Full View Optical Flow Estimation Leveraged From Light Field Superpixel

Hao Zhu, Xiaoming Sun, Qi Zhang, Qing Wang, Antonio Robles-Kelly, Hongdong Li, and Shaodi You

***Abstract*—In this paper, we present a full view optical flow estimation method for plenoptic imaging. Our method employs the structure delivered by the four-dimensional light field over multiple views making use of superpixels. These superpixels are four dimensional in nature and can be used to represent the objects in the scene as a set of slanted-planes in three-dimensional space so as to recover a piecewise rigid depth estimate. Taking advantage of these superpixels and the corresponding slanted planes, we recover the optical flow and depth maps by using a two-step optimization scheme where the flow is propagated from the central view to the other views in the imagery. We illustrate the utility of our method for depth and flow estimation making use of a dataset of synthetically generated image sequences and real-world imagery captured using a Lytro Illum camera. We also compare our results with those yielded by a number of alternatives elsewhere in the literature.**

***Index Terms*—Depth estimation, light field, light field depth, superpixel, scene flow estimation.**

I. Introduction

LIGHT field imaging [1]–[3] has received much attention in the computer vision community due to its capacity to describe complex 3D scenes. Moreover, light field cameras such as those in [4] and [5] have been successfully applied to many applications since they can capture both, spatial and angular information, in a single shot, thus inherently encoding depth information [6]–[10]. Indeed, the efficient processing and editing of light field images and videos is of great importance for the production of visual special effects. However, most existing approaches are focused on processing a single light field input [11]–[15] with very little attention paid to continuous light field video processing [16].

Manuscript received July 25, 2018; revised December 4, 2018; accepted January 21, 2019. Date of publication February 8, 2019; date of current version January 13, 2020. This work was supported by the NSFC under Grant 61531014. The work of H. Zhu was supported by the CSC Scholarship. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivana Tosic. *(Corresponding author: Qing Wang.)*

H. Zhu, X. Sun, Q. Zhang, and Q. Wang are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zh920307@outlook.com; 18829237586@163.com; nwpuqzhang@gmail.com; qwang@nwpu.edu.cn).

A. Robles-Kelly and S. You are with the Data61, CSIRO, Canberra, ACT 2601, Australia (e-mail: antonio.robles-kelly@data61.csiro.au; shaodi.You@data61.csiro.au).

H. Li is with the Australian National University, Canberra, ACT 0200, Australia (e-mail: hongdong.li@anu.edu.au).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org. This includes the Appendix of regular paper, which contains the background and proofs, and also four videos that show the depth and optical flow results on both synthetic and real light fields. This material is 43.7 MB in size.

Digital Object Identifier 10.1109/TCI.2019.2897937

Along these lines, effective methods that can link multiple light field frames such as light field *optical flow* algorithms or *scene flow* algorithms, are somewhat under-researched since depth-information is often readily available from single-view light fields. Existing light field scene flow methods exploit the optical flow associated with the central view of a light field [17], [18]. As a result, none of them provides an effective solution to full-view light field optical flow estimation.

It is worth noting, however, that the work on two-dimensional optical flow estimation is vast [19]–[25]. These methods are all aimed at recovering the optical flow between two images and are unable to process full-view optical flow maps in light field imagery. Although these methods can be used for light field optical flow estimation view-by-view, ignoring the connection between views in light fields has the potential to preclude optical flow consistency and decrease running efficiency. An alternative to a view-by-view flow estimation is scene flow [26], which recovers the three-dimensional motion at every point in the scene.

Here we note that, although optical flow maps in light field vary from view-to-view, they share the same scene flow map while actually describing different 3D point clouds (see Fig. 5(c), (d)). Existing scene flow algorithms [27]–[32] face a similar problem, whereby full-view optical flow map computation is often complicated due to occlusion and viewpoint changes. As applied to light fields, Heber *et al.* [17] formulated a convex global energy function using the sub-aperture representations of the light field. The scene flow is then computed using a preconditioned primal-dual algorithm. Srinivasan *et al.* [18] employed oriented windows to model the 4D light field for scene flow estimation. Despite effective, these algorithms have two main disadvantages. Firstly, they are prone to occlusion. Secondly, as a result of their use of 2D or 4D patches, the estimated scene flow tends to exhibit an over-smoothed depth near object boundaries.

Here, we present a scene flow method which is capable of producing a full-view optical flow field across the whole 4D light field (see Fig. 1). Our method makes use of the recently proposed light field super-pixels (LFSPs) (also known as superrays) introduced in [33] and [34] so as to cast the full-view optical flow estimation as a scene flow recovery with a central view reference. Furthermore, the application of LFSPs allows the objects in the scene to be effectively matched across several views while permitting the recovery of a piecewise rigid optical flow map. This is as a result of a planarity constraint on the LFSPs, guaranteeing all pixels common to each LFSP sharing the same motion model. This planar representation of the LFSPs

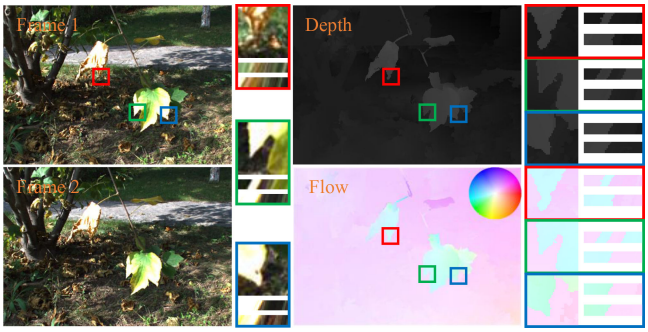


Fig. 1. Full view optical flow on real light fields. Leftmost: the central view images of two light fields as reference and target images respectively. Middle-right: the estimated depth and optical flow maps of reference image. Middle-left and Rightmost: enlarged versions of three interested regions in reference image marked with red, green, and blue squares respectively. Also the 4-D light field, depth and flow maps in horizontal, and vertical EPIs are plotted.

has the additional advantage that it supports the non-parallel-planes stereo matching approach [35], which leads to a smooth map. Finally, the occlusion points in the object boundaries can be better smoothed using other reliable un-occluded points in the LFSP [7], [8]. Here we also note that, because the light field records the rays in 3D space and each sub-aperture image can be seen as an all-in-focus image [36], it is unnecessary to consider blur effects in light field. This also reduces the non-blur effects [33], [37] since each virtual camera in the light field can be regarded as a pinhole model [36].

The paper is organized as follows. In the following section we commence by formulating the problem of recovering the scene flow, where we review LFSPs and motivate the notion that each super-pixel can be modelled using a slanted plane in 3D space so as to obtain a piecewise rigid optical flow map. We then present our approach in Section III. We show results and compare against alternatives in Section IV. In the experiments section we also introduce and motivate the use of our dataset together with other benchmarks available elsewhere. This dataset contains synthetically generated light fields, camera parameters, depth and optical flow and, upon publication, will be available on-line. We conclude on the work presented here in Section V.

II. Problem Formulation

In this section, we will show what are difficulties in full view optical flow estimation and how LFSP helps to solve it.

A. Light Field and LFSP

A 4D light field $LF(u, v, x, y)$ describes the light rays between two parallel planes, named as the uv camera plane and the xy image plane, which we refer to as angular and spatial spaces, respectively. Each (u, v) corresponds to the location of the viewpoint of light field and each (x, y) corresponds to a pixel location.

LFSP is a 4D light ray set, where all rays describe a same proximate, similar and continuous surface in 3D space [33]. Different from traditional 2D superpixel which is built on image pixels, the LFSP is defined in 4D space and each 2D slice in a fixed view corresponding to traditional 2D superpixel. Because

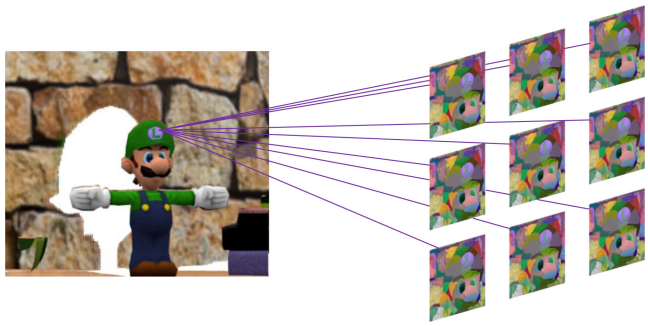


Fig. 2. Demonstration of LFSP. In the right sub-aperture images, different colors represent different LFSPs. The surrounding areas of character 'L' in hat are captured multiple times in light field. These 2-D superpixel slices in different views can be used to model the geometry of the purple areas in the scene.

