Fast Depth Estimation for Light Field Cameras

Kazu Mishiba, *Member, IEEE*

***Abstract*— Fast depth estimation for light field images is an important task for multiple applications such as image-based rendering and refocusing. Most previous approaches to light field depth estimation involve high computational costs. Therefore, in this study, we propose a fast depth estimation method based on multi-view stereo matching for light field images. Similar to other conventional methods, our method consists of initial depth estimation and refinement. For the initial estimation, we use a one-bit feature for each pixel and calculate matching costs by summing all combinations of viewpoints with a fast algorithm. To reduce computational time, we introduce an offline viewpoint selection strategy and cost volume interpolation. Our refinement process solves the minimization problem in which the objective function consists of ℓ_1 data and smoothness terms. Although this problem can be solved via a graph cuts algorithm, it is computationally expensive; therefore, we propose an approximate solver based on a fast-weighted median filter. Experiments on synthetic and real-world data show that our method achieves competitive accuracy with the shortest computational time of all methods.**

***Index Terms*— Light fields, depth estimation, multi-view stereo matching, approximate solver.**

I. Introduction

A LIGHT field camera records both spatial and angular light information in a single shot; therefore, it has various applications such as image-based rendering [1], refocusing [2], and 3D reconstruction. As most of these applications use depth information, depth estimation from a light field image is an important topic for light field image processing research.

Lenslet-based light field cameras, known as plenoptic cameras, are hand-held and commercially available. They place a microlens array between the main lens and the image sensor, enabling a single light field image to be decoded as multiple stereo images, often referred to as sub-aperture images. Because the baselines are extremely small, traditional depth estimation methods for multiple large-baseline stereo images cannot produce satisfactory results for lenslet-based light field cameras. Thus, various approaches have been proposed for estimating depth in lenslet-based light field images based on multi-view stereo matching (MVSM) or epipolar-plane image (EPI) analysis [3].

Manuscript received November 28, 2018; revised September 4, 2019 and January 15, 2020; accepted January 24, 2020. Date of current version February 10, 2020. This work was supported by JSPS KAKENHI under Grant 17K17889. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gene Cheung.

The author is with the Department of Electrical and Electronic Engineering, Tottori University, Tottori 680-8550, Japan (e-mail: mishiba@tottori-u.ac.jp).

Digital Object Identifier 10.1109/TIP.2020.2970814

Similar to depth estimation methods for traditional multi-view stereo images, most of these methods consist of two stages: an initial depth estimation stage and a refinement stage. However, these methods involve high computational costs due to the many redundant viewpoints during initial estimation and as a result of solving optimization problems during depth refinement. This is limiting for light field applications such as image-based rendering and refocusing, which perform depth estimation during preprocessing and therefore require rapid depth estimation.

Shin et al. [4] proposed a deep learning-based approach that achieves state-of-the-art results in terms of both accuracy and fast computation. Compared with other conventional methods, their method is relatively fast because it directly infers depth maps using trained networks without a refinement step. However, a disadvantage of their method is that it takes a long time to train the networks and different networks are required to ensure high accuracy in different environments; e.g., different noise levels and different numbers of sub-aperture images.

In this paper, we propose an MVSM-based depth estimation method for light field cameras that boasts fast performance and high accuracy without requiring learning. The key ways we ensure fast computation are by reducing redundancy in the initial depth estimation stage and introducing an approximate solver for optimization during the refinement stage.

Our initial depth map estimation consists of the following three steps: pixel feature calculation, calculation of matching cost to construct a cost volume, and application of a winner-takes-all (WTA) scheme to the cost volume. Our method represents a pixel feature using only one bit and limits the number of viewpoints used for estimation. Although matching cost calculation using more pixel features on more viewpoints may achieve high accuracy, exceeding a certain amount of pixel features results in less accuracy improvement but higher computational cost. In other words, using fewer features can achieve sufficient accuracy. To ensure stable estimation with few pixel features, the matching cost is calculated by summing all combinations of viewpoints used then applying a cost aggregation method. A typical matching cost calculation for all combinations has poor computational efficiency due to the massive number of combinations; however, we show that the calculation of the matching cost can be accelerated using the proposed algorithm (discussed in later section). Subsequently, we employ a cost volume interpolation scheme that interpolates the cost volume in the depth direction to reduce the computational time.

Our refinement step uses an approximate solver to solve the optimization problem to remove outliers. This reduces the

computational costs compared to the traditional approach of using a graph cuts algorithm.

The main contributions of this research are as follows:

1) We propose a computationally efficient initial depth estimation method using one-bit feature calculation, fast matching cost calculation, a view selection strategy, and cost volume interpolation.
2) We propose a rapid approximate solver for refinement of the initial depth map.

The proposed method performs significantly faster than conventional methods whilst achieving competitive accuracy without the need for a training process.

II. Related Literature

Existing depth estimation methods can be categorized into four groups: MVSM-based, EPI-based, focal stack symmetry, and learning-based approaches.

A. MVSM-Based Approach

Tao et al. [5] estimated two initial depth maps using correspondences and defocus cues, respectively. Their refinement step combines the maps into a Markov Random Field (MRF) optimization process. Based on Tao's method [5], Wang *et al.* [6] proposed a depth estimation method that models occlusions and utilizes them for the initial depth estimation and refinement steps. Jeon *et al.* [7] shifted sub-aperture images using the phase shift theorem to accurately determine the sub-pixel shift. Multi-label optimization was then performed using graph cuts to enhance the initial estimation. Tomioka *et al.* [8] calculated the matching cost based on census transform to withstand degradation caused by sensor noise and vignetting. Their refinement step minimizes an objective function that consists of a data term and an edge-preserving smoothness term. Huang [9] formulated a depth estimation problem as an optimization problem using MRFs. Without initial estimation, photo-consistency is directly considered in the MRF model.

B. EPI-Based Approach

This approach estimates the slopes of the lines in the EPI, which correspond to depth. Wanner and Goldluecke [10] used a structure tensor to compute the vertical and horizontal slopes and obtain estimation reliability for each pixel. An optimized depth is obtained by minimizing the objective function, which consists of estimated depths from these slopes and the estimation reliability. This approach tends to assign high reliability to incorrect estimations in areas where the depth is discontinuous. Li *et al.* [11] refined the estimation reliability so that incorrect estimations are assigned low reliability. To remove the influence of occlusion, Zhang *et al.* [12] proposed a spinning parallelogram operator to determine the slopes. While their refinement step applies a guided filter to the cost volume instead of an optimization method to avoid high computational costs for synthetic images, their method requires a further optimization process to reduce noise in real data.

C. Focal Stack Symmetry Approach

This approach uses the property of the light field focal stack that pixel values are symmetric along the depth dimension centered at the correct depth, which holds only for non-occlusion pixels. Lin *et al.* [13] proposed the property and data consistency measure based on analysis-by-synthesis. Depth is computed by the iterative optimization framework which incorporates them. Strecke et al. [14] proposed occlusion-free partial focal stacks. Their method jointly estimates consistent depth and normal maps by using iterative optimization.

D. Learning-Based Approach

Johannsen et al. [15] constructed a light field dictionary using EPI patches and corresponding depths then estimated depth based on a sparse coding approach that solves the optimization problem. The depth estimation process of all approaches discussed in this section includes solving an optimization problem, which generally involves long computational times. Jeon et al. [16] predicted the initial depth using a random forest-based depth label prediction then applied a weighted median filter to the initial estimation, which reduces the computational time compared with solving an optimization problem. Although prediction using feature vectors is fast, calculating the feature vectors involves high computational costs. Shin et al. [4] used an end-to-end deep neural network approach, which boasted relatively fast performance. To overcome the lack of training data, they introduced light-field image-specific data augmentation. While learning-based approaches achieve high accuracy when training and prediction are performed in the same conditions, many outliers can occur when they are performed in different conditions, as shown in Section VI.

