Take Your Model Further: A General Post-refinement Network for Light Field Disparity Estimation via BadPix Correction

Rongshan Chen[1,2], Hao Sheng[1,2,3*], Da Yang[1,2], Sizhe Wang[1,2], Zhenglong Cui[1,2], Ruixuan Cong[1,2]

[1]School of Computer Science and Engineering, Beihang University
[2]Beihang Hangzhou Innovation Institute Yuhang
[3]Faculty of Applied Sciences, Macao Polytechnic University
{rongshan, shenghao, da.yang, sizhewang, zhenglong.cui, congrx}@buaa.edu.cn

Abstract

Most existing light field (LF) disparity estimation algorithms focus on handling occlusion, texture-less or other areas that harm LF structure to improve accuracy, while ignoring other potential modeling ideas. In this paper, we propose a novel idea called Bad Pixel (BadPix) correction for method modeling, then implement a general post-refinement network for LF disparity estimation: Bad-pixel Correction Network (BpCNet). Given an initial disparity map generated by a specific algorithm, we assume that all BadPixs on it are in a small range. Then BpCNet is modeled as a fine-grained search strategy, and a more accurate result can be obtained by evaluating the consistency of LF images in this limited range. Due to the assumption and the consistency between input and output, BpCNet can perform as a general post-refinement network, and can work on almost all existing algorithms iteratively. We demonstrate the feasibility of our theory through extensive experiments, and achieve remarkable performance on the HCI 4D Light Field Benchmark.

Introduction

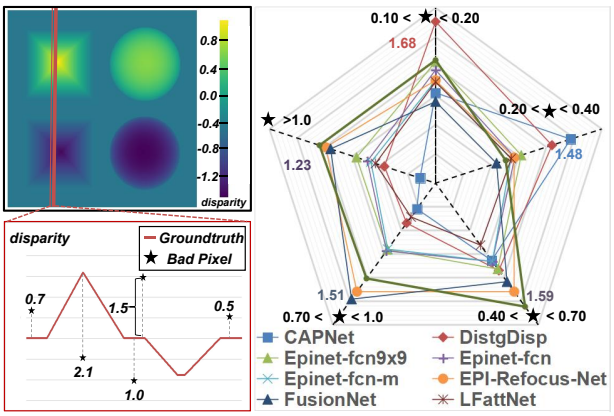
By collecting the lights from different directions of a scene, the light field (LF) is formed as regular and dense images sampled (Shi, Jiang, and Guillemot 2019). With these images, people can alleviate the problems caused by multi-view stereo matching effectively, and estimate scene depth information quickly and accurately (Johannsen, Sulc, and Goldluecke 2016; Chen et al. 2022). Moreover, with the advent of modern hand-held LF cameras, data acquisition becomes a simple task, then LF is naturally considered for depth estimation of real scenes. Since the scene depth information is directly related to the disparity between images, people usually estimate disparity map instead of directly calculating depth map in LF (Mishiba 2020; Sheng et al. 2022).

Although researchers recently have made significant progress in this field, especially with the rise of the convolutional neural network(Wang et al. 2022a; Yang and Tong 2022; Sheng et al. 2020), learning-based algorithms have improved the accuracy of disparity estimation to a new level (Wang et al. 2022c; Shin et al. 2018; Tsai et al. 2020; Chen, Zhang, and Lin 2021). However, when reviewing these algorithms, it can be found that most works, whether conventional or learning-based, focus on handling occlusion, texture-less or other areas that harm LF structure to improve accuracy, such as occlusion-aware cost construction (Park, Lee et al. 2017; Han et al. 2021) and edge detection (Chen et al. 2018; Huang et al. 2021), since these issues have been a pain point in this field (Neri, Carli, and Battisti 2018). And with the targeted research in previous years (Chen, Zhang, and Lin 2021; Wang, Lin, and Zhang 2021; Chen et al. 2014), such issues have been handled very well, but there is very little dedicated research to solve the problem of bad pixels for LF disparity estimation.

As shown in Fig. 1, nine high ranking algorithms are selected from the Benchmark (Johannsen et al. 2017) and analyzed. It can be found that 1) these algorithms still have some room for improvement in terms of BadPix, where still nearly 5-7% BadPixs in $(0.10, 1.0)$ interval even for these high ranking algorithms. 2) there are nearly 99% BadPix(<1.0), so it can be assumed that all BadPixs are in a small range δ for simplicity. 3) As these BadPixs can cause mse error, correcting them also has a positive effect on MSE$\times 100$. Following these observations, we propose a novel idea called BadPix correction for LF disparity estimation, which is the first time to the best of our knowledge, and implement a general post-refinement network–Bad-pixel Correction Network (BpCNet), which could work on almost all existing algorithms for disparity refinement.

To achieve our goal, there exist some challenges needed to be overcome. Firstly, since the distribution of BadPix is irregular, a reasonable strategy needs to be considered. Here, we model our BpCNet as: a fine-grained search strategy to find the best result in a limited range δ, composed of hypothesis disparities generation, feature extraction, cost volume construction and disparity fusion modules. Secondly, in BpCNet, some warp functions are needed to construct the cost volume of hypothesis disparities for consistency evaluation. However, as δ is small (≤ 1) and the number of disparities generated is large (≥ 9), ensuring that nearly close disparities are discriminative is another challenge. Conventional bilinear or bicubic-based warp function may cause the misjudgment between adjacent disparities if used, as their interpolated features are very similar. Here, we propose a novel phase shift-based warp function and get more accu-

*Corresponding author: Hao Sheng (shenghao@buaa.edu.cn)



(a) Visual description of Bad Pixel (b) Bad Pixel Analysis on Different Algorithms

Figure 1: An illustration of Bad Pixel and our analysis. (a) A sample visualization of BadPix, whose value equals to $|gt - algo|$, gt is the ground truth, $algo$ is provided by a specific algorithm. (b) BadPix analysis of existing algorithms on HCI 4D Light Field Dataset. Nine high ranking algorithms are selected and we analyze their performance on BadPix(α): the percentage of pixels with $|gt - algo| > \alpha$. α is the threshold predefined. It can be seen that these algorithms still have some room for improvement on BadPix.

rate warped features by applying phase shift theory within a local window. Thirdly, as a post-refinement network, BpCNet needs an initial disparity map as input. Take HCI 4D Light Field Dataset as an example, only 16 or fewer disparity maps can be provided, making network training a difficulty. To solve it, a unique training strategy based on data augmentation is proposed for BpCNet offline training, and experiments have shown that it does work.