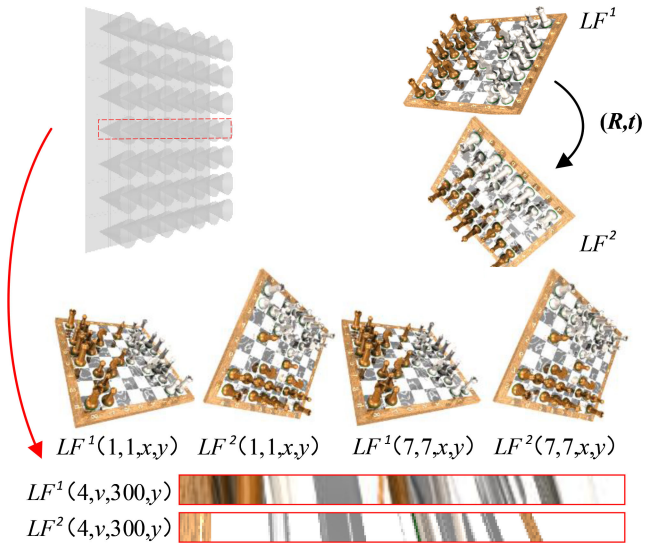


Fig. 3. Traditional methods for full view optical flow estimation. The first row, two light fields captured at different times where a motion matrix $(\boldsymbol{R}, \boldsymbol{t})$ is applied to the chessboard. The second row, straightforward method by calculating optical flow maps view by view. The third row (bottom 2 EPIs), propagating the optical flow maps along the EPI direction with the help of scene flow.

all 2D slices describe a same area and are captured from different views, LFSP can be modelled using plane or curve surface to build the geometry distribution of the area in 3D space (Fig. 2).

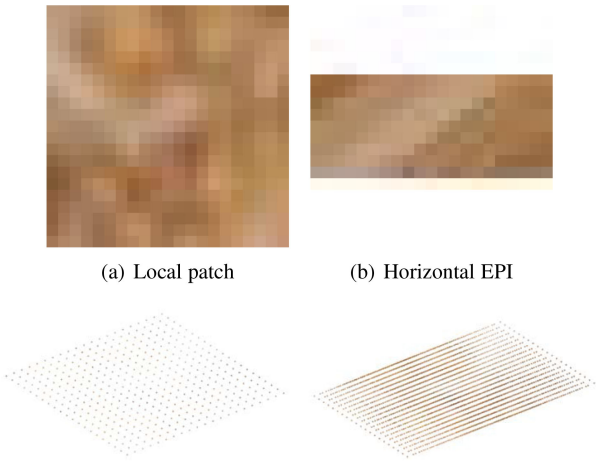
B. Formulation Model

In the first row of Fig. 3, the light field camera captures two light fields LF^1 and LF^2 at different times. To obtain full view optical maps between LF^1 and LF^2, it is straightforward to calculate the optical flow maps view by view (the second row in Fig. 3). However, it is tedious and the running time increases linearly with the increase of number of views in light field camera.

It is noticed that different views in light field are connected together according to the disparities. In a 2D light field, *i.e.* EPI (the third row in Fig. 3), each EPI line corresponds to a point in 3D space [38]. Hence, the optical flows of all pixels in this line can be calculated efficiently if the motion of the point in 3D space, *i.e.* the scene flow, is known.

(a) No occlusion (b) Occlusion (c) LFSP segmentation

Fig. 4. Propagation problem. Propagation method only holds for no occlusion case as shown in (a). (b) When the occlusion appears, there is conflict on the propagation because both red and gray EPI lines go through same areas. (c) Different colors represent different LFSPs. Each LFSP model only holds for pixels sharing a same label.



(a) Local patch (b) Horizontal EPI

(c) Central view point cloud (d) All views point cloud

Fig. 5. The discrete sampling problem of light field. (a) the local patch of central view in light field. (b) the horizontal EPI of (a). (c) the reconstructed point cloud using pixels in (a). (d) the reconstructed point cloud using all view pixels. Because the disparities of (b) are not integer values, different view of light field does not record same point cloud but a little translation. So there are more 3D points in (d) than (c).

There are two problems in the above EPI line based propagation method. Firstly, it only holds in free space and the conflict appears in occlusion areas (see Fig. 4(a), 4(b), the data comes from [39]). Secondly, due to the discrete sampling of light field, the sampled integral pixels near the EPI line do not describe a same point in 3D space (see Fig. 5, we call these pixels 'float pixels' later.), it is difficult to propagate the motion of the EPI line to these neighboring pixels.

Considering the above problems, we propose the LFSP based solution. We adopt a similar assumption as [30] and [37] that each LFSP is modelled using a slanted plane in 3D space. Because pixels in occlusion boundaries have been segmented during the LFSP processing, the conflict in Fig. 4(b) can be well handled by assigning motion to the pixels which share a same LFSP label (Fig. 4(c)). Secondly, the discrete sampling problem can be solved because the float pixels in LFSP can all be described by a slanted plane function.

In the proposed solution, each LFSP s_i is modelled as a slanted-plane in 3D space. Suppose that the normal of superpixel slice $s_i^{0,0}$ in central view $(0,0)$ is $\boldsymbol{n}_i^{0,0} = (a,b,c)^\top$ and the plane function is $\boldsymbol{n}_i^{0,0\top}\boldsymbol{X} = 1$ ($\boldsymbol{X} \in \mathbb{R}^3$), the normal of slice in the (u,v) view $s_i^{u,v}$ can be computed as,

$$\boldsymbol{n}_i^{u,v} = \frac{\boldsymbol{n}_i^{0,0}}{1 - a \cdot bl \cdot u - b \cdot bl \cdot v}, \tag{1}$$

where bl is the baseline in light field camera. In our formulation, we adopt a same assumption as previous papers on light field calibration [40]–[42] or benchmark [43], [44] that, a light field camera is equivalent to a camera array where all cameras have the same intrinsic matrix. Please see the supplemental material for obtaining $\boldsymbol{K}$, fl and bl of a real light field camera. With the slanted-plane assumption, the motion of each LFSP s_i is described by a rotation matrix $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ and a translation vector $\boldsymbol{t}_i \in \mathbb{R}^3$[1]. Given equivalent intrinsic matrix $\boldsymbol{K}$ for the central view of light field camera, for each pixel $p = (u,v,x,y) \in s_i$, the disparity $d(p,s_i)$ and flow $f(p,s_i)$ can be computed as,

$$\begin{aligned} d(p,s_i) &= bl \cdot fl \cdot \boldsymbol{n}_i^{u,v\top}\boldsymbol{K}^{-1}(x,y,1)^\top \\ f(p,s_i) &= \boldsymbol{K}(\boldsymbol{R}_i - \boldsymbol{t}_i \cdot \boldsymbol{n}_i^{u,v\top})\boldsymbol{K}^{-1}(x,y,1)^\top - (x,y,1)^\top, \end{aligned} \tag{2}$$

where fl is the equivalent focal length of light field camera. Noting that, it is difficult to obtain the focused depth of camera, which plays an important role to convert the disparity to real depth. We skip this question by setting the focused depth of camera as infinity, please refer the appendix for details of obtaining these intrinsic parameters.