In conventional methods, a refinement step, typically based on optimization, is required to remove outliers and achieve high accuracy. However, while this requires high computational costs in conventional methods, our proposed method finds an approximate solution with low computational costs.

III. Depth Estimation Framework

A. Problem Setting

Let $s = (u, v) \in S$ be the view coordinates, where $o = (0, 0) \in S$ is the central viewpoint. Our purpose is to estimate the depth at the central viewpoint from multiple sub-aperture images I_s. In our method, we use grayscale images as input viewpoint images. The problem of recovering depth from multiple images is as follows: Given a point (x, y) on the central viewpoint, find the corresponding point (X_s, Y_s) from another viewpoint s. Because each viewpoint is on the regular grid (coordinate) and its interior camera parameters can be considered equal, the corresponding point can be formulated as

$$(X_s, Y_s) = (pu + x, pv + y), \tag{1}$$

where p is the disparity between adjacent views. In this study, we estimate discrete depth parameters

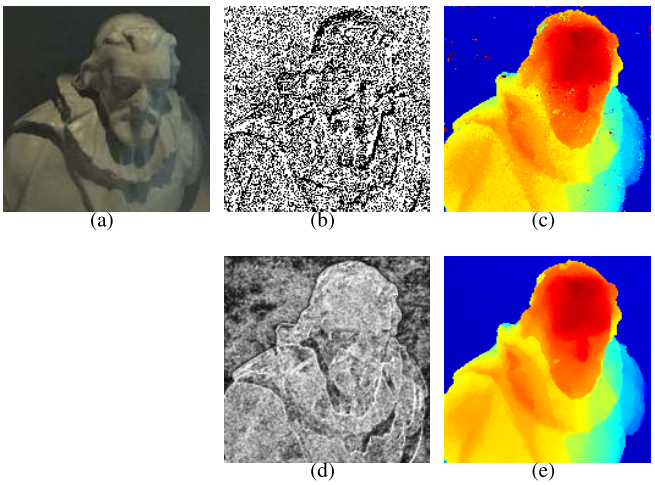

Fig. 1. Example of estimation results derived from the proposed method. (a) Center view. (b) One-bit feature map of center view. (c) Initial depth map. (d) Estimation confidence. (e) Final depth map.

$\alpha \in A = \{1, 2, \ldots, \alpha_{max}\}$, where α_{max} is the depth resolution. The relationship between α and p is defined as

$$p(\alpha) = p_{min} + \frac{p_{max} - p_{min}}{\alpha_{max} - 1}(\alpha - 1), \tag{2}$$

where p_{min} and p_{max} are the minimum and maximum disparities between adjacent views, respectively, and can either be approximately estimated from a scene or be set experimentally.

B. Initial Depth Estimation

Our proposed method estimates the initial depth map as follows: First, we calculate pixel features in each pixel on all sub-aperture images. Next, we calculate matching costs for all depth parameters. Then, cost aggregation is performed to construct a cost volume. Finally, the WTA scheme is applied to the cost volume.

Our method uses the following pixel feature at pixel i on sub-aperture image I_s for parameter α:

$$f^{\alpha}_{s,i} = \begin{cases} 1 & (e^{\alpha}_{s,i} \geq 0) \\ 0 & (\text{otherwise}), \end{cases} \tag{3}$$

where

$$e^{\alpha}_{s,i} = \nabla_x [I_{s,i}]^{\alpha}_s + \nabla_y [I_{s,i}]^{\alpha}_s \tag{4}$$

is the sum of the pixel difference in horizontal and vertical directions. Here, $[I_{s,i}]^{\alpha}_s$ is a remap operator that maps pixel $i = (x, y)$ onto $(p(\alpha)u + x, p(\alpha)v + y)$ and $I_{s,i}$ is a pixel value at i on I_s. The concept of the pixel feature is similar to the census transform. While census transform outputs a binary vector, our transform outputs only one binary for each pixel. Therefore, our transform requires less computation cost and memory to store the transformed results than the census transform. Figure 1 (b) shows a feature map of our method.

Ideally, pixel features of corresponding pixels on all viewpoints are the same for the correct α in the case of no occlusion. Our matching cost of pixel i for parameter α is defined as the sum of the absolute differences between pixel features for all combinations of viewpoints:

$$h^{\alpha}_i = \frac{1}{2} \sum_{s \in S} \sum_{t \in S \setminus \{s\}} |f^{\alpha}_{s,i} - f^{\alpha}_{t,i}|. \tag{5}$$

After cost calculation, the cost slice C^{α} is calculated to construct a cost volume C by aggregation of the matching cost using box filtering with a window size of $W_1 \times W_1$. Finally, we apply the winner-takes-all (WTA) scheme to C for each pixel to obtain the initial depth at pixel i as

$$\bar{\alpha}_i = \arg\min_{\alpha \in A} C^{\alpha}_i, \tag{6}$$

where C^{α}_i is the matching cost at pixel i for parameter α.

C. Depth Refinement

As shown in Fig. 1, the initial depth maps contain outliers; thus, we refine the initial depth maps with global optimization.

Our refinement strategy is designed to minimize the following objective function:

$$E(\alpha) = \lambda \sum_i c_i |\alpha_i - \bar{\alpha}_i| + \sum_i \sum_{j \in N(i)} w_{i,j} |\alpha_i - \alpha_j|, \tag{7}$$

where λ controls the balance between the first and second terms, c_i is the estimation confidence on pixel i, $w_{i,j}$ is the weight between pixel i and j, and $N(i)$ is a set of neighborhood pixels around pixel i. In this study, we refer to pixels inside the $W_2 \times W_2$ window centered around i but excluding i as the neighborhood pixels of i. The first term is a data term, which is weighted by the estimation confidence for each pixel. Several methods for measuring the confidence have been proposed [17]. Here, we calculate the estimation confidence using the entire cost as follows:

$$c_i = 1 - \frac{\min_{\alpha \in A} C^{\alpha}_i}{\text{ave}_{\alpha \in A} C^{\alpha}_i}, \tag{8}$$

where ave calculates the average of C_i. The proposed measure, which is similar to the Probabilistic Measure described in [17], has relatively low computational cost and achieves good performance. The second term is a smoothness term, which is weighted by the affinity between pixels i and j. We use the following affinity function:

$$w_{i,j} = \exp\left(\frac{-|I_{o,i} - I_{o,j}|^2}{2\sigma^2}\right), \tag{9}$$

where σ^2 is an affinity control parameter and $I_{o,i}$ is the intensity of pixel i on the central viewpoint. The proposed objective function uses ℓ_1 norm both in the data and smoothness terms, which efficiently removes outliers [18].

IV. Acceleration for Initial Depth Estimation

In this section, we describe three algorithms employed for faster computation: fast cost calculation, cost volume interpolation, and view selection.

A. Fast Cost Calculation

A straightforward implementation of the matching cost calculation (4) and (5) results in poor computational efficiency; however, it can be improved by transforming the equations as follows.

Equation (4) can be written as

$$e^{\alpha}_{s,i} = [\nabla_x I_{s,i} + \nabla_y I_{s,i}]^{\alpha}_{s}. \tag{10}$$

This transformation is possible if we use a space-invariant scheme for remapping, such as bilinear and bicubic interpolations. Using (10) reduces the number of additions, differentiations, and remapping operations.