Let us summarize our main contributions:

a. We propose a novel BadPix correction-based idea for LF disparity estimation, and implement a general post-refinement network–BpCNet.

b. A phase-shift-based warp function and a data augmentation-based training strategy are raised separately to solve the problems of adjacent disparity blur and little training data.

c. We demonstrate the feasibility of our theory through extensive experiments, and achieve remarkable performance on the HCI 4D Light Field Benchmark.

Related Work

In this section, previous algorithms are reviewed, including conventional and learning-based algorithms. Then some post-refinement algorithms are introduced.

Conventional Algorithms

Wanner et al. (Wanner and Goldluecke 2012) used structure tensor to compute the line slopes of EPI and got high-quality disparity maps. Zhang et al. (Sheng et al. 2018) applied a spinning parallelogram operator (SPO) on EPIs for disparity estimation. There also exist some algorithms not EPI-based. Tao et al. (Tao et al. 2013) computed dense depth estimation by combining both defocus and correspondence depth cues using LF cameras.

However, most conventional algorithms focus on handling occlusion, texture-less or other areas that harm LF structure to improve accuracy, as such problems are important factors leading to incorrect estimation. Wang et al. (Wang, Efros, and Ramamoorthi 2015) proposed to treat occlusion explicitly and improves their results by dividing regions into occluded and non-occluded. Williem et al. (Williem and Park 2016) proposed an angle entropy measurement and adaptive defocus response for data costs construction, which is robust to occlusion. Chen et al. (Chen et al. 2014) applied a bilateral consistency metric (BCM) on surface camera to tackle occlusions. Han et al. (Han et al. 2021) proposed a occlusion-aware vote cost (OAVC) to handle occlusion, with such assumption that unoccluded pixels are highly consistent with the central-view pixel.

Learning-Based Algorithms

Recently some learning-based algorithms have been proposed (Piao et al. 2021a,b). Luo et al. (Luo et al. 2017) proposed to train the network with EPI patch for disparity generation. Feng et al. (Feng et al. 2018) utilized synthetic LFs and designed a two-stream CNN network. Shin et al. (Shin et al. 2018) proposed a data augmentation method to address the lack of training data for LF. Leistner et al. (Leistner et al. 2019) introduced an idea to virtually shift the LF stack. Wang et al. (Wang et al. 2022c) proposed a generic mechanism to disentangle the coupled spatial and angular information for LFs processing.

Similar to conventional algorithms, more attention is paid to handle occlusion or texture-less areas with networks, such as AttMLNet, LFattNet, OACC-Net. Chen et al. (Chen, Zhang, and Lin 2021) designed an intra-branch fusion strategy and inter-branch fusion strategy to select features of views with fewer occlusions and richer textures. Tsai et al. (Tsai et al. 2020) proposed a view selection module to make LFattNet pay more attention to those views with less occlusion and more textures. OACC-Net(Wang et al. 2022b) constructed an occlusion-aware data cost for LF disparity estimation by dynamically modulating pixels from different views, making it robust to occlusions.

Post-refinement Algorithms

Some post-refinement algorithms are usually selected for refinement, such as multi-label-optimization-based and filter-based algorithms.

Multi-label-optimization-based algorithms formulate the problem as an energy model and solve it by global approaches, such as graph cuts (Boykov and Funka-Lea 2006) or belief propagation (Yedidia, Freeman, and Weiss 2000). Jeon et al. (Jeon et al. 2015) used graph cuts to correct disparity via neighboring estimation. Nevertheless, such algorithms are computationally expensive, especially with a large number of views and disparity labels. For filter-based algorithms, Zhang et al. took guided filter (He, Sun, and Tang 2012) as an edge-preserving operator to smooth the results. OAVC used weighted median filter (WMF) (Brownrigg 1984) and joint bilateral filter (Le, Jung, and Won 2014)

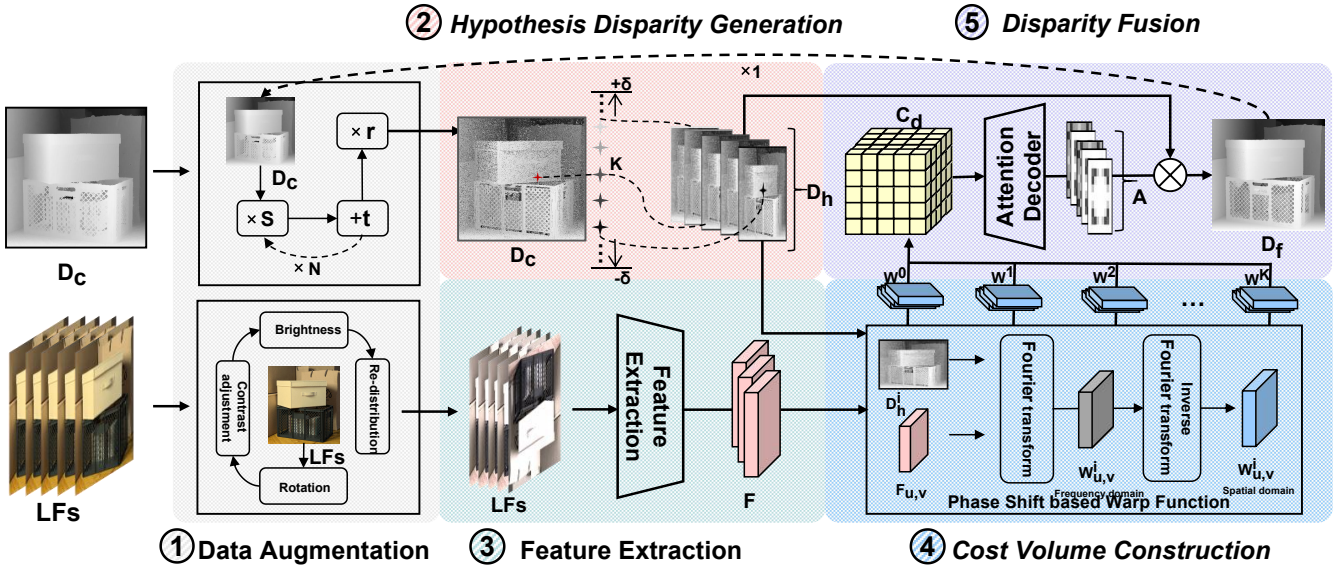


Figure 2: The structure of BpCNet, including ① Data Augmentation ② Hypothesis Disparity Generation ③ Feature Extraction ④ Cost Volume Construction and ⑤ Disparity Fusion. The input is D_c and LFs, and the output is D_f. Here, ① is used for more diverse LFs and D_c; ② aims to generate hypothesis disparities D_h^i for searching; ③ extracts the feature F from LFs; In ④, we warp the $F_{u,v}$ to $W_{u,v}^i$, then concatenate $W_{u,v}$ together to construct the cost volume C_d. Finally, we fuse all D_h^i into a better result D_f with the attention map A in ⑤. Another, we can also take D_f as a new D_c , and input it into BpCNet again for iterative training.

for a smoother result. But these algorithms only use the central view as a guide, which bring good visualization but are not reliable, while BpCNet utilizes all views and is more reliable. There also exist some learning-based networks for disparity refinement(Feng et al. 2018; Leistner et al. 2019), but only works with their methods and are not general.