III. The Proposed Method

In the proposed scene flow framework, the depth and flow are optimized sequentially where optimal techniques can be chosen for each task. This idea is suitable especially in light field configuration, because there are huge amounts of views in the disparity[2] term and only two images in the flow term. With the huge amount of views in light field, modern approaches have demonstrated surprising performance in depth estimation and the accuracy has achieved 0.02 sub-pixel [44], [45], however the flow estimation still stays at pixel level accuracy [46]. Consequently, we optimize the depth and flow sequentially.

A. Notations

With a given 4D light field $LF(u,v,x,y)$, it can be sheared [36], [47] using,

$$LF_d(u,v,x,y) = LF_0(u,v,x-ud,y-vd) \tag{3}$$

where LF_0 is the input 4D light field and LF_d is the sheared light field at the disparity d. For simplicity, the color of pixel $p = (x,y)$ in the view (u,v) is denoted as $A_{p_d}(u,v)$ when the light field is sheared at the disparity d, *i.e.* $A_{p_d}(u,v) := LF_d(u,v,x,y)$.

B. Depth Optimization

The energy function for depth optimization is,

$$E(\boldsymbol{n}) = \sum_{s_i \in S} E^d_{data}(s_i, \boldsymbol{n}_i) + \lambda^d_s \sum_{s_i \backsim s_j \in S} E^d_{smo}(\boldsymbol{n}_i, \boldsymbol{n}_j), \tag{4}$$

[1]The motion $\boldsymbol{R}_i, \boldsymbol{t}_i$ are constants while the normal $\boldsymbol{n}_i^{u,v}$ changes with views.
[2]In this paper, we will use disparity and depth interchangeably.

where S is the LFSP set and s_i, s_j are neighbouring LFSPs. ($\lambda_s^d = 0.286$). The data term is,

$$E_{data}^d(s_i, \boldsymbol{n}_i) = 1 - e^{-\frac{C(s_i, \boldsymbol{n}_i)^2}{2\sigma_d^2}}$$
$$C(s_i, \boldsymbol{n}_i) = \frac{1}{|s_i^{0,0}|} \sum_{p \in s_i^{0,0}} \frac{1}{|\Omega_p^{uocc}|} \sum_{u,v} |A_{p_{d(p,s_i)}}(u,v) - A_{p_{d(p,s_i)}}(0,0)|, \quad (5)$$

where $|\cdot|$ denotes the size of the set $\cdot$, $(u,v) \in \Omega_p^{uocc}$, Ω_p^{uocc} denotes the un-occluded views for each pixel p in light field, and σ_d ($= 2$) controls the sensitivity of the function to a large distance. The main reason for choosing a Gaussian-like metric is that it is more easier to jump out of a local minima than traditional L_2 penalty function in the global optimization [48]. Noting that, only the un-occluded views are selected for cost computation. Different from [8] where the un-occluded views are selected by the K-means clustering [49] pixel by pixel, we select the background areas as un-occluded views directly for each boundary pixel according to the superpixel segmentation in central view. As a result, the un-occluded views selection process can be sped up.

The smoothness term consists of two parts,

$$E_{smo}^d(\boldsymbol{n}_i, \boldsymbol{n}_j) = \omega_{ij}(E_{orient}(\boldsymbol{n}_i, \boldsymbol{n}_j) + E_{depth}(\boldsymbol{n}_i, \boldsymbol{n}_j))$$
$$E_{orient}(\boldsymbol{n}_i, \boldsymbol{n}_j) = \theta_1 \rho_{\tau_1}\left(1 - \frac{|\boldsymbol{n}_i^\top \boldsymbol{n}_j|}{\|\boldsymbol{n}_i\|\|\boldsymbol{n}_j\|}\right)$$
$$E_{depth}(\boldsymbol{n}_i, \boldsymbol{n}_j) = \theta_2 \frac{1}{|\mathcal{B}_{ij}|} \sum_{p \in \mathcal{B}_{ij}} \rho_{\tau_2}(d(p, s_i) - d(p, s_j)), \quad (6)$$

where $\omega_{ij} = e^{-\frac{[LF(s_i^{0,0}) - LF(s_j^{0,0})]^2}{2\sigma_{cd}^2}}$. $LF(s_i^{0,0})$ denotes the mean color of $s_i^{0,0}$ and σ_{cd} ($= 14$) controls the sensitivity between neighboring superpixels. E_{orient} controls neighboring LFSPs to orient in the same direction and E_{depth} controls sharing boundaries to be co-aligned in the disparity space ($\theta_1 = 20, \theta_2 = 7.5$). $\mathcal{B}_{ij}$ is the set of sharing boundaries between the LFSP slices $s_i^{0,0}$ and $s_j^{0,0}$. $\rho_\tau(x)$ denotes the truncated L_1 penalty function $\rho_\tau(x) = min(|x|, \tau)$ ($\tau_1 = 0.4$, $\tau_2 = 2$).

C. Flow Optimization

Based on the optimized normal $\boldsymbol{n}_i$ for each LFSP s_i in the LF^1, the goal of flow optimization is to find the best motion $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$. The energy function is defined as follows,

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{s_i \in S} E_d^f(s_i, \boldsymbol{R}_i, \boldsymbol{t}_i) + \lambda_s^f \sum_{s_i \backsim s_j \in S} E_s^f(\boldsymbol{R}_i, \boldsymbol{R}_j, \boldsymbol{t}_i, \boldsymbol{t}_j), \quad (7)$$

where $\lambda_s^f = 1/70$.

The data term is,

$$E_d^f(s_i, \boldsymbol{R}_i, \boldsymbol{t}_i) = 1 - e^{-\frac{C(s_i, \boldsymbol{R}_i, \boldsymbol{t}_i)^2}{2\sigma_f^2}}$$
$$C(s_i, \boldsymbol{R}_i, \boldsymbol{t}_i) = \frac{1}{|s_i^{0,0}|} \sum_{p \in s_i^{0,0}} |LF^1(p) - LF^2(p + f(p, s_i))|, \quad (8)$$

Algorithm 1: The Optical Flow Estimation Algorithm.

Input: 4D light fields LF^1 and LF^2
Output: 4D flow f and depth d of LF^1

1: **for** $i = 1$ to 2 **do**
2: $(S^i, d^i) = LFSP(LF^i)$
3: $\boldsymbol{n}_{ini}^i = NormalFitting(S^i, d^i)$
4: $\boldsymbol{n}_{opt}^i = NormalOptimize(\boldsymbol{n}_{ini}^i, S^i, LF^i)$ ▷ Eqn. (4)
5: **end for**
6: $f_{ini} = FlowEstimation(LF^1(0,0,x,y), LF^2(0,0,x,y))$
7: $(\boldsymbol{R}_{ini}, \boldsymbol{t}_{ini}) = MotionFitting(\boldsymbol{n}_{opt}^1, \boldsymbol{n}_{opt}^2, S^1, S^2, f_{ini})$
8: $(\boldsymbol{R}_{opt}, \boldsymbol{t}_{opt}) = MotionOptimize(\boldsymbol{n}_{opt}^1, \boldsymbol{R}_{ini}, \boldsymbol{t}_{ini}, LF^1, LF^2)$ ▷ Eqn. (7)
9: $(f, d) = ReadRTN(S^1, \boldsymbol{n}_{opt}^1, \boldsymbol{R}_{opt}, \boldsymbol{t}_{opt})$ ▷ Eqn. (1), (2)

where $\sigma_f = 3$ and other symbols have the same meaning as Eqn. (5). It is noticed that the cost of flow just considers the matching between the 2D central view slices in the LF^1 and LF^2. There are two reasons. First, the central view slices can represent the whole LFSP [33]. Adding more matching in other views can not improve the performance but may increase the running time. Second, the light field camera has a low signal-to-noise ratio, especially in the boundary views, which will lead to an unreliable matching.