In (5), calculating the differences between pixel features for all combinations of viewpoints is time-consuming because of the massive number of combinations. Fortunately, (5) can be efficiently calculated as follows. Let $F^{\alpha}_{i}(k) = \#S_k$ where # indicates the number of elements of a set and $S_k = \{s | s \in S, f^{\alpha}_{s,i} = k\}$. Because $S_0 \cup S_1 = S$, $S_0 \cap S_1 = \emptyset$, $|f^{\alpha}_{s\in S_j,i} - f^{\alpha}_{t\in S_k,i}| = 0$ for $j = k$ and $|f^{\alpha}_{s\in S_j,i} - f^{\alpha}_{t\in S_k,i}| = 1$ for $j \neq k$, (5) is rewritten as

$$\begin{aligned} h^{\alpha}_{i} &= \frac{1}{2}\{\sum_{s\in S_0}\sum_{t\in S}|f^{\alpha}_{s,i} - f^{\alpha}_{t,i}| + \sum_{s\in S_1}\sum_{t\in S}|f^{\alpha}_{s,i} - f^{\alpha}_{t,i}|\} \\ &= \sum_{s\in S_0}\sum_{t\in S_1} 1 \qquad (11) \\ &= F^{\alpha}_{i}(0)F^{\alpha}_{i}(1), \qquad (12) \end{aligned}$$

where

$$F^{\alpha}_{i}(0) = \sum_{s\in S_0} f^{\alpha}_{s,i} = \#S - F^{\alpha}_{i}(1), \tag{13}$$

$$F^{\alpha}_{i}(1) = \sum_{s\in S_1} f^{\alpha}_{s,i} = \sum_{s\in S} f^{\alpha}_{s,i}. \tag{14}$$

As a result, the matching cost for parameter α can be calculated as follows. First, pixel features of all viewpoints are summed to calculate $F^{\alpha}_{i}(1)$ as shown in (14). Then, $F^{\alpha}_{i}(0)$ is calculated by subtracting $F^{\alpha}_{i}(1)$ from the number of viewpoints, as shown in (13). Finally, $F^{\alpha}_{i}(0)$ and $F^{\alpha}_{i}(1)$ are multiplied, as shown in (12).

B. Cost Volume Interpolation

We also introduce cost volume interpolation for faster computation. In the initial depth estimation step, cost calculation in each cost slice is the main task, which occupies most of the processing time. The cost calculation for each slice repeats α_{max} times. Although using smaller α_{max} reduces the number of computations, it also decreases the estimation accuracy due to insufficient depth resolution. Because aggregated matching costs change smoothly according to the change of α, as shown in Fig. 2, the matching cost can be predicted by interpolation along the α direction. Instead of interpolating matching costs for all decimated slices, we only interpolate the matching cost around the parameter that has the minimum cost among sampled slices for computational efficiency. This approach is similar to subpixel displacement estimation [19].

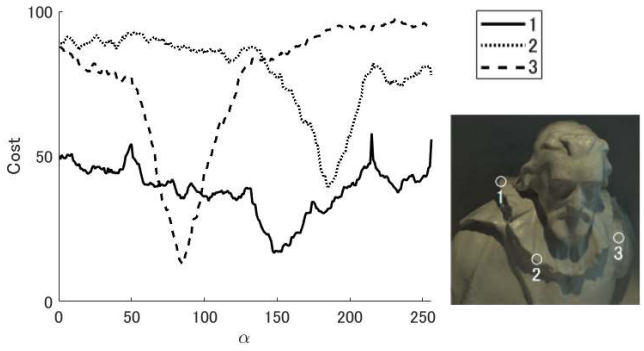

Fig. 2. Matching costs after aggregation.

We first construct a cost volume with sampled parameters $\alpha \in \hat{A} \subseteq A$ then apply the WTA scheme, described as

$$\hat{\alpha}_i = \arg\min_{\alpha\in\hat{A}} C^{\alpha}_{i}. \tag{15}$$

In this study, A is sampled at a regular interval, t; i.e., $\hat{A} = \{1, 1+t, \ldots, \alpha_{max} - t, \alpha_{max}\}$. Next, we interpolate parameter α as follows:

$$\bar{\alpha}_i = \begin{cases} \hat{\alpha}_i & (\hat{\alpha}_i = 1, \alpha_{max}) \\ \hat{\alpha}_i + \mathrm{round}(\delta^{\alpha}_{i}) & (\text{otherwise}), \end{cases} \tag{16}$$

where δ^{α}_{i} is a fitting function. Equiangular line fitting and parabola fitting are commonly used for subpixel estimation. Through experiments, we found that equiangular line fitting is more suitable for our cost interpolation than parabola fitting. Using equiangular line fitting, δ^{α}_{i} is formulated as

$$\delta^{\alpha}_{i} = \begin{cases} \dfrac{1}{2}\dfrac{C^{\hat{\alpha}_i+t} - C^{\hat{\alpha}_i-t}}{C^{\hat{\alpha}_i} - C^{\hat{\alpha}_i-t}} & (C^{\hat{\alpha}_i+t} < C^{\hat{\alpha}_i-t}) \\ \dfrac{1}{2}\dfrac{C^{\hat{\alpha}_i+t} - C^{\hat{\alpha}_i+t}}{C^{\hat{\alpha}_i} - C^{\hat{\alpha}_i-t}} & (\text{otherwise}). \end{cases} \tag{17}$$

C. View Selection

For faster computation, we limit the number of viewpoints used for depth estimation instead of using all viewpoints. Subaperture images taken by a lenslet-based light field camera have high redundancy because of the short baseline between images. It is desirable to reduce the number of viewpoints because using overly redundant viewpoints increase the computational time without improving the estimation accuracy. Our strategy is to first determine the order of viewpoints then use a certain number of viewpoints for the estimation according to this order. The number of viewpoints to be used is determined experimentally. Here, we focus on determining the order of viewpoints, which is known as view selection.

View selection has been employed for multi-view stereo images [20] and for estimating the depth from light fields [21]. Conventional view selection methods select the next view that optimizes a certain objective function based on already-selected viewpoints. Such online selection requires additional time to calculate the next view in the depth estimation procedure; therefore, we propose an offline selection strategy.

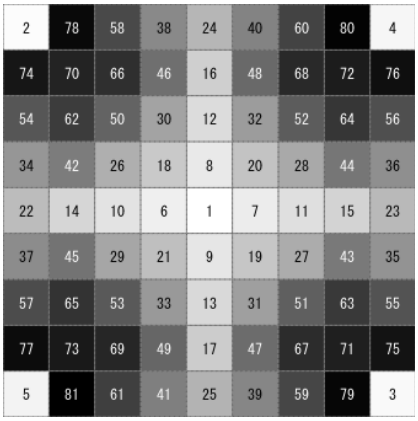

Fig. 3. Visualization of view selection. Each square corresponds to a viewpoint and the number in the square indicates the order of selection.

Our strategy uses a symmetric viewpoints selection to be independent of orientation. Let Q be a set of symmetric viewpoints q where

$$q=\begin{cases}\{(i,j),(-i,-j),(i,-j),(-i,j)\} & (i\neq 0\wedge j\neq 0)\\ \{(i,0),(-i,0),(0,i),(0,-i)\} & (\text{otherwise}).\end{cases} \tag{18}$$

Our view selection method first selects the central viewpoint then sequentially determines the optimal next symmetric viewpoints q^* by solving the following minimization problem

$$q^* = \arg\min_{q\in Q\setminus\bar{Q}} V(q), \tag{19}$$

where $\bar{Q}$ is a set of selected symmetric viewpoints and

$$V(q)=\sum_{s\in q}\left(\gamma\,\|s\|_1-\frac{1}{\#\bar{S}}\sum_{\bar{s}\in\bar{S}}\|s-\bar{s}\|_1\right), \tag{20}$$

where γ is a parameter and $\|\cdot\|_1$ is the ℓ_1 norm. The first term in the bracket in (20) seeks to select a view close to the center view to avoid occlusions. The second term in the bracket in (20) seeks to use a view far from already-selected viewpoints to avoid redundancy in view positions. When multiple solutions for (19) exist, one of them is randomly selected. Figure 3 illustrates an order of view selection in 9×9 viewpoints.