Methodology

In this section, we propose a novel idea of BadPix correction, and implement a general post-refinement network which could work on almost all existing algorithms. Here, we assume that all BadPixs are in a small range δ for simplicity and model our method as a fine-grained search strategy in this limited range. Given an initial disparity map with corresponding light field images LFs, BpCNet can output a more accurate result. For simplicity of explanation, we agree that the input disparity maps as D_c, the output one as D_f, and the ground truth as D_g.

System Overview

As shown in Fig. 2, the input is $D_c \in R^{H\times W}$ and $LFs \in R^{U\times V\times H\times W}$, and the output is $D_f \in R^{H\times W}$. Where $U \times V$ is angular resolution, and $H \times W$ is spatial resolution for LFs. Its architecture is composed of five modules, from left to right are ① Data Augmentation: an offline training strategy for BpCNet with little data, including i) LFs augmentation for light field images, ii) D_c augmentation and iii) Iterative training for more diverse D_c. ② Hypothesis Disparity Generation: generate K hypothesis disparities $D_h^i (i = 1, 2, ..., K)$ in δ space around D_c for searching based on our assumption. ③ Feature Extraction: extract feature representation $F_{u,v} \in R^{H\times W}$ for each image $LFs_{u,v}$. ④ Cost Volume Construction: to evaluate the consistency of LFs under D_h^i, the feature representation $C_d \in R^{H\times W\times K}$ of D_h is constructed via a transformation formula and a phase shift based warp function. ⑤ Disparity Fusion: an attention based network are used to decode C_d into the attention map $A \in R^{H\times W\times K}$, then we can fuse these D_h^i into a better result D_f by weighted fusion. In addition, BpCNet is a lightweight network with only about 1.14M parameters.

For BpCNet, the execution flow of it is: (D_c, LFs) $\rightarrow$(①)$\rightarrow$(②,③)$\rightarrow$(④)$\rightarrow$(⑤)$\rightarrow$ $(D_f) \rightarrow \{(①\cdots)\}$. $\{\cdot\}$ is optional for iii) iterative training. The network details are given in the supplementary material's 'BpCNet' section.

Data Augmentation

As a post-refinement network, BpCNet needs an initial disparity map D_c as input. However, there is too little D_c to train. Take HCI 4D Light Field Dataset as an example, only 16 initial disparity maps can be provided for each algorithm. In this module, three augmentations are adopted, including LFs augmentation. D_c augmentation and Iterative training.

LFs **augmentation**. To get more diverse LFs, we borrow some image augmentation methods from previous work, including random color channel re-distribution, random brightness, contrast adjustments, random rotations by multiples of 90°, as ① in Fig. 2 shows.

D_c **augmentation**. As ① in Fig. 2 shows, we augment D_c by adding appropriate noise, including the scale s, the translation t and the random r. For scale noise, $D_c^{'} =$

$2\times s\times(D_c-D_g)+D_g$, and $s\in[-1,1]$ is a random number. By scaling the error between D_c and D_g, we can perform augmentation while maintaining the original distribution of D_c. For translation noise, $D_c^{'}=0.2\times t+D_c$, and $t\in[-1,1]$ is a random number. By translating D_c as a whole, the decoder in ⑤ enables to better learn the attention for D_h. For random noise, $D_c^{'}=r\times(D_c-D_g)+D_g$, $r\in R^{H\times W}$ is pixel wise and sampled from the truncated normal distribution (Burkardt 2014) with $[0.95,1.05]$. We introduce a weak perturbation r for a more diverse D_c. When training, we apply s and t to D_c for $N(\leq 3)$ times , then r is once,

Iterative training. As input D_c equals output D_f, we can also take D_f as a new D_c , and input it into BpCNet again for iterative training, one iteration is enough in experiments.

Hypothesis Disparity Generation

Following our assumption: the BadPixs in D_c are all within a limited range δ, so we just need to search within it to get a correct disparity value. Here , we generate K hypothesis disparities D_h^i at equal intervals for consistency evaluation between $(D_c-\delta)$ and $(D_c+\delta)$, as Form. 1 shows.

$$D_h^i=D_c+\frac{2\times\delta\times(i-\lceil K/2\rceil)}{K}.(i=1,2,...,K). \quad (1)$$

Feature Extraction

To be more robust, a feature extraction network is used to extract features $F_{u,v}$ for each image $LFs_{u,v}$, then we can get $F\in R^{U\times V\times H\times W}$. For the network architecture, we refer to Tsai's work and use the same feature extraction network as theirs.

Cost Volume Construction

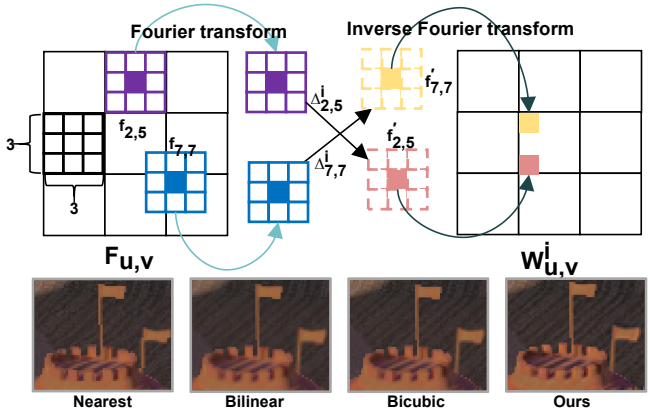


Figure 3: The process and results for our phase shift-based warp function. We apply the phase shift theory within a local window, then project $F_{u,v}$ to the warped feature $W_{u,v}^i$ for D_h^i efficiently. We take the pixel $(2,5),(7,7)$ as an example for explanation and compare the warped results between different methods.

