [26]	31.6	✓	20.76	61.55	35.95	14.65	12.59
LBDE-E [25]	198.8	✗	11.12	36.86	29.02	20.86	16.09
Ours	**1.8**	✓	**0.93**	**15.04**	**7.05**	**3.95**	**2.80**

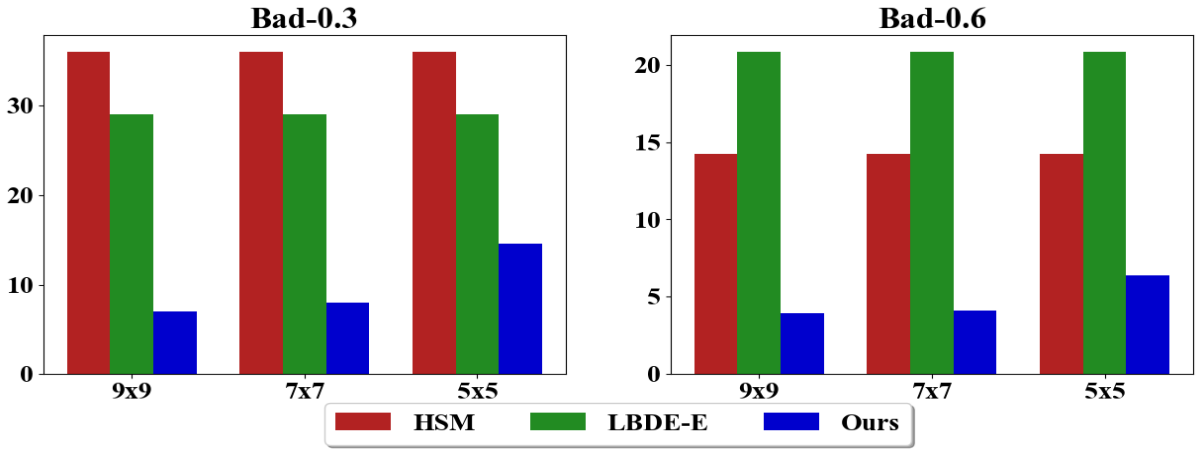

Fig. 11. Performance comparisons results from testing the various angular light field inputs.