V. Fast Approximate Solver for Refinement

Although a global minimizer for (7) can be found via graph cuts [22] if α is restricted to discrete values, the computational time is high. A possible solution is to obtain an approximate solution by stopping the iteration in an iterative optimization process. Although gradient-based methods, e.g., the steepest descent method, can be used as iterative optimization methods, their rate of convergence is slow. Thus, we propose an approximate solver for (7), which, despite not guaranteeing convergence, can experimentally find approximate solutions in fewer iterations.

We introduce an auxiliary variable β and reformulate the objective function for refinement into

$$E(\alpha,\beta)=\frac{\lambda}{2}\sum_i c_i|\bar{\alpha}_i-\beta_i|+\frac{\lambda}{2}\sum_i c_i|\bar{\alpha}_i-\alpha_i| +\sum_i\sum_{j\in N(i)} w_{i,j}|\alpha_i-\beta_j|+\mu\sum_i|\alpha_i-\beta_i|, \tag{21}$$

where $\mu > 0$ is a parameter for controlling the similarity between α and β. Parameter α which minimizes (7) and (21) becomes identical with $\mu\to\infty$. We alternately solve the following problems with increasing μ:

$$\alpha^{(k+1)} = \arg\min_{\alpha} E(\alpha,\beta^{(k)}), \tag{22}$$

$$\beta^{(k+1)} = \arg\min_{\beta} E(\alpha^{(k+1)},\beta). \tag{23}$$

The above problems can be solved for each pixel. The objective function of these problems can be written as the same formulation:

$$F(x_p,y)=\frac{\lambda}{2}c_p|x_p-\bar{\alpha}_p|+\sum_{q\in N(p)} w_{p,q}|x_p-y_q|+\mu|x_p-y_p|. \tag{24}$$

Here, using $x_p=\alpha_p$, $y=\beta^{(k)}$, and $x_p=\beta_p$, $y=\alpha^{(k+1)}$ corresponds to the objective function for each pixel in Eqs. (22) and (23), respectively. Consequently, our refinement algorithm iteratively solves the following minimization problem:

$$\alpha_p^{(k+1)} = \arg\min_{\alpha_p} F(\alpha_p,\alpha^{(k)}). \tag{25}$$

We begin with $\mu=\mu_0$ and $\alpha^{(0)}=\bar{\alpha}$. In each iteration, μ is increased by multiplying $\kappa > 1$. The convergence properties of the proposed algorithm are not analyzed in this research. The solution of (25) is a weighted median [23]; we propose a method for solving (25) based on the faster weighted median filter (WMF) [24].

A formulation of the WMF is as follows. Let $\tilde{X}=\left(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{N_I}\right)$ be the sequence of all possible pixel values in an ascending order. The WMF then finds $\tilde{x}_r$ where

$$r=\min k \quad \text{s.t.} \quad \sum_{i=1}^{k}\sum_{q\in Q_i} w_{p,q}-\sum_{i=k+1}^{N_I}\sum_{q\in Q_i} w_{p,q}\geq 0, \tag{26}$$

where $Q_i=\{q\in R(p)|x_q=\tilde{x}_i\}$ and $R(p)$ is a set of pixels in a local window. $w_{p,q}$ is a weight calculated from the affinity between pixels p and q, which can be written as $w_{p,q}=g(\mathbf{f}_p,\mathbf{f}_q)$ where $\mathbf{f}_p$ and $\mathbf{f}_q$ are pixel features at p and q, and g is an affinity function, such as (9).

Next, we review the faster WMF [24]. The WMF can be regarded as finding the optimal cut point k satisfying $b_{k-1}<0$ and $b_k\geq 0$ where the balance b_k is

$$b_k=\sum_{i=1}^{k}\sum_{q\in Q_i} w_{p,q}-\sum_{i=k+1}^{N_I}\sum_{q\in Q_i} w_{p,q}. \tag{27}$$

Let H be a 2D joint-histogram denoted as

$$H(i,f)=\#\{q\in R(p)|X_q=\tilde{X}_i,\mathbf{f}_q=\tilde{\mathbf{f}}_f\}, \tag{28}$$

where $\tilde{\mathbf{f}}_f \in \tilde{F} = \left(\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_{N_F}\right)$ and $\tilde{F}$ is the sequence of all possible pixel features in ascending order. For any pixel q belonging to bin (i, f), the weight $w_{p,q}$ between pixel p and q can be computed as $g(\mathbf{f}_p, \tilde{\mathbf{f}}_f)$. Using this, b_k can be rewritten as

$$\sum_{i=1}^{k}\sum_{f=1}^{N_f} H(i,f)g(\mathbf{f}_p,\tilde{\mathbf{f}}_f) - \sum_{i=k+1}^{N_I}\sum_{f=1}^{N_f} H(i,f)g(\mathbf{f}_p,\tilde{\mathbf{f}}_f). \quad (29)$$

For fast computation to determine the optimal cut point, a balance counting box (BCB) is introduced as follows:

$$B_k(f) = \sum_{i=1}^{k} H(i,f) - \sum_{i=k+1}^{N_I} H(i,f). \quad (30)$$

Using B_k, b_k is formulated as

$$b_k = \sum_{f=1}^{N_f} B_k(f)g(\mathbf{f}_p,\tilde{\mathbf{f}}_f). \quad (31)$$

For the initial cut, b_k is calculated as in (31). If $b_k \geq 0$, k is updated to $k-1$, otherwise to $k+1$. b_k can be updated as

$$b_{k-1} = b_k - 2\sum_{f=1}^{N_f} H(k,f)g(\mathbf{f}_p,\tilde{\mathbf{f}}_f) \quad (32)$$

in the case of $b_k \geq 0$. This step is repeated until we find k satisfying $b_{k-1} < 0$ and $b_k \geq 0$. After finding the optimal cut point for a pixel, the filter window is shifted to the next pixel and the joint-histogram and BCB are updated. Because the current window and subsequent window largely overlap, only exiting and entering pixels are examined for the update of the joint-histogram and BCB. For more details, see the description of computational efficiency in [24].

Our optimization problem in (25) has a similar formation to the WMF but with slight differences. The WMF is equivalent to

$$\arg\min_{x_p} \sum_{q\in R(p)} w_{p,q}|x_p - x_q|. \quad (33)$$

To use an efficient algorithm to solve (25), we rewrite (24) using $R(p) = \{N(p) \cup p\}$ into

$$F(x_p, y) = \sum_{q\in R(p)} w_{p,q}|x_p - y_q| + \frac{\lambda}{2}c_p|x_p - \bar{\alpha}_p| + (\mu - w_{p,p})|x_p - y_p|. \quad (34)$$

The minimization of (34) corresponds to finding the optimal cut point of the following b_k:

$$b_k = \sum_{f=1}^{N_f} B_k(f)g(\mathbf{f}_p,\tilde{\mathbf{f}}_f) + \delta(\bar{\alpha}_p, k, \frac{\lambda}{2}c_p) + \delta(y_p, k, \mu - w_{p,p}), \quad (35)$$

where

$$\delta(j,k,l) = \begin{cases} l & (j \leq k) \\ -l & (j > k). \end{cases} \quad (36)$$

The first term on the right-hand side of (35) can be calculated by the same procedure as described in [24]. Calculation of the second and third terms can be easily built into the cut point update procedure as follows. For $\delta(j,k,l)$, when k is updated to $k-1$ to find the optimal cut, we substitute $2l$ from b_k if $j = k-1$ in the same manner as (32).