The smoothness term is,

$$E_s^f(\boldsymbol{R}_i, \boldsymbol{R}_j, \boldsymbol{t}_i, \boldsymbol{t}_j) = \omega_{ij} \frac{1}{|\mathcal{B}_{ij}|} \sum_{p \in \mathcal{B}_{ij}} \rho_{\tau_3}(f(p, s_i) - f(p, s_j)), \quad (9)$$

where $\tau_3 = 150$. Noting that, the weight term ω_{ij} is determined by combining the color and optimized disparity here,

$$\omega_{ij} = e^{-\frac{[LF^1(s_i^{0,0}) - LF^1(s_j^{0,0})]^2 + [\bar{d}(s_i^{0,0}) - \bar{d}(s_j^{0,0})]^2}{2\sigma_{cf}^2}}, \quad (10)$$

where $\sigma_{cf} = 13$ and $\bar{d}(s_i^{0,0})$ denotes the normalized mean disparity of $s_i^{0,0}$. The depth term here can decrease the weight when the foreground and background share the similar textures and yield sharper flow fields in object boundary areas. In real light fields experiments, we will show the influence of the depth term in Eqn. (10).

D. Algorithm

The full algorithm is summarized in Algo.1 which contains two parts. In the depth stage (lines 1-5), first, LFSPs are obtained using [33] while a 2D depth map is produced. Then, an initial normal $\boldsymbol{n}_{ini}^{\{1,2\}}$ for each LFSP is fitted. Finally, the normal is optimized by minimizing Eqn. (4). In the flow stage (lines 6–8), we first estimate the 2D optical flow f_{ini} between two central views of LF^1 and LF^2 using [50], then the motion $(\boldsymbol{R}_{ini}, \boldsymbol{t}_{ini})$ are fitted using the optimized $\boldsymbol{n}_{opt}^{\{1,2\}}$ and the initial f_{ini}. Finally the motion is optimized by minimizing Eqn. (7).

Algorithm 2: Iteration Strategy.

Input: $\boldsymbol{h}_{ini}, \boldsymbol{h}_{ini} \in \{\boldsymbol{n}_{ini}, (\boldsymbol{R}_{ini}, \boldsymbol{t}_{ini})\}$
Output: $\boldsymbol{h}_{opt}, \boldsymbol{h}_{opt} \in \{\boldsymbol{n}_{opt}, (\boldsymbol{R}_{opt}, \boldsymbol{t}_{opt})\}$
1: $\boldsymbol{h}_{opt} = \boldsymbol{h}_{ini}$
2: **for** $iter = 1$ to $maxiter$ **do**
3: $\mathcal{H} = HypothesisGeneration(\boldsymbol{h}_{opt})$
4: $\boldsymbol{h}_{opt} = Optimization(\mathcal{H})$ ▷ Eqn. (4), (7)
5: **end for**

The energy functions in Eqns. (4) and (7) are solved using the tree-reweighted message passing (TRW-S) [51]. To make the normal and motion converge to the optimum, we use an iterative strategy (described in Algo.2). In each iteration, a hypothesis set $\mathcal{H}$ ($|\mathcal{H}| = 60$) for each LFSP is generated by three approaches, *i.e.*, retaining itself, adding Gaussian variation to its optimal value and selecting optimal values from its neighbors in previous iteration. Finally, the depth and motion are optimized by minimizing Eqns. (4) and (7) again in the next iteration until the loop ends.

IV. Experiments

A. Dataset

For our experiments, we have used real-world images acquired using a Lytro Illum camera and synthetic light fields generated in-house.[3] Note that, since the publication of [2], which describes the two-parallel-planes model for light field image capture, many datasets have been published. These are often acquired making use of various plenoptic cameras [4], [52]–[54] or generated using computer graphics techniques [7], [43], [44]. However, real-world light field image datasets are often devoid of depth ground truth, whereas synthetic image benchmarks tend to focus mainly on depth estimation tasks. Although the datasets in [7], [43], [44] can also be considered as light field optical flow benchmarks, they can not be used to evaluate the performance of algorithms accurately due to the depiction of static scenes. Moreover, the camera motion on these datasets is very contrived, being only in the horizontal and vertical axes. An exception to this is the data generated by Srinivasan *et al.* [18], which also provides ground truth. However, this dataset only contains 3 groups and provides the ground truth only for the 2D central view image. Additionally, the camera parameters for the dataset in [18] are not provided, which limits the flow computation to the 2D parallel-planes assumption [29].

As a result, here we have generated light fields for 5 scenes from Blender [55] with different types of motion, including translation and rotation as well as large scale movement. The details of our larger light fields are shown in Table I. We have also included camera egomotion and recorded the camera parameters for future reference. For each scene, we generate both high and low resolution (zoom out by a factor of 2) light fields for evaluation. Each light field has a 9×9 angular resolution. Although only two central views (the views (5, 5) in both two light fields) are used in our algorithm, we have catered for future methods which may use full 4D data for light field scene flow estimation. The availability of lower resolution light fields may also be desirable when evaluating methods with full 4D data input (high computational cost). For example, the method in [18] requires about 107G RAM on our larger images when applied to the full 4D data. Finally, note that the constants $\sigma_d = 2$, $\lambda_s^f = 1/70$, $\sigma_c = 14$, $\sigma_f = 3$, $\tau_3 = 150$ and $\sigma_c = 13$ used by our method have been set by cross validation and are kept fixed for all experiments on both, real and synthetic imagery.

[3]Upon publication, this dataset will be made available on-line

TABLE I
Details of the Synthesized Light Fields

	Resolution	Disparity	Max. Flow	Density
NewSecretaire	$9 \times 9 \times 540 \times 920$	$[1.49, 3.63]$	77.57	100%
Mario	$9 \times 9 \times 720 \times 1024$	$[1.14, 4.29]$	69.16	100%
Drawing	$9 \times 9 \times 760 \times 760$	$[0.38, 2.43]$	93.33	100%
Balls	$9 \times 9 \times 720 \times 1024$	$[2.23, 5.41]$	59.27	100%
NewBalls	$9 \times 9 \times 540 \times 920$	$[1.45, 3.63]$	75.68	100%

B. Results

We compare our results with the state-of-the-art non-learning based algorithms both in light field depth estimation and scene flow estimation, which include the globally consistent depth labeling (GCDL) [6], phase-shift based depth estimation (PSD) [56], occlusion-aware depth estimation (OADE) [7], large displacement optical flow (LDOF) [50], scene flow estimation with piecewise rigid scene model (PRSM) [30], object scene flow (OSF) [31] (the PRSM and OSF are two state-of-the-arts in Kitty Benchmark [46]) and oriented light field window for scene flow (OLFW) [18]. Noting that all results are obtained by running their released codes. The code of OSF reports error when running the small data of 'Drawing' so that it is not listed. Since traditional methods (PRSM and OSF) are designed for wider baseline stereo configuration, these algorithms may not be suitable for light field narrow baseline environment. For the sake of fairness, the central and the rightmost views of input light fields (the baseline is magnified by 4 times) are selected for these methods. Apart from the comparison of the optical flow map between the central views of two light fields (the term '2D' in Table II and III), we also provide the mean results of full views (the term '4D' in Table II and III).

1) Synthetic Data: **Iteration.** Fig. 10 shows the errors with the iteration number. The root of mean square error (RMSE) of depth and endpoint error (EPE) of optical flow decreases with iteration. Our initial values are obtained based on previous methods [7], [50], and the errors of the first iteration is very close to these two methods. With the increase of the iteration, the errors decrease rapidly. The curves of disparity and optical flow tend to be stable after 4 and 6 iterations, respectively. It can be found that the optimizations of depth and optical flow converge quickly at the beginning and tend to be stable after about 8 and 15 iterations, respectively. To ensure the algorithm converges to the optimum, the numbers of iterations are set 10 and 15 for depth and flow optimization respectively.