After hypothesis disparity generation and feature extraction, the cost volume C_d is constructed to evaluate the consistency of LFs under each D_h^i.

According to LF structure, we can calculate the projection coordinate ϕ^i on F for each pixel on D_h^i by the transformation Form. 2.

$$\phi_{u,v}^i(x,y)=(x+d_u\times D_h^i(x,y),\quad y+d_v\times D_h^i(x,y)),$$
$$d_u=u_c-u,d_v=v_c-v, \quad (2)$$

Here, (x,y) is the pixel coordinate in D_h^i, (u_c,v_c) is the coordinate of center view in LFs, and $\phi_{u,v}^i(x,y)$ is the projection coordinate for $F_{u,v}$ to $D_h^i(x,y)$, which is not an integer usually. Then we can use a warp function for C_d construction, such as bilinear or bicubic interpolation based. However, since δ is small (≤ 1) and the number K of D_h is large (≥ 9), it means that the disparity distance between D_h^i and D_h^{i+1} is very close. Hence, conventional bilinear or bicubic interpolation-based warp function may cause the misjudgment for adjacent disparities if used, as their features interpolated are very similar.

Compared to bilinear and bicubic interpolation, phase shift-based interpolation can get more accurate subpixels (Jeon et al. 2015). As Form. 3 shows, if the feature $F_{u,v}$ is shifted by $\Delta_{x,y}\in R^2$, we can get the shifted feature $F_{u,v}^{'}$ via the 2D Fourier transform $S(\cdot)$ and Inverse Fourier transform $S^{-1}(\cdot)$.

$$F_{u,v}^{'}=S^{-1}\{S\{F_{u,v}\}exp^{2\pi i\Delta_{x,y}}\}, \quad (3)$$

However, this process is extremely inefficient if introduced into a warp function: as the offset $\Delta_{x.y}$ is calculated as $\phi_{u,v}^i(x,y)-(x,y)$, it is different as (x,y) changes, and $F_{u,v}$ needs to be shifted multiple times for the warped feature $W_{u,v}^i$ of D_h^i. To solve this problem, we propose to apply the phase shift theory within a local window. As shown in Fig. 3, we apply the 2D Fourier transform and Inverse Fourier transform to $F_{u,v}$ within a 3×3 window. To warp the pixel $(2,5)$ of $F_{u,v}$ to $W_{u,v}^i$ with D_h^i, the window $f_{2,5}$ is built and transformed into frequency domain, then shifted by $\Delta_{2,5}^i$. Finally, we transform $f_{2,5}^{'}$ into spatial domain and extract its center pixel. By performing this process for each (x,y), $W_{u,v}^i$ is constructed. In theory, this warp function can also apply to other cases not only in this work.

As ⑤ in Fig. 2 shows, we can get W^i for each D_h^i after warped, then the cost volume $C_d\in R^{H\times W\times K}$ is obtained by concatenating.

Disparity Fusion

With C_d, an attention decoder is built to evaluate the consistency of LFs under D_h, and output an attention map A_i for each D_h^i. Finally the D_f is obtained by weighted fusion, as Form. 4.

$$D_f=\sum_{i=1}^{K}(A_i\times D_h^i),(A_i\in A,i=1,2,...,K), \quad (4)$$

where $A\in R^{H\times W\times K}$ is output by a Softmax layer, $A_i\in R^{H\times W}$ is the pixel wise attention weight.

The metrics before refinement are shown with black values. Since some algorithms are refined multiple times, we

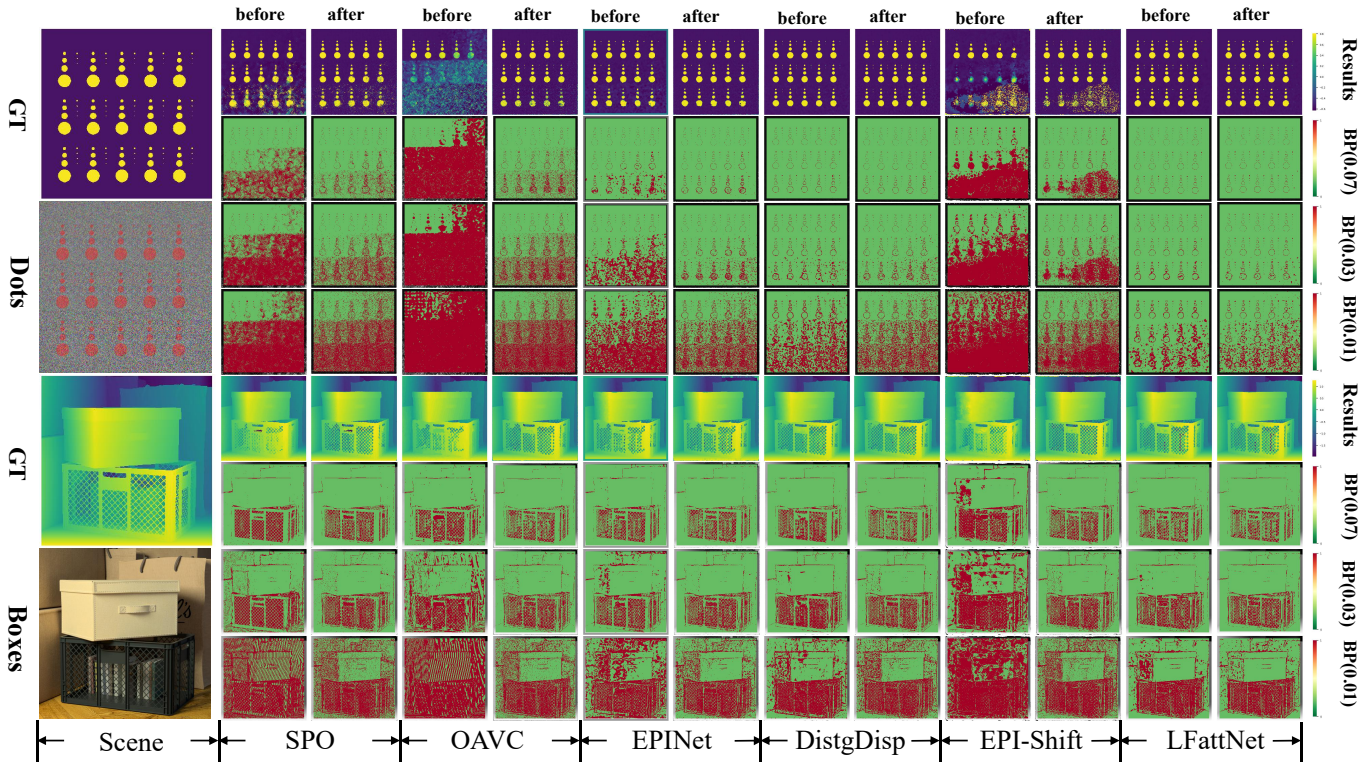


Figure 4: Refinement experiment on each algorithm selected. From top to bottom is the disparity map and errors in BadPix(0.07), BadPix(0.03), BadPix(0.01) in order. (bright color denotes large errors)

show the variation after each refinement, which is calculated as $M_{before} - M_{after}$. (red means better and blue means worse), M is the metric of result before/after.