In our implementation, we use a coarse-to-fine strategy to speed up the convergence. The iteration is stopped when the ratio of elements whose values are changed before and after the last iteration is below a threshold τ.

VI. Experimental Results and Discussion

In this section, the performance of the proposed method is evaluated using synthetic and real-world datasets. For the synthetic experiments, we used the 4D light field benchmark [25], which contains 9×9 sub-aperture images with 512×512 spatial resolution and ground truth depth data. For the real-world experiments, we used center 9×9 sub-aperture images from light field images captured with a Lytro Illum camera and two Lytro first generation cameras. Our method was implemented in C++ and used parallel computation with CUDA for initial estimation and OpenMP for optimization. We used a Core i7-3770 @ 3.40 GHz CPU with a 12 GB RAM and NVIDIA GeForce GT 640 and set the parameters as follows: $\alpha_{max} = 256$, $\gamma = 0.8$, $\lambda = 15$, $\sigma = 10$, $W_1 = 5$, and $W_2 = 7$.

A. Algorithm Validation

In this section, we discuss the effectiveness of our proposed view selection, cost volume interpolation, and approximate solver using four light-field test images on the 4D light field benchmark [25].

1) View Selection: To verify the effectiveness of our view selection strategy, we compared the following four strategies: best view selection, random selection, static selection, and the proposed strategy. The best view selection strategy selects the next view, which minimizes the mean squared error (MSE) between the ground truth and the estimated depth. In a real situation, we cannot adopt this strategy because the ground truth data are unavailable. The random selection strategy randomly selects the next views. The static selection strategy selects central vertical, central horizontal, right diagonal, and left diagonal viewpoints first, which are used in [4], followed by the remaining viewpoints. Visualization of the order of the static selection is shown in Fig. 4. In this experiment, we used no cost volume interpolation, i.e., $t = 1$.

Figure 4 shows the relationship between the number of viewpoints used and the average MSEs of the initial estimation. The computation time of our initial estimation is also shown in the figure. The figure clearly indicates that our proposed view selection strategy achieved better performance in MSE than the random and static selection strategies. Moreover, two key findings are clear. One is that an increase in the number of viewpoints increases the computational time for the initial depth estimation but does not necessarily lead to a decrease in estimation error. The other is that using approximately 20 viewpoints achieved satisfactory results in our strategy.

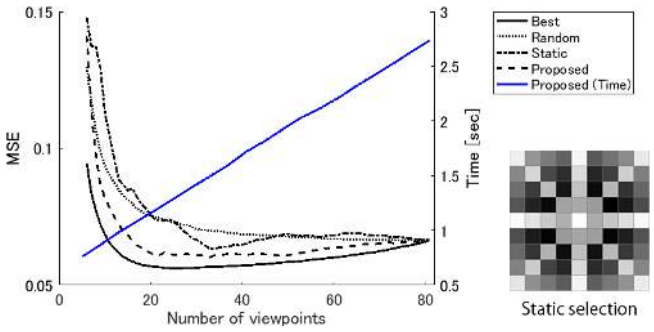

Fig. 4. Average MSEs of initial estimation for each viewpoint selection, computational times for the initial depth estimation using the proposed view selection, and the visualization of the static selection. The MSEs of the random selection is the average of 10 trials per image.

TABLE I

AVERAGE MSES AND COMPUTATIONAL TIMES FOR DIFFERENT SAMPLING RATES (t)

t	1	3	5	15	51	85
MSE($\times$100)	6.53	6.51	6.56	6.88	23.08	51.37
Time [sec]	2.74	1.15	0.81	0.52	0.41	0.38

2) Cost Volume Interpolation: To show the effectiveness of cost volume interpolation, we performed an initial estimation with different sampling rates and all viewpoints. The sampling rates were set to $t = 1, 3, 5, 15, 51, 85$. Table I lists the averages of the MSEs and computation times of the initial estimation for each t. While the MSEs for $t = 1, 3, 5$ were similar, the computational times decreased as t increased. Thus, cost volume interpolation can reduce the computational time for initial estimation without significantly reducing the accuracy.

3) Optimization: To verify the effectiveness of our proposed solver, we solved the optimization problem of (7) by the following three methods: a graph cut algorithm, the steepest descent algorithm, and our proposed method. An initial depth was obtained with all viewpoints and $t = 1$. For a clearer comparison, a coarse-to-fine strategy was not employed. The parameters for our optimization were set to $\mu_0 = 0.001$ and $\kappa = 1.2$ and the step size at each iteration on the steepest descent algorithm was determined by the Armijo rule [26]. Our method and the steepest descent algorithm used parallel computation with OpenMP and we used the GCoptimization software [27]–[29] for graph cuts.

While the steepest descent algorithm and our method took approximately 100 ms and 10 ms in each iteration, respectively, the graph cut algorithm took approximately 15 min to solve. Figure 5 represents the values of (7) over iterations for the *Cotton* scene, in which the objective function value decreased rapidly when using our proposed method.

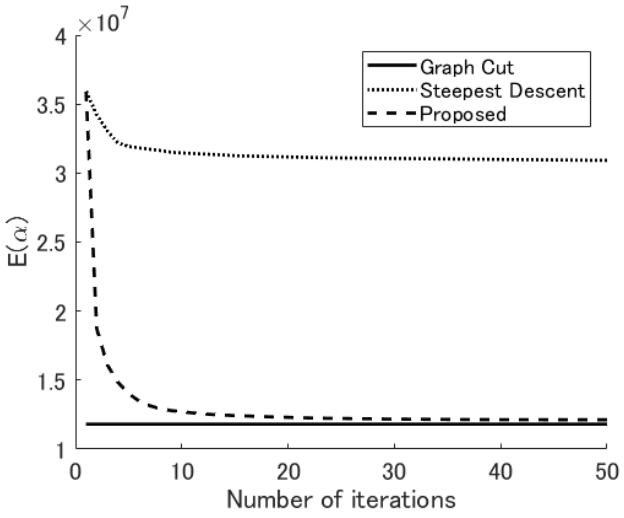

Fig. 5. Comparison of objective function variations for the *Cotton* scene, in the benchmark dataset [25].

B. Comparisons

In this section, the performance of the proposed method is evaluated using synthetic and real-world datasets. We compared our method (FDE) with the following state-of-the-art methods: EPI1 [15], EPI2 [10], PS_RF [16], RPRF-5view [9], SPO [12], LF [7], LF_OCC [6], and Epinet-fcn9x9 [4]. The names of these methods were derived from the benchmark website [25].

In our proposed optimization, the coarse-to-fine strategy starts with data downsampled by 2 then upsamples the convergence result to the original resolution. We set $\kappa = 1.2$, $\tau = 0.001$ and $\mu_0 = 0.001$ and 0.1 for the downsampled and original resolutions, respectively. From the experimental results in Section VI-A, we set $t = 5$ and used 21 viewpoints, which revealed promising results regarding fast computation and high accuracy.

1) Synthetic Data: We used the 4D light field benchmark [25] with the eight light-field test images for the quantitative evaluation.[1] The eight test images include four photorealistic scenes and four *stratified* scenes, which are designed to pose specific isolated challenges. As quality measurements, we used the MSE and the bad pixel ratio (BP), which represents the percentage of pixels whose disparity error is larger than 0.07 pixels. Estimation results, quality measurements and computational times of the conventional methods were derived from the benchmark website. Computational times of our proposed method exclude the times taken for view selection because it can be computed in advance.