TABLE II
rms ERRORS OF ESTIMATED DISPARITIES FOR ALL PIXELS

	NewSecretaire		Mario		Drawing		Balls		NewBalls	
	Small	Big	Small	Big	Small	Big	Small	Big	Small	Big
GCDL [6]	0.134	0.123	0.176	0.273	0.084	0.067	0.277	0.211	0.092	0.069
PSD [56]	0.350	0.136	0.543	0.092	0.115	0.119	0.595	0.111	0.148	0.077
OADE [7]	0.193	0.138	0.196	0.165	0.074	0.068	0.245	0.111	0.172	0.079
PRSM [30]	0.136	0.125	0.139	0.102	0.079	0.061	**0.051**	**0.036**	0.059	0.048
OSF [31]	0.131	0.120	0.216	0.103	—	0.141	0.062	0.053	0.068	0.061
OLFW [18]	0.147	0.126	0.188	0.123	0.097	0.067	0.094	0.064	0.077	0.057
Ours (2D)	**0.110**	**0.080**	**0.136**	**0.073**	**0.058**	**0.039**	0.068	**0.036**	**0.051**	**0.041**
Ours (4D)	0.123	0.088	0.145	0.082	0.062	0.045	0.090	0.038	0.059	0.044

TABLE III
ENDPOINT ERRORS OF ESTIMATED FLOWS FOR ALL PIXELS

	NewSecretaire		Mario		Drawing		Balls		NewBalls	
	Small	Big	Small	Big	Small	Big	Small	Big	Small	Big
PRSM [30]	1.261	1.809	1.120	1.395	1.257	3.093	0.289	0.495	0.670	0.892
OSF [31]	0.877	1.513	2.749	6.239	—	4.355	0.744	1.613	0.547	0.794
LDOF [50]	3.780	3.174	2.136	4.524	1.129	1.766	0.440	0.587	1.259	1.883
OLFW [18]	2.265	4.441	4.893	7.166	2.495	5.005	1.167	6.073	1.206	13.713
Ours (2D)	**0.781**	**1.393**	**0.826**	**1.012**	**0.928**	**1.324**	**0.284**	**0.481**	**0.496**	**0.693**
Ours (4D)	1.337	1.853	1.174	1.299	1.019	1.427	0.397	0.555	0.588	0.807

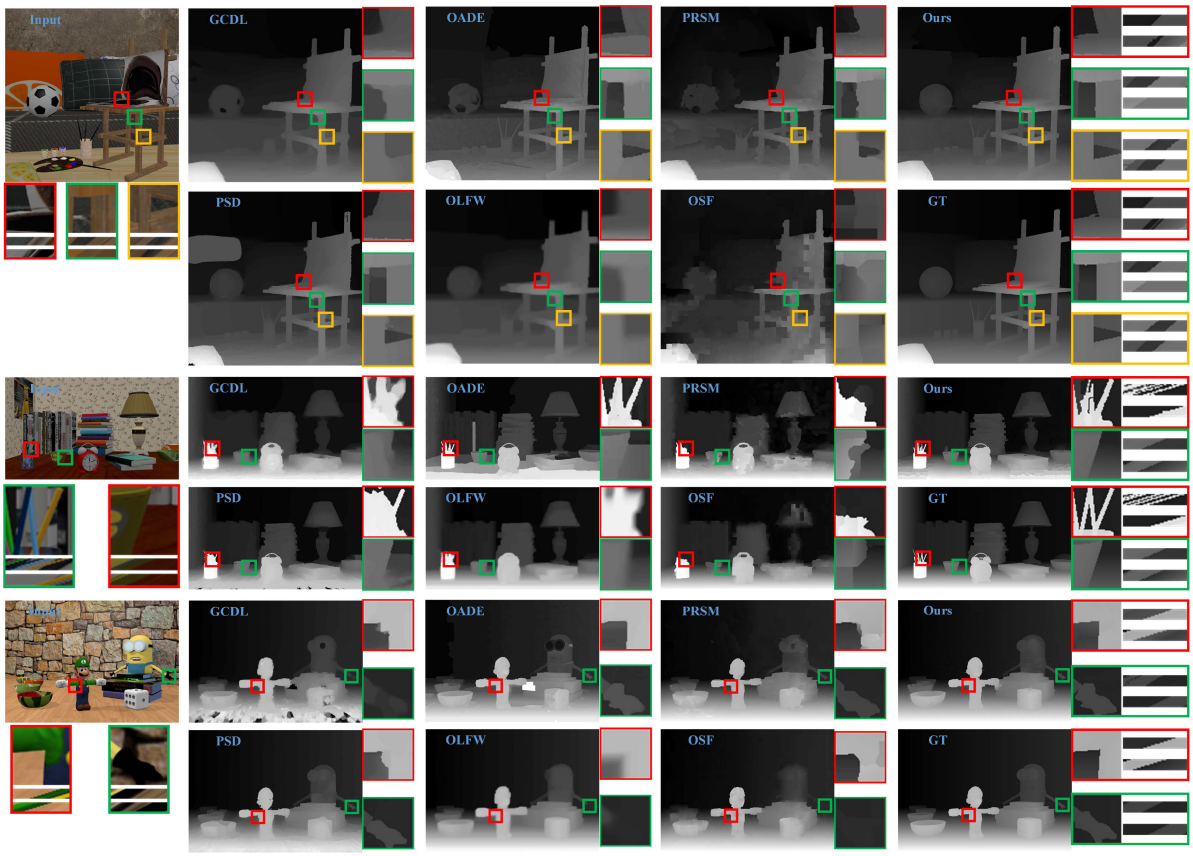


Fig. 6. The comparison of depth estimation on synthetic data 'Drawing', 'NewSecretaire' and 'Mario'. For each box in our results and GT (the rightmost column), the first and second rows show the 4-D depth map in the horizontal and vertical EPIs respectively.

Depth. Table II and Fig. 6 show quantitative and qualitative comparisons on depth estimation on the synthetic light fields respectively. We achieve the best performance on 9 of 10 light fields. Compared with light field based methods (GCDL, OADE, PSD, OLFW), our algorithm can obtain much smoother depth maps while preserving occlusion boundaries. In Fig. 6, previous methods always provide over-smooth results in the pens area(red box in the 'NewSecretaire'). Our method can also reconstruct thin structures well. It is all due to the introduction of the slanted plane model for each LFSP. Compared with traditional scene

TABLE IV
ACCURACY OF DEPTH AND OPTICAL FLOW ESTIMATION AS A FUNCTION OF THE NUMBER OF VIEWS

NumOfView	NewSecretaire		Mario		Drawing		Balls		NewBalls	
	Depth	Flow	Depth	Flow	Depth	Flow	Depth	Flow	Depth	Flow
3×3	0.121	1.504	0.095	1.415	0.068	1.527	0.045	0.577	0.055	0.775
5×5	0.103	1.500	0.089	1.333	0.055	1.500	0.040	0.521	0.046	0.714
7×7	0.092	1.463	0.079	1.152	0.046	1.444	0.037	0.499	0.041	0.701
9×9	0.080	1.393	0.073	1.012	0.039	1.324	0.036	0.481	0.041	0.693

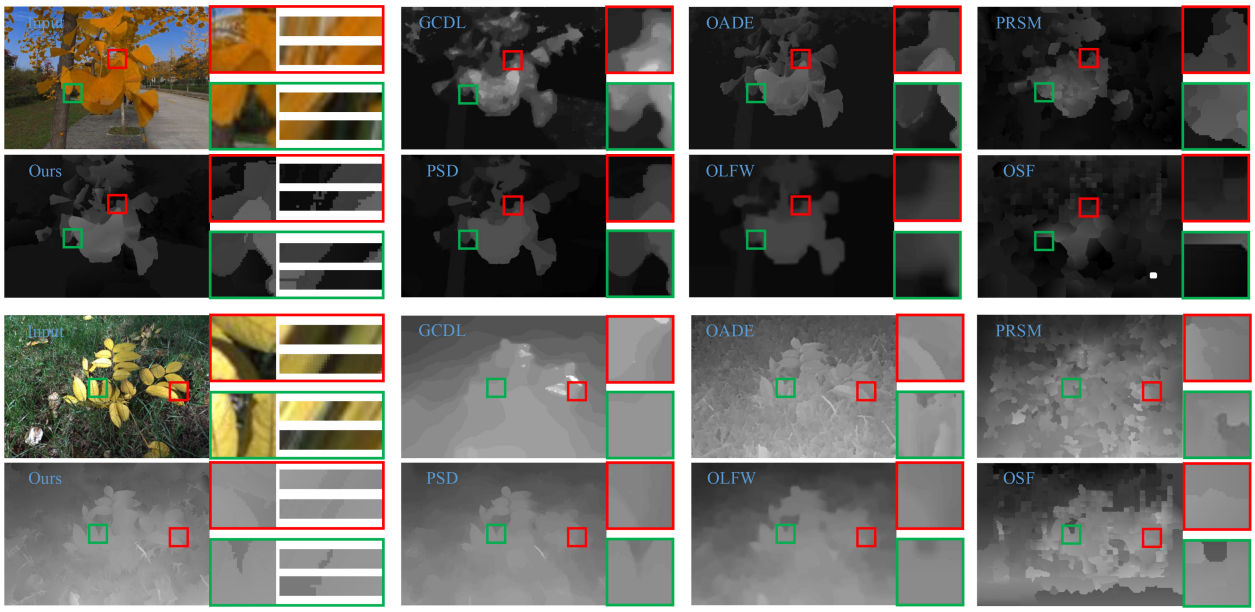


Fig. 7. The comparison of depth estimation on real light fields 'Ginkgo' and 'Leaves'.