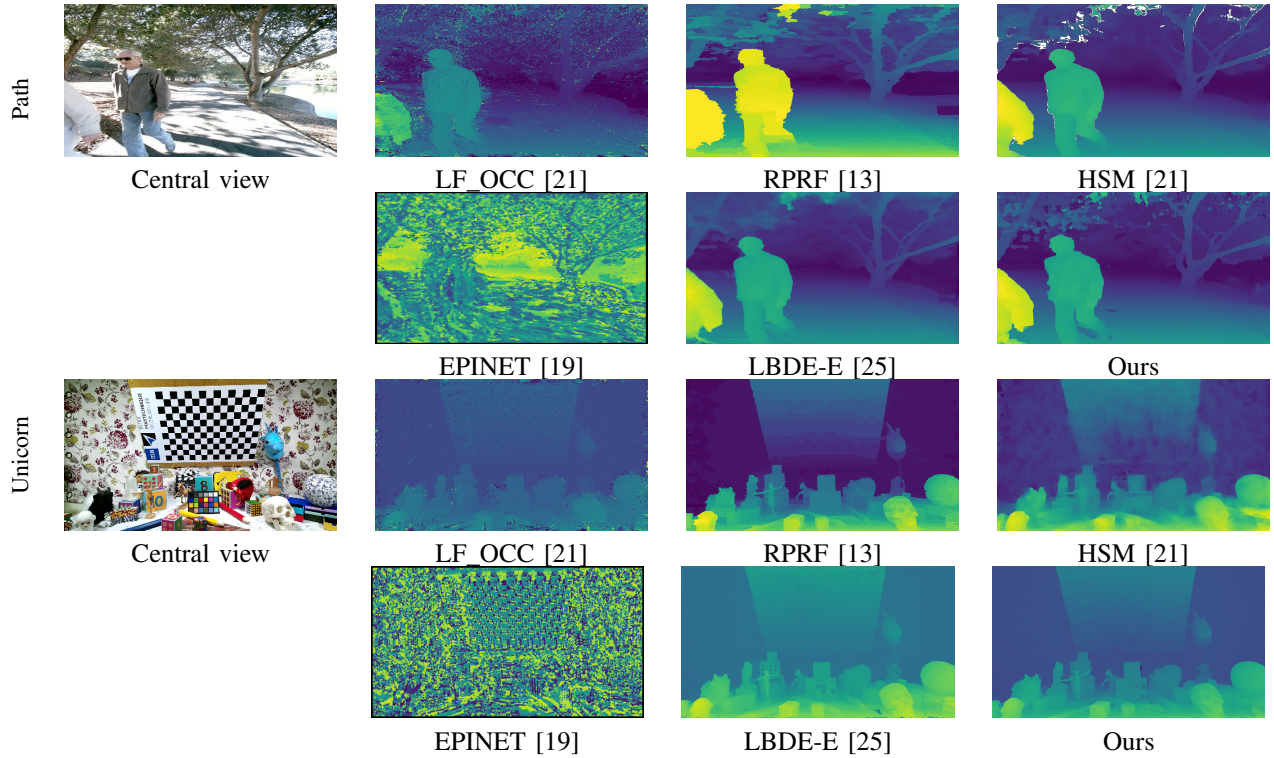

Fig. 12. Visual comparison results of wide-baseline real-world datasets: the central view and colored disparity map are shown (best viewed in color).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2021.3051761, IEEE Transactions on Image Processing

TABLE IX
PERFORMANCE COMPARISON RESULTS ON NARROW-BASELINE LIGHT FIELD DATASETS.

Method	Parameters (M)	End-to-end trained	mse	bad-0.1	bad-0.07	Training (days)	Inference time (s)
LF_OCC [3]	-	-	3.89	17.89	30.16	-	1.05e4
LF [4]	-	-	5.61	10.74	16.20	-	1.01e4
RPRF [13]	-	-	3.37	10.32	15.93	-	34.53
EPINET [19]	5.1	✓	2.68	9.06	**10.54**	5-6	1.98
EPI-Shift [26]	31.6	✓	11.41	10.89	14.84	4	22.57
LBDE-E [25]	198.8	✗	3.86	9.86	13.61	≈ 2	1.92
Ours	**1.8**	✓	**2.13**	**6.60**	10.66	≈ **1.6**	**0.46**