The quantitative results are listed in Table II. The proposed method showed competitive performance in quantitative measurements and the best performance for computational speed. Figure 6 shows estimation results and error maps. As represented by the photorealistic *Cotton* scene, our method estimated reasonable results. We investigate the strength and weakness of our proposed method by using estimation results of *stratified* scenes.

[1]It is preferable to evaluate our method by submitting estimation results of all 12 test images to the benchmark site, including four test images whose ground truth depth maps are not disclosed. However, no new submissions were possible at the time of writing this manuscript due to the server problems. Therefore, we evaluated the eight images with ground truth using the benchmark program.

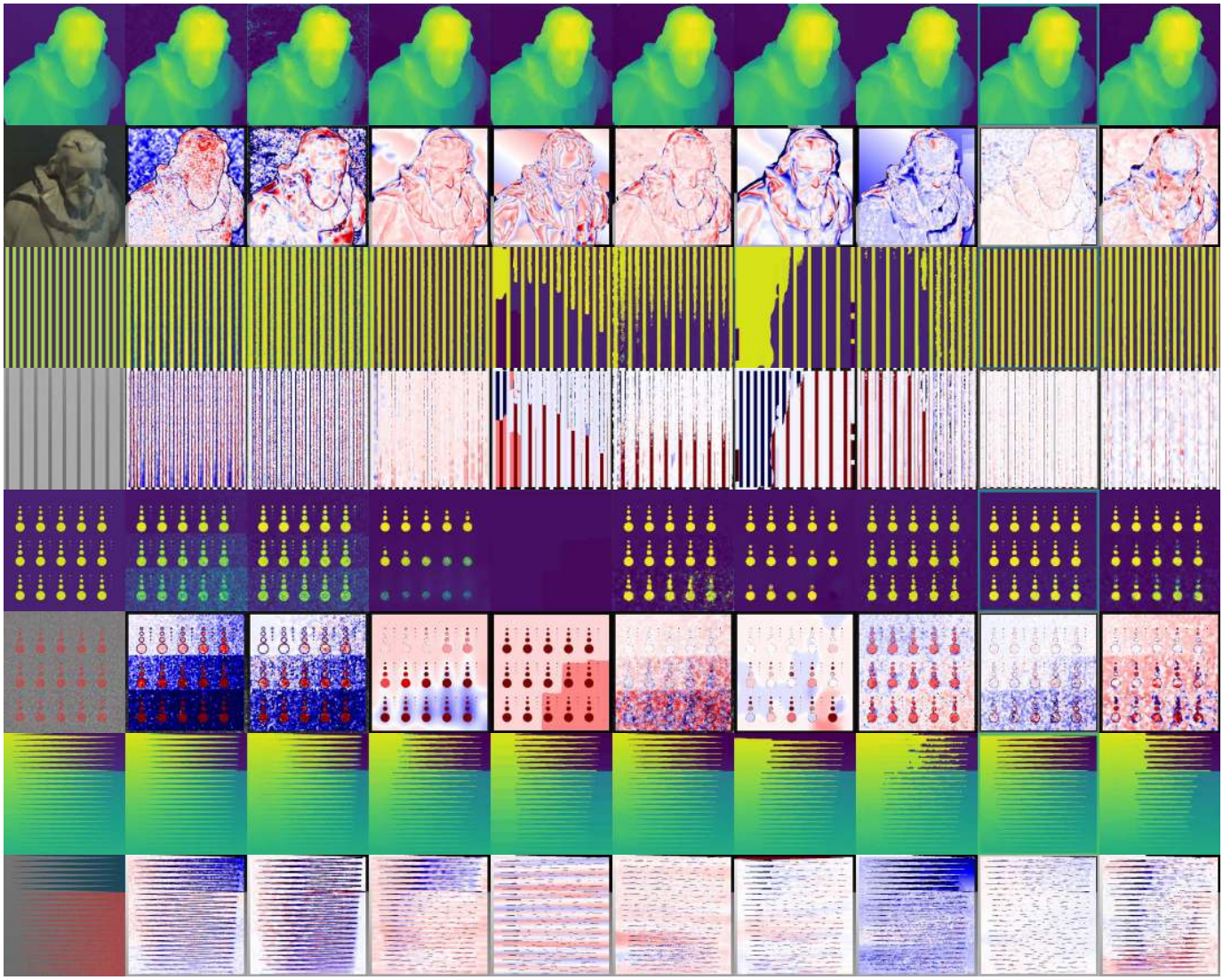

Fig. 6. Comparison of estimation results using *Cotton* scenes (1st and 2nd rows), *Stripes* scenes (3rd and 4th rows), *Dots* scenes (5th and 6th rows), and *Backgammon* scenes (7th and 8th rows) from the [25] dataset. The odd rows are the estimated depth and even rows are the error maps for MSEs. (a) Ground truth depth map and center view image. (b) EPI1 [15]. (c) EPI2 [10]. (d) PS_RF [16]. (e) RPRF-5view [9]. (f) SPO [12]. (g) LF [7]. (h) LF_OCC [6]. (i) Epinet-fcn9x9 [4]. (j) FDE (proposed method).

TABLE II

AVERAGED EVALUATION METRICS AND RANKING FOR EIGHT SYNTHETIC SCENES OF THE [25] DATASET. OUR METHOD ACHIEVES COMPETITIVE RESULTS FOR THE MEAN SQUARED ERROR (MSE) AND BAD PIXEL RATIO (BP) WITH THE SHORTEST COMPUTATIONAL TIME

Method	MSE ($\times 100$)	BP [%] (> 0.07)	Time [sec]
EPI1 [15]	4.13 (5)	22.14 (9)	86.27 (5)
EPI2 [10]	6.94 (8)	21.83 (8)	8.18 (3)
PS_RF [16]	3.69 (3)	6.96 (2)	1089.96 (7)
RPRF-5view [9]	6.03 (7)	12.93 (5)	11.75 (4)
SPO [12]	3.57 (2)	8.23 (4)	2100.63 (8)
LF [7]	8.66 (9)	16.05 (7)	1004.79 (6)
LF_OCC [6]	5.93 (6)	14.07 (6)	10915.13 (9)
Epinet-fcn9x9 [4]	**1.75** (1)	**3.58** (1)	2.03 (2)
FDE (Ours)	3.80 (4)	7.46 (3)	**1.12** (1)

The strength of our method is that it is not significantly affected by the amount of contrast and texture due to the proposed one-bit feature calculation, which can be seen from the *Stripes* scene. This scene consists of high and low contrast stripes with texture whose amount varies spatially. As can be seen from the low scores of the *Stripes* scene shown in Fig. 7 and the estimation result shown in Fig. 6, our proposed method shows decent performance on all areas, i.e., low texture, low contrast, and high contrast areas.

The weakness of our method is that it is challenging to reconstruct small geometries and occlusion boundaries due to the cost aggregation and refinement step, which can be seen from the *Dots* and *Backgammon* scenes. The *Dots* scene consists of a plane and circles which suffer from Gaussian noise whose variances varies spatially. The refinement step reduced the influence of noise in planer areas as can be seen from the low score of Background MSE shown in Fig. 7, while missing small geometries as can be seen from the high score of Missed Dots shown in Fig. 7. The *Backgammon* scene consists of two parallel background planes and one

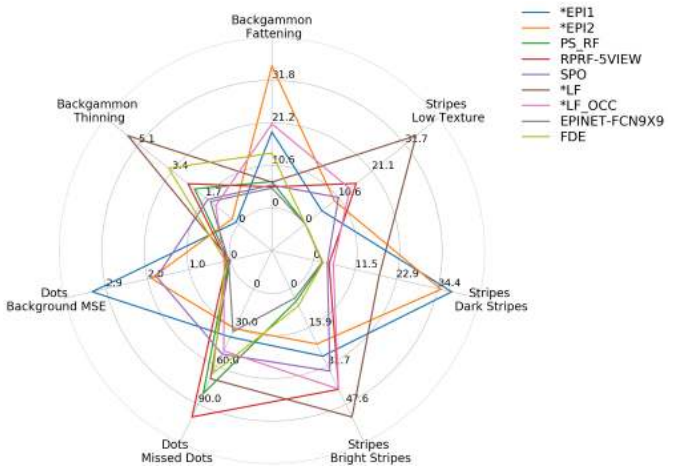

Fig. 7. The radar chart on *stratified* scenes. Lower scores represent better performance. The details of the metrics are shown in [25].