Experiment

This section introduces the dataset we selected for experiments, then the implementation details are described. Finally, the results before/after refinement are reported, along with the generalizability proof, ablation study and evaluation on real world. more results could be found in the supplementary material.

Dataset

We use the HCI 4D Light Field Dataset in our experiments. It contains 28 light field scenes, which are partitioned into four sub-sets: 'Stratified', 'Test', 'Training' and 'Additional'. The image resolution is 512×512, and the number of sub-aperture views is 9×9. Here, we use 16 scenes in 'Additional' for training, 8 scenes from 'Stratified' and 'Training' for validating, and 4 from 'Test' for testing. While we randomly sample 48×48 image patches for training, we use the full resolution 512×512 for validation. Another, Inria Dense Light Field Dataset(Shi, Jiang, and Guillemot 2019), Stanford Dataset and Wanner Dataset(Wanner, Meister, and Goldluecke 2013) are used to further verify the generalization of BpCNet. For Inria, 'Black white','Kiwi bike','Toy friends','White roses' are selected for test. Since Stanford, Wanner are real scene dataset and have no groundtruth, we only display the visual results.

Implementation Details

To prove our theory, we select 6 typical algorithms from HCI 4D Light Field Benchmark for refinement experiments, including LFattNet, DistgDisp, EPI-Shift, EPINet, OAVC and SPO. For evaluation, we use the standard metrics in LF disparity estimation: MSE$\times 100$ and BadPix$(0.01, 0.03, 0.07)$. (smaller is better)

$$
\begin{aligned}
MSE \times 100 &= 100 \times \frac{1}{m} \sum_{i=1}^{m} (D_f^{(i)} - D_g^{(i)})^2, \\
BadPix(\alpha) &= 100 \times \frac{1}{m} \sum_{i=1}^{m} (|D_f^{(i)} - D_g^{(i)}| > \alpha).
\end{aligned} \tag{5}
$$

For parameter settings in Sec. 'Data Augmentation', K is 9 and δ is set differently: LFattNet($\delta = 0.5$), DistgDisp(1.0), EPI-Shift(1.0), EPINET(1.0), OAVC(1.0) and SPO(1.0), based on their performance (higher BadPix(α) with larger δ). We use Adam optimizer to minimize the L_1 loss, the batch size is 16 and the learning rate is 1e-3. To speed up the convergence, a simple estimation network is trained and we cascade BpCNet onto it for training 2 days, and details could be found in the supplementary material's 'Simple Estimation Network' section. Then we continue training it on each algorithm for 3 days.

Methods	SPO	OAVC	EPINet	DistgDisp	EPI-Shift	LFattNet
			BadPix(0.07)			
Backgammon	3.78-**0.31-0.05**	3.12+0.01-**0.06**	3.58+0.19-**0.11**	5.82-**0.89-0.77**	22.89-**17.53-0.85**	3.13-**0.02**
Dots	16.27-**4.84-3.90**	69.11-**49.28-10.05**	3.18-**1.43-0.28**	1.84-**0.19-0.08**	43.92-**28.18-1.74**	1.43-**0.06**
Pyramids	0.86-**0.70**+0.05	0.83-**0.56**+0.04	0.19+0.09+0.01	0.11+0.01+0.11	1.24-**0.97**+0.05	0.20-**0.00**
Stripes	14.99-**9.84-0.58**	2.90-**0.02-0.22**	2.46-**0.00-0.06**	3.91-**0.54-0.46**	22.72-**17.28-1.84**	2.93-**0.15**
Boxes	15.89-**2.18-0.46**	16.14-**4.49-0.23**	12.84-**0.13-0.56**	13.31-**1.28-0.80**	25.95-**12.62-0.54**	11.04-**0.47**
Cotton	2.59-**0.92-0.58**	2.55-**1.89-0.12**	0.51-**0.06-0.03**	0.49-**0.09-0.05**	2.18-**1.60-0.11**	0.27-**0.01**
Dino	2.18-**0.65-0.16**	3.94-**2.68-0.08**	1.29-**0.11-0.13**	1.41-**0.22-0.06**	5.96-**4.24-0.21**	0.85-**0.01**
Sideboard	9.30-**5.26-0.41**	12.42-**8.32-0.46**	4.80-**0.78-0.19**	4.05-**0.34**+0.05	11.80-**7.43-0.06**	2.87-**0.20**
Average	8.23-**3.09-0.76**	13.88-**8.40-1.41**	3.61-**0.28-0.17**	3.88-**0.44-0.26**	17.08-**11.23-0.67**	2.84-**0.12**
			BadPix(0.03)			
Backgammon	8.64–**2.76-0.77**	5.12-**0.95**+0.17	6.29+0.17-**0.21**	10.54-**1.68-1.90**	40.53-**28.54-4.75**	3.98-**0.10**
Dots	35.07-**10.48-5.63**	75.38-**36.31-16.96**	12.74-**12.14-0.08**	4.46-**0.01**+0.80	53.18-**29.79-3.60**	3.01-**0.24**
Pyramids	6.26-**4.33-1.23**	9.03-**7.59-0.67**	0.91-**0.04**+0.04	0.54-**0.05**+0.06	7.32-**6.50-0.06**	0.49-**0.02**
Stripes	15.46-**9.38-0.79**	19.88-**15.69-0.80**	3.12-**0.15**+0.08	6.89-**0.86-1.57**	47.70-**33.70-6.20**	5.42-**1.41**
Boxes	29.53-**7.59-1.86**	33.68-**14.17-0.93**	19.76-**0.64-0.39**	21.13-**2.52-0.59**	44.15-**20.70-3.01**	18.97-**1.82**
Cotton	13.71-**8.20-3.31**	20.79-**18.67-0.77**	2.31-**1.13-0.16**	1.48-**0.34-0.14**	10.68-**7.90-1.34**	0.70-**0.04**
Dino	16.36-**6.69-4.84**	19.03-**12.06-2.83**	3.45+0.09-**0.44**	4.02-**0.12-0.42**	22.15-**16.18-1.46**	2.34-**0.05**
Sideboard	28.81-**12.41-6.26**	37.83-**23.96-4.26**	12.08-**2.48-0.53**	9.58-**0.80-0.02**	36.64-**24.72-1.37**	7.24-**0.28**
Average	19.23-**7.73-3.09**	27.59-**16.81-3.39**	7.58-**2.04-0.21**	7.33-**0.80-0.41**	32.19-**21.00-2.73**	5.27-**0.50**
			BadPix(0.01)			
Backgammon	49.94–**24.53-6.40**	49.05-**23.30-9.86**	20.90+2.35-**3.64**	26.17-**1.69-4.97**	70.58-**27.55-19.09**	11.58-**0.56**
Dots	58.08-**5.43-4.92**	92.33-**22.83-19.15**	41.05-**12.89**+0.14	25.37+7.44-**1.07**	74.55-**19.62-11.46**	15.06+2.86
Pyramids	79.21-**55.94-15.02**	33.66-**11.44-0.42**	11.88-**2.74-2.98**	4.95+1.91-**3.27**	40.48-**12.27-19.80**	2.06-**0.05**
Stripes	21.88-**2.53-1.10**	28.14-**12.55-0.42**	15.67-**0.93-1.76**	19.25-**1.32-2.60**	78.95-**33.32-18.46**	18.21-**6.54**
Boxes	73.23-**17.40-9.88**	71.91-**19.99-9.43**	49.04+1.87-**8.08**	41.62+0.19-**0.95**	74.36-**21.18-8.66**	37.05-**3.41**
Cotton	69.06-**33.57-19.77**	61.35-**33.80-15.44**	28.07-**7.68-10.39**	7.59+0.67-**0.50**	46.86-**15.89-17.25**	3.64-**0.13**
Dino	69.88-**18.69-17.33**	61.82-**11.45-19.68**	22.40+3.86-**6.06**	20.46+7.27-**3.77**	64.16-**17.20-18.12**	12.22-**2.71**
Sideboard	73.37-**19.38-14.54**	73.85-**20.78-17.06**	41.88-**5.79-4.16**	28.28+2.34-**1.26**	73.42-**24.62-11.83**	20.74-**0.19**
Average	61.83-**22.18-11.12**	59.02-**19.52-11.43**	19.63-**2.74-4.61**	21.71+2.10-**2.29**	65.42-**21.46-15.58**	15.07-**1.34**