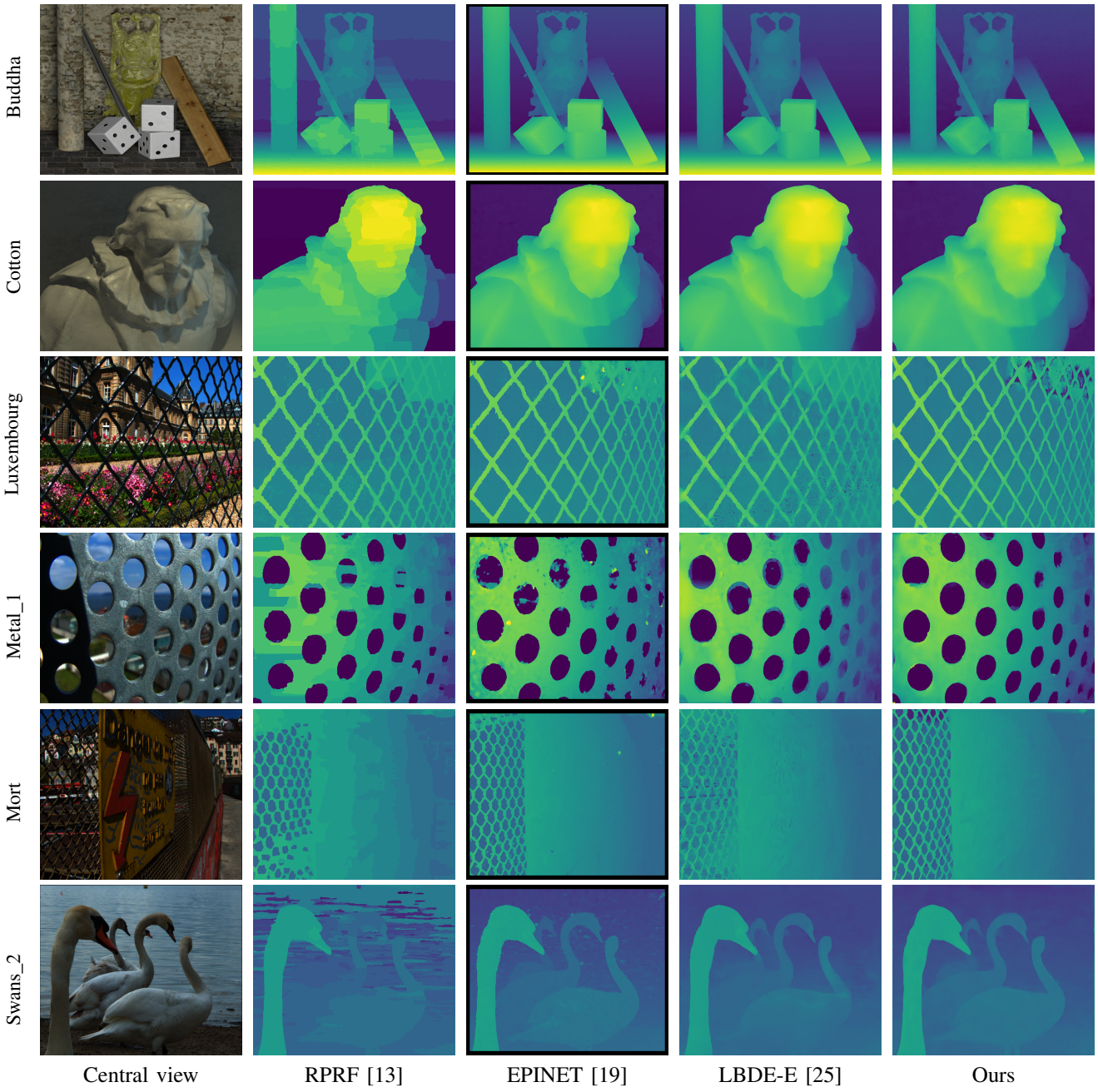

Fig. 13. Visual comparisons of synthetic and real-world datasets: the central view and colored disparity map are shown (best viewed in color).

a large number of similar pixels or features in the search range. One example (Luxembourg scene) is shown in Fig. 13, where the large textureless region, i.e. sky, is in the background. The correct disparity should be in dark color, which is almost not found in the disparity maps of our method and compared methods. This may be tackled by utilizing the segmentation technique [2] to partition the image into the textureless and texture regions beforehand. The heavily occluded (textureless or view-dependent) region is also a tough issue to handle, as shown in the Sideboards scene of Fig. 10 and the Path scene of Fig. 12. This issue is considered in our future work, in which the ground truth occlusion data samples are created and used in supervision. Finally, similar to most of the existing ConvNet-based methods, the proposed *LLF-Net* does not pay attention to the non-lambertian issue [47] either, such as the illumination changes, specular reflection and transparency, while this exists in the real-world scenes (e.g. Luxembourg scene in Fig. 13). Thus designing a new ConvNet that precisely estimates depth of both lambertian and non-lamertian surfaces is a promising future work.

VI. Conclusion

We introduce a large-scale wide-baseline synthetic light field dataset, which can be used for training or comparing competing methods in wide-baseline light field scenarios. We present a novel end-to-end trainable lightweight network *LLF-Net* based on the cost volume and attention modules for wide-baseline light field depth estimation, allowing flexible angular inputs. Experimental results show that the *LLF-Net* not only significantly improves the state-of-the-art performance in estimating depth in wide-baseline scenarios, but also narrow-baseline scenarios. Compared to the ConvNet-based methods on flexible angular inputs, our *LLF-Net* is adaptive and achieves better depth accuracy.

References

[1] R. Ng, "Lytro redefines photography with light field cameras." Accessed on May, 2017.

[2] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 73:1–73:12, 2013.

[3] T. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 3487–3495.

[4] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon, "Accurate depth map estimation from a lenslet light field camera," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1547–1555.

[5] L. Si and Q. Wang, "Dense depth-map estimation and geometry inference from light fields via global optimization," in *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III*, 2016, pp. 83–98.

[6] H. Zhu, Q. Wang, and J. Yu, "Occlusion-model guided anti-occlusion depth estimation in light field," *J. Sel. Topics Signal Processing*, vol. 11, no. 7, pp. 965–978, 2017.

[7] M. Strecke, A. Alperovich, and B. Goldluecke, "Accurate depth and normal maps from occlusion-aware focal stack symmetry," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 2529–2537.

[8] I. K. Park and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2484–2497, 2018.

[9] J. Navarro and A. Buades, "Robust and dense depth estimation for light field images," *IEEE Trans. Image Processing*, vol. 26, no. 4, pp. 1873–1886, 2017.