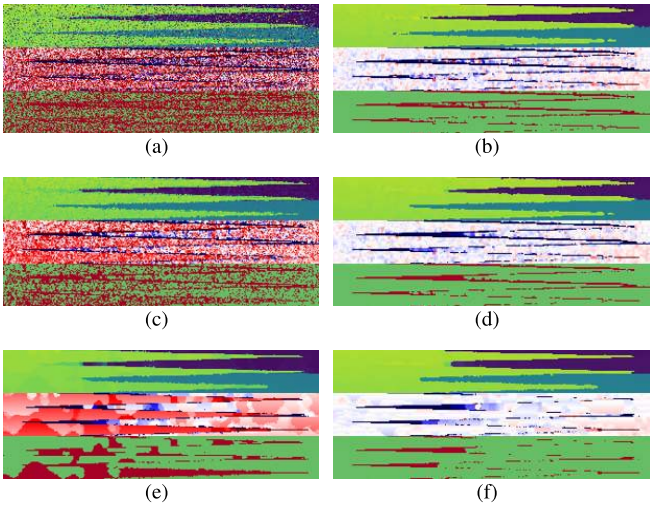

Fig. 8. Zoomed-in results of the *Backgammon* scene. In each result, 1st row is the estimated depth, 2nd row is the error map for MSE, and 3rd row is the error map for BP. Parameters (W_1, W_2) are (a) $(1, -)$, (b) $(5, -)$, (c) $(1, 3)$, (d) $(5, 3)$, (e) $(1, 7)$, and (f) $(5, 7)$.

foreground jagged plane. As can be seen from the high score of Backgammon Thinning shown in Fig. 7 and the estimation result shown in Fig. 6, accurate depth estimation on occlusion boundaries is challenging for our proposed method. For a more detailed analysis of the cost aggregation and refinement step, we performed our depth estimation method on the *Backgammon* scene with different parameters, $(W_1, W_2) = (1, -)$, $(1, 3)$, $(1, 7)$, $(5, -)$, $(5, 3)$, and $(5, 7)$, which is the default parameters. Using $W_1 = 1$ is equivalent to no cost aggregation. $W_2 = -$ represents the initial depth estimation. Figure 8 shows parts of results. Table III lists MSE, BP, Backgammon Thinning, and Backgammon Fattening of the *Backgammon* scene. Backgammon Thinning represents the percentage of the foreground that is estimated as the background. Backgammon Fattening represents the percentage of the background that is estimated as the foreground. First, we focus on the initial estimation to see the effect of the cost aggregation. As can be seen from the comparison between Fig. 8 (a) and (b),

TABLE III

EVALUATION METRICS AND RANKING FOR DIFFERENT PARAMETERS ON THE *Backgammon* SCENE

Parameter (W_1, W_2)	MSE $(\times 100)$	BP [%] (> 0.07)	Backgammon Fattening [%]	Backgammon Thinning [%]
$(1, -)$	25.47 (6)	50.90 (6)	14.24 (4)	17.58 (6)
$(1, 3)$	10.02 (2)	37.19 (5)	10.14 (2)	8.38 (5)
$(1, 7)$	**6.58** (1)	24.67 (4)	**10.08** (1)	8.18 (4)
$(5, -)$	17.33 (5)	9.31 (3)	15.48 (5)	**2.47** (1)
$(5, 3)$	17.27 (4)	9.26 (2)	15.80 (6)	2.50 (2)
$(5, 7)$	10.17 (3)	**8.57** (1)	13.67 (3)	3.55 (3)

the cost aggregation reduced noisy outliers while tending to lead to estimation error of which the background on the occlusion boundary is estimated as the foreground. This error is caused by the cost aggregation across occlusion boundaries. As can be seen from the comparison between $(1, -)$ and $(5, -)$ in Table III, using the cost aggregation exhibited a low score of Backgammon Thinning, while exhibiting a high score of Backgammon Fattening. Since the occlusion area is small relative to the entire image, an initial estimation result with the cost aggregation exhibited lower scores of MSE and BP than one without the cost aggregation. Next, we focus on the effect of the refinement step. The refinement step with larger window size removed noisy outliers more and improved MSE and BP more. In contrast, it worsened Backgammon Thinning score when $W_1 = 5$. As can be seen from the estimation results shown in Fig. 8 (d) and (f), the refinement process expanded estimation error of which the background on the occlusion boundary is estimated as the foreground, which greatly worsens MSE because the difference between the foreground and background depth is large. Using an adaptive window with occlusion prediction may help to improve estimation accuracy.

2) Real-World Data: We also conducted experiments on real-world scenes by comparing our method with those of SPO [12], LF [7], LF_OCC [6], and Epinet-fcn9x9 [4]. We used the codes provided by the authors and ran the algorithms with their default settings except for the minimum and maximum disparities on each scene, which we set manually. Real images were captured with a Lytro Illum camera by [30] and with Lytro first generation cameras by [5] and in this study.

Figure 9 compares the qualitative results. The top three images were captured with a Lytro Illum camera and the others were captured with Lytro first generation cameras. All methods except LF [7] produced reasonable results for Lytro Illum data, whereas the results of LF [7] tended to be oversmoothed. Depth estimation for Lytro first generation cameras is challenging because captured images suffer from lower resolution, more calibration error, and more severe image noise than Lytro Illum cameras. Our method produced reasonable results by removing outliers in the refinement step, as shown in Fig. 10. Although Epinet-fcn9x9 [4] exhibited robustness against noise, as the scores of *Dots* in Fig. 7 indicated, the results for images captured by Lytro first generation cameras included many outliers. This is because the trained network of Epinet-fcn9x9 was unsuitable for the images.