Table 1: Metrics evaluation on BadPix$(0.07, 0.03, 0.01)$ before/after refinement. The metrics before refinement are shown in the first order. Since some algorithms are refined multiple times, we show the variation after each refinement, which is calculated as $M_{before} - M_{after}$, M is the metric of result before/after. (Bloded means better result achieved)

Refinement Experiment

To prove the theory that we can improve existing algorithms by correcting bad pixels, experiments are conducted and the results before/after refinement are compared. Since BpCNet is iterable, we refine LFattNet once($\delta_1 = 0.5$), and DistgDisp, EPI-Shift, EPINET, OAVC, SPO 2 times($\delta_1 = 1.0, \delta_2 = 0.5$) by experience.

As shown in Fig. 4, the BadPix(α) of these algorithms are improved visibly after refinement except LFattNet in BadPix(0.01) of Dots, which may be caused by the incorrect δ setting in 'Dots': (too large δ for a small baseline scene with disparity range in [-0.7,0.9]).

As Tab.1 shows, the variation of metrics after each refinement are shown respectively. It can be seen that BpCNet performs well in BadPix correction, especially when the algorithm has poor BadPix(α): the BadPix$(0.07/0.03/0.01)$ of EPI-Shift are reduced by 11.90/23.73/37.04% after refinement. For those algorithms with good BadPix(α) such as LFattNet, BpCNet still works: the BadPix$(0.07/0.03/0.01)$ are reduced by 0.12/0.50/1.34%. In summary, BpCNet plays an active role for disparity map refinement, and can work on most algorithms.

Ablation Study

As shown in Tab.2, For 'Data Augmentation', we validate the effect of LFs, D_c augmentation and 'Iterative training', then find that BpCNet performs the worst without D_c. This means more diverse D_c must be provided if better results are wanted. For 'Hypothesis Disparity Generation', we study the effect of the disparity number K and find the results get better as K increases, which proves that BpCNet realizes the search process with hypothesis disparities, and K controls the search granularity. For 'Cost Volume Construction', we compared different warp functions, and experiments prove that our phase-shifted based warp function makes the best. Another, to demonstrate the necessity of modeling BpCNet as a fine-grained search strategy, we design a 'Simple Refine Model' for comparison, and it almost has no optimization as shown in Tab.2. Details could be found in the supplementary material's 'Simple Refine Model' section.

Generalizability Proof

In this section, we prove that BpCNet has good generalization, whose weights pre-trained on one algorithm/dataset can also be applied to another algorithm/dataset effectively.

BadPix(0.07) | BadPix(0.03) | BadPix(0.01)				
Algorithms		SPO	EPINet	DistgDisp
Data Augmentation	$w/o\ LFs$	-3.42|-9.81|-29.95	-0.43|-1.99|-6.98	-0.59|-1.15|-0.15
	$w/o\ D_c$	-2.33|-7.59|-26.41	-0.19|-1.49|-3.47	-0.42|-0.70|-1.19
	w/o Iterative	-3.74|-10.20|-31.54	-0.39|-2.17|-6.77	-0.66|-1.21|-0.14
Hypothesis Disparity Generation (K)	3	-1.03|-4.37|-9.95	-0.01|-0.51|-1.02	-0.10|-0.27|-0.01
	5	-2.57|-6.88|-17.33	-0.17|-1.01|-4.31	-0.27|-0.68|-0.04
	7	-3.43|-8.31|-25.46	-0.32|-1.89|-6.15	-0.59|-1.09|-0.13
Cost Volume Construction	Bilinear	-3.67|-9.62|-28.60	-0.37|-1.95|-6.03	-0.62|-1.13|-0.11
	Bicubic	-3.74|-10.33|-30.30	-0.42|-2.05|-6.74	-0.65|-1.19|-0.15
Simple Refine Model		+5.32|+11.46|+15.03	+4.26|+11.86|+40.15	+4.79|+15.20|+50.51
Full Model		**-3.85|-10.82|-33.30**	**-0.45|-2.25|-7.35**	**-0.70|-1.21|-0.19**

Table 2: Ablation study for BpCNet on Data Augmentation, Hypothesis Disparity Generation and Cost Volume Construction Module. The average of BadPix$(0.07, 0.03, 0.01)$ is shown, and the best result is bolded.