TABLE V
DEPTH COMPARISONS ON THE PUBLIC LIGHT FIELD BENCHMARK [44]

	MSE		Bumpiness Planes	
	Value	Rank	Value	Rank
GCDL [6]	0.082	44	1.769	28
PSD [56]	0.091	47	1.593	22
OADE [7]	0.068	40	2.026	37
Ours	0.113	48	0.727	6

flow methods (PRSM, OSF), our algorithm exploits more views and thus benefit depth estimation. Additionally, we also provide the 4D comparison. For each scale-up region in the input light field, our results and ground truth, horizontal and vertical EPIs are also plotted. It can be found that the orientations of EPI lines in our results are very close to those of input light field and 4D ground truth.

As the results in Table II are obtained from our synthetic dataset, to be fair, we also provide results on the existing light field depth benchmark [44]. Table V shows quantitative comparisons. The term 'Bumpiness Planes' [44] measures the algorithm performance on smooth planar and curved surfaces. For each metric, we provide both the metric values and the rank in [44]. The proposed method performs not good on the mean square error (MSE), because the light fields [44] contain many thin structure objects which are difficult to be modelled using superpixel. However, our method shows great advantages than previous methods on the 'Bumpiness Planes', since the slanted plane model describes the depth distribution on plane areas accurately. More importantly, the slanted plane model is the basis for obtaining full view optical flow later.

Flow. In Table III and Fig. 8 we show the results and comparisons on optical flow computation on the synthetic light fields. Our algorithm performs best on all of 10 light fields. Compared with previous methods, our algorithm provides finer optical flow results. For example, there are three optical flow layers in the red box in the 'Drawing' sequence, *i.e.*, the front and rear legs of chair and the sofa. Only our algorithm can reconstruct these areas precisely whereas other methods tend to over-smooth them. In the green box of the "Mario" sequence, the dice is a continuous whole object where the motion vectors between the closest and farthest points have a smooth distribution. Previous methods either assign them the same motion vector (OSF and OLFW) or several discontinuous vectors (PRSM and LDOF). Only our method obtains continuous results. This may be the result of our method utilising multiview sub-apertures in the light field, which provides more precise normals for each of the LFSPs.

Similarly to our depth results, we also show, in Fig. 8, the 4D optical flows of scaled-up regions and plot the EPIs. Note that, as the optical flows in other views are propagated from the central view using the LFSPs, the accuracy may be affected by the LFSP segmentation. For example, in the green box of the Drawing in Fig. 8, our method achieves good results in the central view (the left patch), however, as the LFSP segmentation does not cling the

Fig. 8. The comparison of flow estimation on synthetic data 'Drawing', 'NewSecretaire', 'Mario', 'Balls' and 'NewBalls'. For each box in our results and GT (the rightmost column), the first and second rows show the 4-D flow map in the horizontal and vertical EPIs respectively.

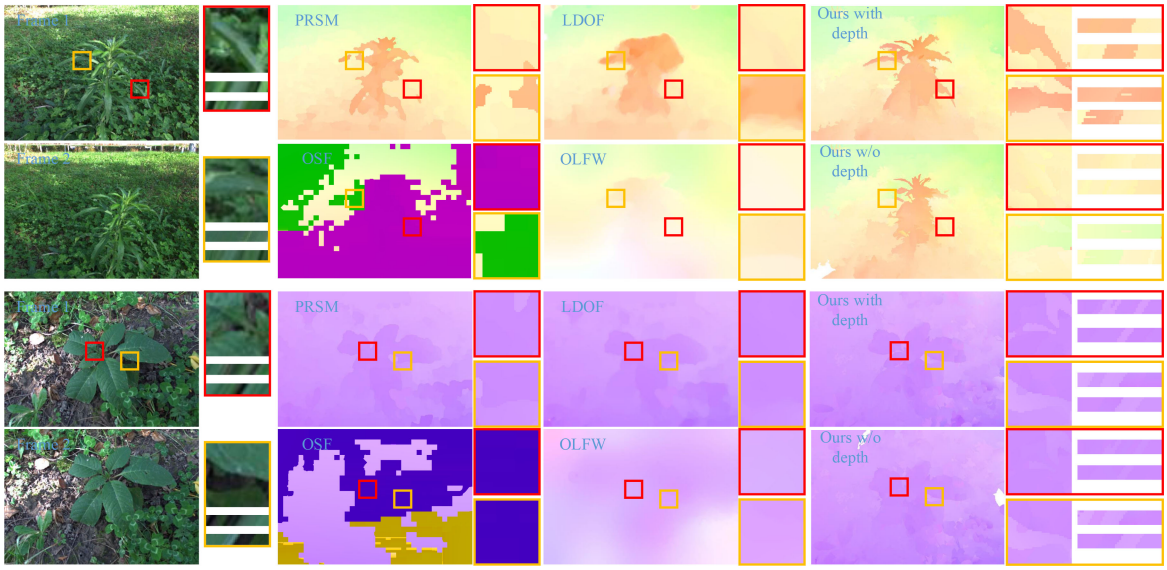


Fig. 9. The comparison of flow estimation on real light fields 'Plant1' and 'Plant2'.

direction of EPI line perfectly (the right bottom horizontal EPI), the optical flow in other views is over smoothed. As a result, the RMS and endpoint errors of the 4D depth and optical flow are slightly bigger than those delivered in 2D. Nonetheless, our algorithm can provide accurate optical flow maps in all views. Finally, we also evaluate the accuracy of depth and optical flow estimation as a function of the number of views in the light field (see Table IV). From the table, we can conclude that, as expected, the results improve as the number of views increases.

2) Real Data: **Depth.** Fig. 1 and 7 show depth estimation results on real light fields acquired from Lytro Illum. There are many occlusion boundary areas in the 'Ginkgo' and 'Leaves'. Compared with pervious work, our algorithm provides sharper occlusion boundaries (the boundaries of leaves in both red and green boxes in Fig. 7) and continuous depth in the leaves. Apart from this, the 4D depth on EPIs clings to the direction of epipolar lines.

Flow. Fig. 1 and 9 show flow maps on real light fields. Due to the high-accuracy depth maps from light field, our algorithm generates finer optical flow maps especially in object boundaries, such as the slender leaves (both red and yellow boxes in 'Plant1') and the gaps among leaves (the red box in 'Plant2'). The 4D flow maps are also propagated well thanks to the LFSP (the flow boundaries cling along the epipolar lines).

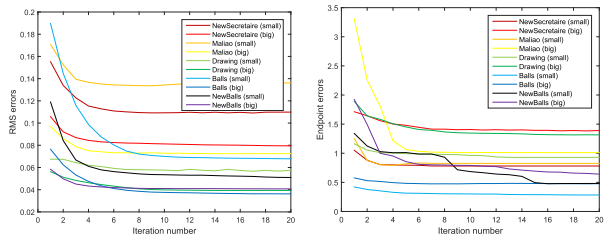
Influence of depth. As mentioned in Eqn. (10), the depth term can help to distinguish the foreground and background when they share similar textures. In Fig. 9, we also show flow maps without the depth term (the lower right corner). It can be found that the foreground leaves are over-smoothed by the background since the leaves and grasses are green. This problem can be partly solved when introducing the depth term (however the depth fails when the occlusion also appears, seeing Sec. IV-D.).