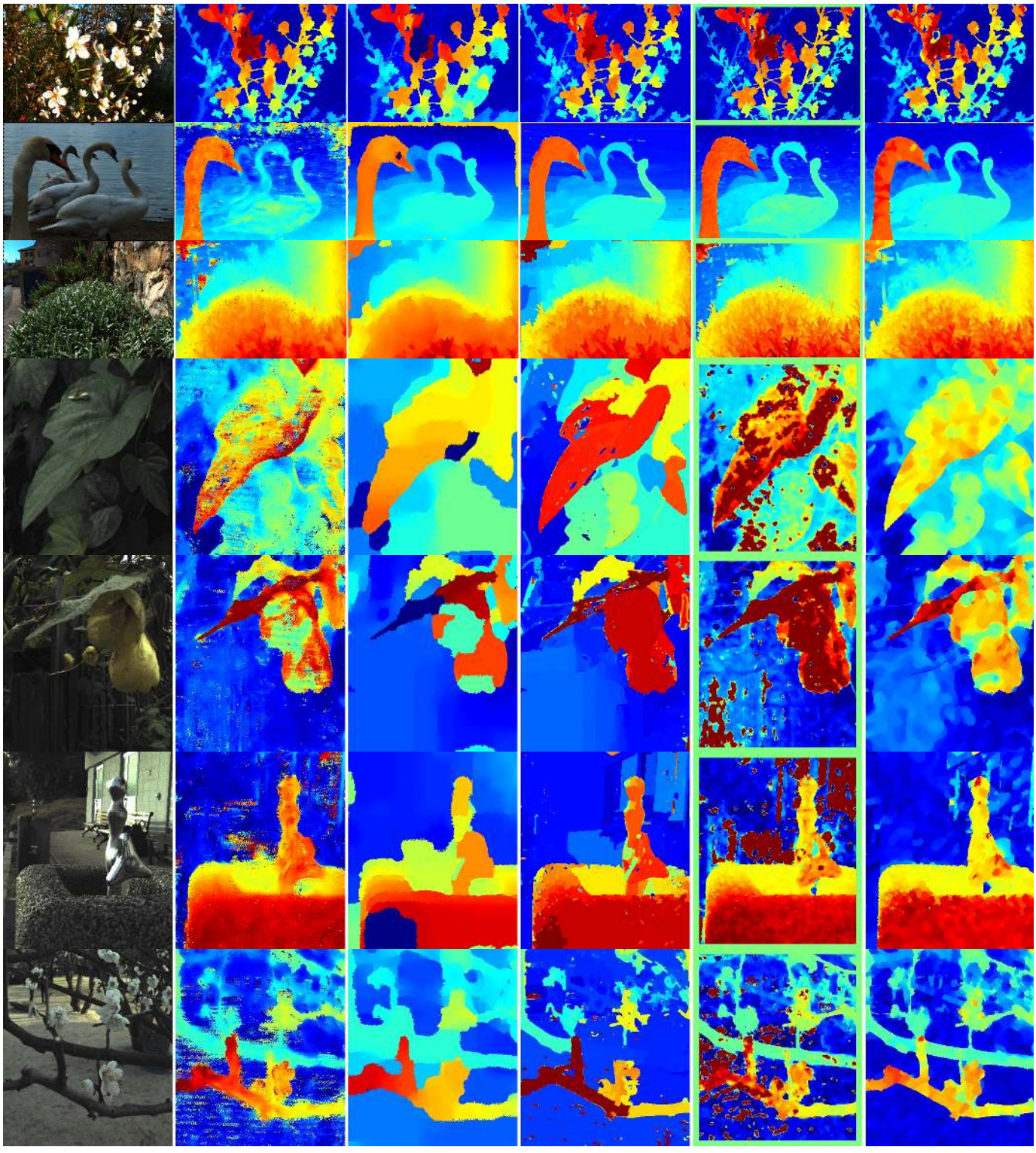

Fig. 9. Depth estimation results for selected real-world images. The top three images were captured with a Lytro Illum camera by [30] and other images were captured with Lytro first generation cameras. The fourth and fifth images are from [5] and the last two images are from this study. (a) Center view. (b) SPO [12]. (c) LF [7]. (d) LF_OCC [6]. (e) Epinet-fcn9x9 [4]. (f) FDE (proposed method).

The training data of Epinet-fcn9x9 in all the experiments only comprised synthetic data and did not include real-world data. Moreover, adapting to various conditions requires a large-scale data set that includes real-world data. However, it is difficult owing to the unavailability of the ground truth depth of real-world data.

As can be seen from the experimental results of synthetic and real data, the advantage of our proposed method is that

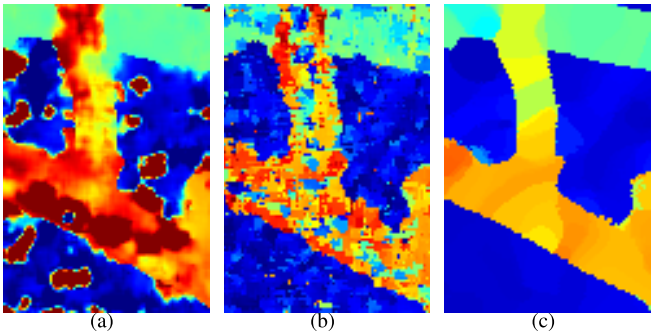

Fig. 10. Zoomed-in results of the last row image in Fig. 9. (a) Epinet-fcn9x9 [4]. (b) Initial estimation of our method. (c) After refinement of our method.

it achieves fast performance and reasonable estimation under various conditions without requiring learning.

VII. Conclusion

In this study, we proposed and verified a fast and accurate depth estimation method for light field cameras. In the initial depth estimation step, our method uses one-bit feature calculation, fast matching cost calculation, view selection, and cost volume interpolation to achieve fast initial estimation. In the refinement step, our method uses a rapid approximate solver to minimize the objective function, which consists of ℓ_1 data and smoothness terms. Experiments on synthetic data showed that our method achieved competitive accuracy with the shortest computational time when compared to other methods. The proposed method also produced plausible results in real-world scenes thanks to our refinement step.

As the proposed method is a simple MVSM approach, it has the capacity for additional processes; therefore, future work will aim to improve the estimation accuracy of the proposed method by also considering occlusions.

References

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, New York, NY, USA, 1996, pp. 31–42.

[2] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, New York, NY, USA, 2000, pp. 297–306.

[3] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 7–55, 1987.

[4] C. Shin, H.-G. 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4748–4757.

[5] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 673–680.

[6] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3487–3495.

[7] H.-G. Jeon *et al.*, "Accurate depth map estimation from a lenslet light field camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1547–1555.

[8] T. Tomioka, K. Mishiba, Y. Oyamada, and K. Kondo, "Depth map estimation using census transform for light field cameras," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2016, pp. 1641–1645.

[9] C.-T. Huang, "Robust pseudo random fields for light-field stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 11–19.

[10] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 41–48.

[11] J. Li, M. Lu, and Z.-N. Li, "Continuous depth map reconstruction from light fields," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3257–3265, Nov. 2015.

[12] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Comput. Vis. Image Understand.*, vol. 145, pp. 148–159, Apr. 2016.

[13] H. Lin, C. Chen, S. B. Kang, and J. Yu, "Depth recovery from light field using focal stack symmetry," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3451–3459.

[14] M. Strecke, A. Alperovich, and B. Goldluecke, "Accurate depth and normal maps from occlusion-aware focal stack symmetry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2529–2537.

[15] O. Johannsen, A. Sulc, and B. Goldluecke, "What sparse light field coding reveals about scene structure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3262–3270.

[16] H.-G. Jeon *et al.*, "Depth from a light field image with learning-based matching costs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 297–310, Feb. 2019.

[17] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2121–2133, Nov. 2012.

[18] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[19] M. Shimizu and M. Okutomi, "Significance and attributes of subpixel estimation on area-based matching," *Syst. Comput. Jpn.*, vol. 34, no. 12, pp. 1–10, Nov. 2003.

[20] A. Hornung, B. Zeng, and L. Kobbelt, "Image selection for improved multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[21] C. Kim, K. Subr, K. Mitchell, A. Sorkine-Hornung, and M. Gross, "Online view sampling for estimating depth from light fields," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 1155–1159.

[22] H. Ishikawa, "Exact optimization for Markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, Oct. 2003.

[23] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 3, pp. 157–192, Mar. 1996.

[24] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2830–2837.

[25] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Computer Vision—ACCV* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10113. Cham, Switzerland: Springer, 2017, pp. 19–34.

[26] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, Jan. 1966.

[27] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[28] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[29] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[30] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. Quality Multimedia Exper.*, 2016.

Kazu Mishiba (Member, IEEE) received the B.E. and M.E. degrees from Keio University, Yokohama, Japan, in 2004 and 2006, respectively, and the Ph.D. degree from Keio University, in 2011. In 2006, he joined Fujifilm Co., Ltd. He became an Assistant Professor at Keio University in 2011. He is currently an Associate Professor with Tottori University. His research interests are image processing and computer vision.