As shown in Tab.3, We apply the BpCNet trained on EPINET to other algorithms directly. Here, for OACC-Net, FastLFnet(Huang et al. 2021), CAPNet(Liu et al. 2020), FusionNet(Zhou et al. 2019a), FocalStackNet(Zhou et al. 2019b) and SPO-MO(Sheng et al. 2018), we only refine once with $\delta = 0.5$, and it can be seen that even without any specific training, BpCNet still performs well. In addition, we conduct cross-dataset experiment on Inria dataset, and the weight we use is trained on HCI Dataset with EPINet. the configuration is kept the same as Tab.3. BpCNet also work well when crossing dataset. Some visual results are shown in Fig. 6, The top two rows are FastLFnet on HCI Dataset, and the bottom two rows are OACC-Net on Inria Dataset.

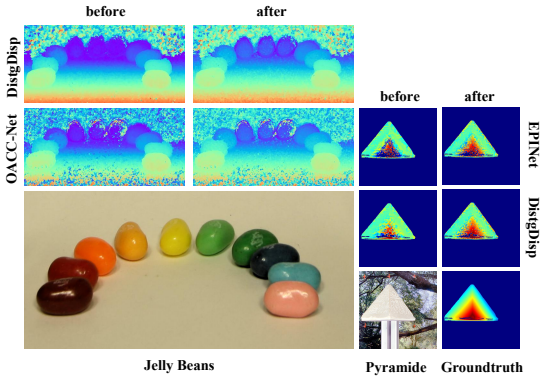


Figure 5: Visual results of real-world scenes.

Evaluation on Real World

To prove that BpCNet also works well in real world scenes, we conduct related experiments on Stanford Dataset and Wanner Dataset. As shown in Fig. 5. the results are markedly improved after refinement.

Conclusion

After reviewing previous algorithms, we propose a BadPix correction-based idea for LF disparity estimation and implement a general post-refinement network–BpCNet. We demonstrate our theory through extensive experiments.

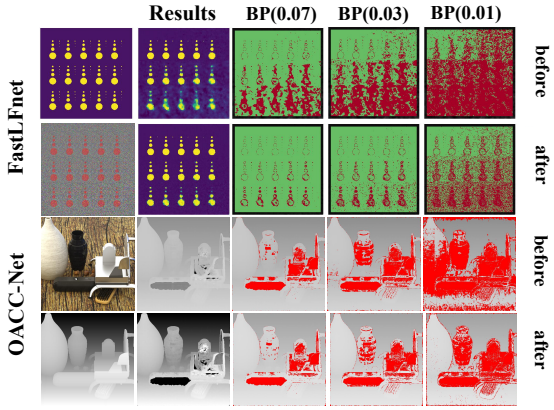


Figure 6: Visual results of Generalizability Proof.

Algorithm	BadPix			MSE
	0.07	0.03	0.01	$\times 100$
HCI 4D Light Field Dataset				
SPO	8.23-**1.46**	19.23-**5.93**	61.83-**21.03**	3.57-**0.53**
EPI-Shift	17.08-**7.51**	32.19-**16.11**	65.42-**27.55**	5.42-**1.08**
DistgDisp	3.88-**0.34**	7.33-**0.44**	21.71+0.99	1.41+**0.00**
OACC-Net	2.98-**0.24**	5.60-**0.53**	20.80-**3.42**	1.24-**0.02**
FastLFnet	8.15-**4.54**	20.80-**13.75**	54.12-**30.11**	1.75-**0.17**
CAPNet	2.86-**0.21**	4.50-**0.05**	13.88+1.26	0.87-**0.02**
FusionNet	3.74-**0.22**	6.38-**0.96**	24.64-**6.75**	2.52-**0.15**
SPO-MO	4.69-**1.01**	11.49-**4.98**	35.97-**12.87**	2.84-**0.35**
FocalStackNet	4.03-**0.18**	7.12-**1.09**	53.71-**31.90**	3.23-**0.25**
Inria Dense Light Field Dataset				
OAVC	12.57-**4.67**	30.48-**15.93**	68.76-**30.59**	6.46-**0.46**
EPI-Shift	17.95-**8.50**	35.73-**19.96**	68.55-**34.48**	12.34-**0.73**
EPINet	13.06-**3.99**	26.27-**11.98**	57.04-**27.81**	9.16-**0.45**
DistgDisp	7.87-**0.17**	13.98-**0.98**	30.46-**2.23**	8.81-**0.12**
OACC-Net	8.44-**0.52**	14.39-**1.5**	36.42-**8.95**	9.72-**0.14**

Table 3: Generalizability proof on HCI Dataset and Inria Dataset. For HCI, We apply the BpCNet trained on EPINet to other algorithms directly. For Inria, We apply the BpCNet trained on HCI Dataset to it directly.

Acknowledgements

This study is partially supported by the National Key R&D Program of China(No.2019YFB2102200), the National Natural Science Foundation of China(No.61872025), and the Open Fund of the State Key Laboratory of Software Development Environment(No.SKLSDE-2021ZX-03). Thank you for the support from HAWKEYE Group.

References

Boykov, Y.; and Funka-Lea, G. 2006. Graph cuts and efficient ND image segmentation. *International journal of computer vision*, 70(2): 109–131.

Brownrigg, D. R. 1984. The weighted median filter. *Communications of the ACM*, 27(8): 807–818.

Burkardt, J. 2014. The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 1: 35.

Chen, C.; Lin, H.; Yu, Z.; Bing Kang, S.; and Yu, J. 2014. Light field stereo matching using bilateral statistics of surface cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1518–1525.

Chen, J.; Hou, J.; Ni, Y.; and Chau, L.-P. 2018. Accurate light field depth estimation with superpixel regularization over partially occluded regions. *IEEE Transactions on Image Processing*, 27(10): 4889–4900.

Chen, J.; Zhang, S.; and Lin, Y. 2021. Attention-based multi-level fusion network for light field depth estimation. In *Proc AAAI Conf Artif Intell*, volume 35, 1009–1017.

Chen, R.; Yin, X.; Yang, Y.; and Tong, C. 2022. Multi-view Pixel2Mesh++: 3D reconstruction via Pixel2Mesh with more images. *The Visual Computer*, 1–14.