C. Complexity

We compare the running time with the only light field based optical flow method (OLFW) (see Table VI). Because only the

TABLE VI
THE COMPARISONS OF RUNNING TIME (SEC)

	OLFW	Ours			
		Total	LFSP	Opt. Depth	Opt. Flow
Time	7620	2006	123	1092	791



(a) Depth case (b) Optical flow case

Fig. 10. The RMSE and EPE with the iteration number.

2D central views are utilized, our un-optimized Matlab implementation takes about 6G RAM and 2700s in our synthetic big light field ($9 \times 9 \times 760 \times 760$), showing advantages than OLFW (107G RAM and 7620s) which utilizes the 4D patches. The running time of our main steps is also listed in Table VI. It can be found that most of the time is taken up by the depth and flow optimization. For each iteration, the depth and flow estimation stages take about 101s and 48s respectively. Therefore, the running time of our algorithm can be reduced by decreasing the number of iteration (see Fig. 10).

Supposing angular and spatial resolutions of light field are $M \times M$ and $N \times N$ respectively, the numbers of label and iteration are L and T, and the patch size is W. Then, the time complexity of flow computation of [18] is $\mathcal{O}(N^2M^2W^2L)$ while ours is only $\mathcal{O}(N^2LT)$. It is noticed that the time complexity of our algorithm is independent of angular resolution of light field. Any other traditional optical flow [19]–[25] or scene flow methods [27]–[32] can only obtain optical flow for the 2D central view. When applying these 2D methods in 4D full view optical

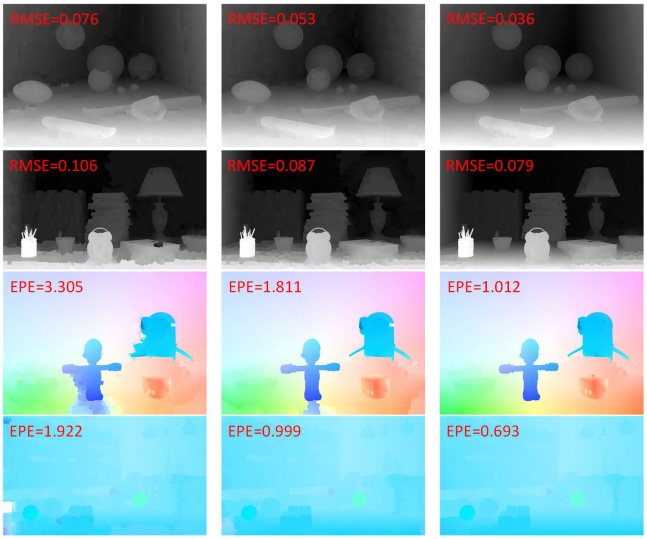


Fig. 11. The disparity and optical flow maps with the iteration number. From left to right: 1st iteration, 3rd iteration and last iteration.

(a) Occlusion and Similar textures (b) Bad LFSP segmentation

Fig. 12. The limitations of the proposed algorithm. For each region in the right red box, the first row shows the close-up of red rectangle in the left-most input image. The second and third rows show the ground truth and estimated optical flow maps respectively. The right-most column shows the horizontal and vertical EPIs respectively.

flow computation view by view, their running time will be magnified by angular sampling number. However, in our algorithm, the number of views only influences the LFSP segmentation and depth optimization and plays a minor role in the final time complexity.

D. Limitations

The algorithm can not handle the situations where the occluder and occludee share similar textures and the occlusion appears together. In Fig. 12(a), the football and pillow share similar white textures and the pillow is occluded in the second frame. Our algorithm can not smooth these areas well. Apart from this, our results are also influenced by bad LFSP segmentation. In Fig. 12(b), there is a crack in the cup area in both depth and flow maps due to bad segmentation of the LFSP. This is also the reason why the 4D results are worse than the 2D results in Table II and III.

V. Conclusion

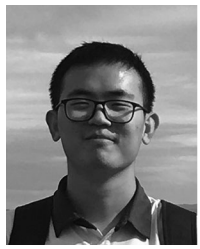
In the paper, we explore full view optical flow estimation in light fields. The whole 4D light field is organized using many small 4D LFSPs. By modeling the object of LFSP as a slanted-plane in 3D space, the optical flow for other views can be effectively propagated from the central view. Then, the scene flow for the central view of light field is estimated by sequentially optimizing the depth and flow. The proposed algorithm not only obtains 4D optical flow for all views in light fields, but also achieves better depth and flow maps than previous methods.

The full view optical flow between light fields is the fundamental for light field video authoring. It can be useful for many other problems such as light field video segmentation and compression. In the future, we will focus on improving the efficiency of the current flow estimation algorithm and exploring efficient methods for light field content manufacturing.

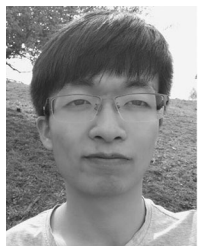
References

[1] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," *Comput. Models Visual Process.*, vol. 1, no. 2, pp. 3–20, 1991.
[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, 1996, pp. 31–42.
[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, 1996, pp. 43–54.
[4] B. Wilburn *et al.*"High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, 2005.
[5] Lytro, "The ultimate creative tool for cinema and broadcast," 2016. [Online]. Available: http://blog.lytro.com/?s=cinema
[6] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 606–619, Mar. 2014.
[7] T. C. Wang, A. A. Efros, and R. Ramamoorthi, "Depth estimation with occlusion modeling using light-field cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2170–2181, Nov. 2016.
[8] H. Zhu, Q. Wang, and J. Yu, "Occlusion-model guided anti-occlusion depth estimation in light field," *IEEE J. Sel. Topics Signal Process*, vol. 11, no. 7, pp. 965–978, Oct. 2017.
[9] T.-C. Wang, M. Chandraker, A. Efros, and R. Ramamoorthi, "SVBRDF-invariant shape and reflectance estimation from light-field cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 740–754, Mar. 2017.
[10] Z. Li, Z. Xu, R. Ramamoorthi, and M. Chandraker, "Robust energy minimization for BRDF-invariant shape from light fields," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2017, pp. 5571–5579.
[11] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?" *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 146.
[12] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, Y. Mukaigawa, and H. Nagahara, "4-D light field segmentation with spatial and angular consistencies," in *Proc. IEEE Int. Conf. Comput. Photography*, 2016, pp. 1–8.
[13] F.-L. Zhang, J. Wang, E. Shechtman, Z.-Y. Zhou, J.-X. Shi, and S.-M. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 5, pp. 1561–1573, May 2017.
[14] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging: Briefly revisiting 25 years of research," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 59–69, Sep. 2016.
[15] M. Hog, N. Sabater, and C. Guillemot, "Light field segmentation using a ray-based graph structure," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 35–50.
[16] C. U. Edussooriya, D. G. Dansereau, L. T. Bruton, and P. Agathoklis, "Five-dimensional depth-velocity filtering for enhancing moving objects in light field videos," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2151–2163, Apr. 2015.
[17] S. Heber and T. Pock, "Scene flow estimation from light fields via the preconditioned primal-dual algorithm," in *Proc. German Conf. Pattern Recognit.*, 2014, pp. 3–14.
[18] P. P. Srinivasan, M. W. Tao, R. Ng, and R. Ramamoorthi, "Oriented light-field windows for scene flow," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 3496–3504.
[19] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vision*, vol. 106, no. 2, pp. 115–137, 2014.
[20] J. Yang and H. Li, "Dense, accurate optical flow estimation with piecewise parametric model," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1019–1027.