[10] Y. Li and G. Lafruit, "Scalable light field disparity estimation with occlusion detection," *Journal of WSCG*, vol. 26, no. 2, pp. 66–75, 2018.

[11] J. Chen, J. Hou, Y. Ni, and L. Chau, "Accurate light field depth estimation with superpixel regularization over partially occluded regions," *IEEE Trans. Image Processing*, vol. 27, no. 10, pp. 4889–4900, 2018.

[12] H. Schilling, M. Diebold, C. Rother, and B. Jähne, "Trust your model: Light field depth estimation with inline occlusion handling," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 4530–4538.

[13] C. Huang, "Empirical bayesian light-field stereo matching by robust pseudo random field modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 552–565, 2019. [Online]. Available: https://doi.org/10.1109/TPAMI.2018.2809502

[14] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 3746–3754.

[15] Y. Luo, W. Zhou, J. Fang, L. Liang, H. Zhang, and G. Dai, "Epi-patch based convolutional neural network for depth estimation on 4d light field," in *Neural Information Processing - 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part III*, 2017, pp. 642–652.

[16] S. Heber, W. Yu, and T. Pock, "Neural epi-volume networks for shape from light field," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2271–2279.

[17] M. Feng, Y. Wang, J. Liu, L. Zhang, H. F. M. Zaki, and A. S. Mian, "Benchmark data set and method for depth estimation from light field images," *IEEE Trans. Image Processing*, vol. 27, no. 7, pp. 3586–3598, 2018.

[18] W. Zhou, L. Liang, H. Zhang, A. Lumsdaine, and L. Lin, "Scale and orientation aware epi-patch learning for light field depth estimation," in *24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018*, 2018, pp. 2362–2367.

[19] C. Shin, H. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 4748–4757.

[20] W. Zhou, E. Zhou, Y. Yan, and L. Lin, "Learning depth cues from focal stack for light field depth estimation," in *2019 IEEE International Conference on Image Processing, ICIP 2019, 2019*, 2019, pp. 16–20.

[21] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical deep stereo matching on high-resolution images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5515–5524.

[22] Ł. Dabała, M. Ziegler, P. Didyk, F. Zilly, J. Keinert, K. Myszkowski, P. Rokita, and T. Ritschel, "Efficient multi-image correspondences for on-line light field video processing," in *Computer Graphics Forum*, vol. 35, no. 7. Wiley-Blackwell, 2016, pp. 401–410.

[23] N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, R. Gendrot, T. Langlois, O. Bureller, A. Schubert *et al.*, "Dataset and pipeline for multi-view light-field video," in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 30–40.

[24] A. Chuchvara, A. Barsi, and A. Gotchev, "Fast and accurate depth estimation from sparse light fields," *arXiv preprint arXiv:1812.06856*, 2018.

[25] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5867–5880, 2019.

[26] T. Leistner, H. Schilling, R. Mackowiak, S. Gumhold, and C. Rother, "Learning to think outside the box: Wide-baseline light field depth estimation with epi-shift," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 249–257.

[27] X. Jiang, M. L. Pendu, and C. Guillemot, "Depth estimation with occlusion handling from a sparse set of light field views," in *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*, 2018, pp. 634–638.

[28] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[29] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4d light fields," in *Proceedings of the Vision, Modeling, and Visualization Workshop 2013, Lugano, Switzerland, 2013*, 2013, pp. 225–226.

[30] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 19–34.

[31] X. Jiang, J. Shi, and C. Guillemot, "A learning based depth estimation framework for 4d densely and sparsely sampled light fields," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*, 2019, pp. 2257–2261.

[32] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66–75.

[33] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1857–1866.

[34] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[35] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[36] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 969–977.

[37] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[38] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German conference on pattern recognition*. Springer, 2014, pp. 31–42.

[39] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[41] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[42] J. Hamwood, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins, "Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of oct retinal layers," *Biomedical optics express*, vol. 9, no. 7, pp. 3049–3066, 2018.

[43] E. Penner and L. Zhang, "Soft 3d reconstruction for view synthesis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 235, 2017.

[44] D. Bonatto, A. Schenkel, T. Lenertz, Y. Li, and G. Lafruit, "Ulb high density 2d/3d camera array data set, version 2," *ISO/IEC JTC1/SC29/WG11 MPEG2017, m41083*.

[45] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 606–619, 2013.

[46] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, no. CONF, 2016.

[47] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light field intrinsics with a deep encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9145–9154.