Feng, M.; Wang, Y.; Liu, J.; Zhang, L.; Zaki, H. F.; and Mian, A. 2018. Benchmark data set and method for depth estimation from light field images. *IEEE Transactions on Image Processing*, 27(7): 3586–3598.

Han, K.; Xiang, W.; Wang, E.; and Huang, T. 2021. A Novel Occlusion-aware Vote Cost for Light Field Depth Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

He, K.; Sun, J.; and Tang, X. 2012. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6): 1397–1409.

Huang, Z.; Hu, X.; Xue, Z.; Xu, W.; and Yue, T. 2021. Fast Light-Field Disparity Estimation With Multi-Disparity-Scale Cost Aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6320–6329.

Jeon, H.-G.; Park, J.; Choe, G.; Park, J.; Bok, Y.; Tai, Y.-W.; and So Kweon, I. 2015. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1547–1555.

Johannsen, O.; Honauer, K.; Goldluecke, B.; Alperovich, A.; Battisti, F.; Bok, Y.; Brizzi, M.; Carli, M.; Choe, G.; Diebold, M.; et al. 2017. A taxonomy and evaluation of dense light field depth estimation algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 82–99.

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

Le, A. V.; Jung, S.-W.; and Won, C. S. 2014. Directional joint bilateral filter for depth images. *Sensors*, 14(7): 11362–11378.

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

Liu, X.; Fu, D.; Wu, C.; and Si, Z. 2020. The Depth Estimation Method Based on Double-Cues Fusion for Light Field Images. In *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*, 719–726. Springer.

Luo, Y.; Zhou, W.; Fang, J.; Liang, L.; Zhang, H.; and Dai, G. 2017. Epi-patch based convolutional neural network for depth estimation on 4d light field. In *International Conference on Neural Information Processing*, 642–652. Springer.

Mishiba, K. 2020. Fast depth estimation for light field cameras. *IEEE Transactions on Image Processing*, 29: 4232–4242.

Neri, A.; Carli, M.; and Battisti, F. 2018. A maximum likelihood approach for depth field estimation based on epipolar plane images. *IEEE Transactions on Image Processing*, 28(2): 827–840.

Park, I. K.; Lee, K. M.; et al. 2017. Robust light field depth estimation using occlusion-noise aware data costs. *IEEE transactions on pattern analysis and machine intelligence*, 40(10): 2484–2497.

Piao, Y.; Ji, X.; Zhang, M.; and Zhang, Y. 2021a. Learning multi-modal information for robust light field depth estimation. *arXiv preprint arXiv:2104.05971*.

Piao, Y.; Zhang, Y.; Zhang, M.; and Ji, X. 2021b. Dynamic fusion network for light field depth estimation. *arXiv preprint arXiv:2104.05969*.

Sheng, H.; Cong, R.; Yang, D.; Chen, R.; Wang, S.; and Cui, Z. 2022. UrbanLF: a comprehensive light field dataset for semantic segmentation of urban scenes. *IEEE Transactions on Circuits and Systems for Video Technology*.

Sheng, H.; Wang, S.; Zhang, Y.; Yu, D.; Cheng, X.; Lyu, W.; and Xiong, Z. 2020. Near-online tracking with co-occurrence constraints in blockchain-based edge computing. *IEEE Internet of Things Journal*, 8(4): 2193–2207.

Sheng, H.; Zhao, P.; Zhang, S.; Zhang, J.; and Yang, D. 2018. Occlusion-aware depth estimation for light field using multi-orientation EPIs. *Pattern Recognition*, 74: 587–599.

Shi, J.; Jiang, X.; and Guillemot, C. 2019. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Transactions on Image Processing*, 28(12): 5867–5880.

Shin, C.; Jeon, H.-G.; Yoon, Y.; Kweon, I. S.; and Kim, S. J. 2018. Epinet: A fully-convolutional neural network using

epipolar geometry for depth from light field images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4748–4757.

Tao, M. W.; Hadap, S.; Malik, J.; and Ramamoorthi, R. 2013. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 673–680.

Tsai, Y.-J.; Liu, Y.-L.; Ouhyoung, M.; and Chuang, Y.-Y. 2020. Attention-based view selection networks for light-field disparity estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12095–12103.

Wang, S.; Sheng, H.; Yang, D.; Zhang, Y.; Wu, Y.; and Wang, S. 2022a. Extendable multiple nodes recurrent tracking framework with RTU++. *IEEE Transactions on Image Processing*, 31: 5257–5271.

Wang, T.-C.; Efros, A. A.; and Ramamoorthi, R. 2015. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 3487–3495.

Wang, W.; Lin, Y.; and Zhang, S. 2021. Enhanced spinning parallelogram operator combining color constraint and histogram integration for robust light field depth estimation. *IEEE Signal Processing Letters*, 28: 1080–1084.

Wang, Y.; Wang, L.; Liang, Z.; Yang, J.; An, W.; and Guo, Y. 2022b. Occlusion-Aware Cost Constructor for Light Field Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19809–19818.

Wang, Y.; Wang, L.; Wu, G.; Yang, J.; An, W.; Yu, J.; and Guo, Y. 2022c. Disentangling Light Fields for Super-Resolution and Disparity Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wanner, S.; and Goldluecke, B. 2012. Globally consistent depth labeling of 4D light fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 41–48. IEEE.

Wanner, S.; Meister, S.; and Goldluecke, B. 2013. Datasets and benchmarks for densely sampled 4D light fields. In *VMV*, volume 13, 225–226. Citeseer.

Williem, W.; and Park, I. K. 2016. Robust light field depth estimation for noisy scene with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4396–4404.

Yang, T.; and Tong, C. 2022. Real-time detection network for tiny traffic sign using multi-scale attention module. *Science China Technological Sciences*, 65(2): 396–406.

Yedidia, J. S.; Freeman, W.; and Weiss, Y. 2000. Generalized belief propagation. *Advances in neural information processing systems*, 13.

Zhou, W.; Wei, X.; Yan, Y.; Wang, W.; and Lin, L. 2019a. A hybrid learning of multimodal cues for light field depth estimation. *Digital Signal Processing*, 95: 102585.

Zhou, W.; Zhou, E.; Yan, Y.; Lin, L.; and Lumsdaine, A. 2019b. Learning depth cues from focal stack for light field depth estimation. In *2019 IEEE International Conference on Image Processing (ICIP)*, 1074–1078. IEEE.