[21] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 5704–5712.
[22] D. Gadot and L. Wolf, "Patchbatch: A batch augmented loss for optical flow," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4236–4245.
[23] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 471–488.
[24] Y. Yang and S. Soatto, "S2F: Slow-to-fast interpolator flow," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2087–2096.
[25] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2017, pp. 686–695.
[26] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 475–480, Mar. 2005.
[27] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proc. IEEE 11th Int. Conf. Comput. Vision*, 2007, pp. 1–7.
[28] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3-D motion understanding," *Int. J. Comput. Vision*, vol. 95, no. 1, pp. 29–51, 2011.
[29] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *Int. J. Comput. Vision*, vol. 101, no. 1, pp. 6–21, 2013.
[30] C. Vogel, K. Schindler, and S. Roth, "3-D scene flow estimation with a piecewise rigid scene model," *Int. J. Comput. Vision*, vol. 115, no. 1, pp. 1–28, 2015.
[31] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3061–3070.
[32] T. Taniai, S. N. Sinha, and Y. Sato, "Fast multi-frame stereo scene flow with motion segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3939–3948.
[33] H. Zhu, Q. Zhang, and Q. Wang, "4-D light field superpixel and segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1–8.
[34] M. Hog, N. Sabater, and C. Guillemot, "Super-rays for efficient light field processing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017.
[35] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second-order smoothness priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2115–2128, Dec. 2009.
[36] R. Ng, "Digital light field photography," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2006.
[37] L. Pan, Y. Dai, M. Liu, and F. Porikli, "Simultaneous stereo video deblurring and scene flow estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4382–4391.
[38] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[39] "The new Stanford light field archive," 2008. [Online]. Available: http://lightfield.stanford.edu/lfs.html
[40] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 1027–1034.
[41] Q. Zhang, C. Zhang, J. Ling, Q. Wang, and J. Yu, "A generic multi-projection-center model and calibration method for light field cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. [Online]. Available: https://www.computer.org/csdl/trans/tp/preprint/08430574-abs.html
[42] Y. Zhang, P. Yu, W. Yang, Y. Ma, and J. Yu, "Ray space features for plenoptic structure-from-motion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4631–4639.
[43] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4-D light fields." in *Proc. Conf. Vision, Model. Vis.*, 2013, pp. 225–226.
[44] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4-D light fields," in *Proc. Asian Conf. Comput. Vision*, 2016, pp. 19–34.
[45] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "4-D light field dataset," 2016. [Online]. Available: http://hci-lightfield.iwr.uni-heidelberg.de/
[46] G. Andreas, L. Philip, U. Raquel, and M. Moritz, "The KITTI vision benchmark suite," 2012. [Online]. Available: http://www.cvlibs.net/datasets/kitti/
[47] M. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 673–680.
[48] C. Chen, H. Lin, Z. Yu, S. Kang, and J. Yu, "Light field stereo matching using bilateral statistics of surface cameras," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1518–1525.
[49] Y. Anzai, *Pattern Recognition & Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2012.
[50] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.
[51] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.
[52] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 46.
[53] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 73.
[54] A. S. Raj, M. Lowney, and R. Shah, "Light-field database creation and depth estimation," Dept. Comput. Sci., Stanford Univ., Tech. Rep., 2016.
[55] Blender Foundation, Amsterdam, The Netherlands. [Online]. Available: blender.org
[56] H.-G. Jeon *et al.*, "Accurate depth map estimation from a lenslet light field camera," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1547–1555.



Hao Zhu received the B.E. degree, in 2014, from the School of Computer Science, Northwestern Polytechnical University, Xi'an, China, where he is currently working toward the Ph.D. degree with the School of Computer Science. His research interests including computational photography and light field computing theory and application.



Xiaoming Sun received the B.E. degree in computer science, in 2017, from the School of Computer Science, Northwestern Polytechnical University, Xi'an, China, where he is currently working toward the M.S. degree with the School of Computer Science. His research interests include computational photography, light field application, and deep learning.

Qi Zhang received the B.E. degree in electronic and information engineering from Xi'an University of Architecture and Technology, Xi'an, China, in 2013, and the M.S. degree in electrical engineering, in 2015, from Northwestern Polytechnical University, Xi'an, China, where he is currently working toward the Ph.D. degree with the School of Computer Science. His research interests include computational photography and light field computing theory and application.

Qing Wang received the graduate degree from the Department of Mathematics, Peking University, Beijing, China, in 1991, and the master's and Ph.D. degrees from the Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China, in 1997 and 2000, respectively. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University. He then joined the Northwestern Polytechnical University as a Lecturer. He worked as a Research Assistant and Research Scientist with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University from 1999 to 2002. He also worked as a visiting scholar with the School of Information Engineering, The University of Sydney, Australia, in 2003 and 2004. In 2009 and 2012, he visited Human Computer Interaction Institute, Carnegie Mellon University, for six months and Department of Computer Science, University of Delaware, for one month. He has authored-coauthored more than 100 papers in the international journals and conferences. His research interests include computer vision and computational photography, such as three-dimensional reconstruction, object detection, tracking and recognition in dynamic environment, and light field imaging and processing. In 2006, he was awarded as outstanding talent program of new century by the Ministry of Education, China. He is currently a member of IEEE and ACM. He is also a Senior Member of China Computer Federation.

Antonio Robles-Kelly received the B.Eng. (Honors) degree in electronics and telecommunications in 1998, and the Ph.D. degree in computer science from the University of York, York, U.K., in 2003. He remained in York until 2004 as a Research Associate under the MathFit-EPSRC framework and, in 2005, he moved to Australia and took a research scientist appointment with National ICT Australia (NICTA). In 2006, he became the project leader of the Imaging Spectroscopy team at NICTA and, from 2007 to 2009, he was a Postdoctoral Research Fellow of the Australian Research Council. In 2016, he joined commonwealth scientific and industrial research organisation (CSIRO), Eveleigh, NSW, Australia, where he is a Principal Researcher with Data61 and, in 2018, was a Machine Learning and Artificial Intelligence Professor with Deaking University, Geelong, VIC, Australia. His research has been applied to areas, such as biosecurity, forensics, food quality assurance, and biometrics, and is now being deployed by CSIRO under the trademark of Scyllarus (www.scyllarus.com). He has served as the President of the Australian Pattern Recognition Society and is an Associate Editor for the *Pattern Recognition* Journal and the *IET Computer Vision* Journal. He is a Senior Member of the IEEE, the President of the TC2 (Technical Committee on structural and syntactical pattern recognition) of the International Association for Pattern Recognition and an Adjunct Associate Professor with the ANU. He has also been a technical committee member, area and General Chair of several mainstream computer vision and pattern recognition conferences.

Hongdong Li received the Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Reader with the Computer Vision Group, Australian National University (ANU), Canberra, ACT, Australia. He is also a Chief Investigator with the Australia ARC Centre of Excellence for Robotic Vision (ACRV), Brisbane City QLD, Australia. He was a Postdoctoral Fellow with the ANU in 2004. His research interests include three-dimensional vision reconstruction, structure from motion, multi-view geometry, as well as applications of optimization methods in computer vision. Prior to 2010, he was with NICTA Canberra Labs working on the Australia Bionic Eyes project. He is an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and served as Area Chair in recent year International Conference on Computer Vision, European Conference on Computer Vision, and Conference on Computer Vision and Pattern Recognition. He was a Program Chair for (ACRA) 2015—Australia Conference on Robotics and Automation, and a Program Co-Chair for ACCV 2018—Asian Conference on Computer Vision. He won a number of Best Paper Awards in computer vision and pattern recognition, and was the receipt for the CVPR Best Paper Award in 2012 and the Marr Prize Honorable Mention in 2017.

Shaodi You received the bachelor's degree from Tsinghua University, Beijing, China, in 2009, and the M.E. and Ph.D. degrees from The University of Tokyo, Tokyo, Japan, in 2012 and 2015, respectively. He is currently a Senior Research Scientist with Data61, (previously known as NICTA) Canberra Research Lab and a senior Adjunct Lecturer with the College of Engineering and Computer Science, Australian National University, Canberra, ACT, Australia. His research interests include physics-based vision, perception based vision and learning, and three-dimensional geometry.