Abstract—Hand-held light field (LF) cameras have unique advantages in computer vision such as 3-D scene reconstruction and depth estimation. However, the related applications are limited by the ultra-small baseline, leading to the extremely low depth resolution in reconstruction. To solve this problem, we propose to rectify LF to obtain a large baseline. Specifically, the proposed method aligns two LFs captured by two hand-held LF cameras with a random relative pose, and extracts the corresponding row-aligned subaperture images (SAIs) to obtain an LF with a large baseline. For an accurate rectification, a method for pose estimation is also proposed, where the relative rotation and translation between the two LF cameras are estimated. The proposed pose estimation minimizes the degree of freedom (DoF) for pose estimation is also proposed, where the relative rotation and translation between the two LF cameras are estimated. The proposed pose estimation minimizes the degree of freedom (DoF) and explicitly outperforms the state-of-the-art algorithms by providing more accurate results to support rectification. The row-aligned SAIs and significantly improved depth resolution in 3-D reconstruction demonstrate the effectiveness of the proposed LF rectification.

Index Terms—Baseline, depth estimation, light field (LF) cameras, LF rectification, relative pose estimation.

NOMENCLATURE

Notation

| Notation | Definition |
|----------|------------|
| $[x_c, y_c, \lambda]^T$ | Coordinates of an LF-point. |
| $(v_1, v_2, v_3)$ | Scaling factors for the image and disparity coordinate in normalization. |
| $[x_1, x_2, x_3]^T$ | Translation vector in normalization. |
| $(f_x, f_y, c_x, c_y, K_1, K_2)$ | Intrinsic parameters of an MLA-based LF camera. |
| $[s, t, u, v]^T$ | Coordinates of a ray parameterized by the TPP parameterization. |
| $P$ | An LF-point. |
| $H$ | Intrinsic parameter matrix of an micro-lens array (MLA)-based LF camera. |
| $W$ | Projective transformation matrix between two LF-points corresponding to the same 3-D point. |
| $W'$ | Projective transformation matrix determined up to a non-zero factor with $W$. |

$\vec{W}_{16}$ | $16 \times 1$ vector of all elements of $W'$. |
$\vec{W}_{15}$ | $13 \times 1$ vector of non-redundant representation of $\vec{W}_{16}$. |
$R_{3 \times 3}$ | Rotation matrix of An LF-camera. |
$T_{3 \times 1}$ | Translation vector of An LF-camera. |
$vec(R_{3 \times 3})$ | $9 \times 1$ vector of all elements of $R_{3 \times 3}$. |
$vec(T_{3 \times 1})$ | $9 \times 1$ The same as $T_{3 \times 1}$. |
$A_T$ | Coefficient matrix of $vec(R_{3 \times 3})$. |
$A_T$ | Coefficient matrix of $vec(T_{3 \times 1})$. |
$T_{\text{est}}$ | Estimated rotation matrix. |
$T_{\text{est}}$ | Estimated translation vector. |
$R_{\text{rect}}$ | Rectification matrix between two LF cameras. |

I. INTRODUCTION

The commercialization of light field (LF) cameras, such as Lytro [1] and Raytrix [2], have accelerated the development of LF technologies. Unlike the classic pinhole camera, an LF camera can capture both spatial and angular information of rays in the environment by a single 2-D snapshot, which is the reason why the LF camera has gained a significant interest in computer vision. Typical applications of the LF camera include simultaneous localization and mapping (SLAM) [3], [4], geometry measurement [5], digital refocusing [6], super resolution [7], [8], and semantic segmentation [9] etc.

One of the most important tasks of the LF camera is depth estimation [10], [11], [12], [13], [14], [15]. Compared with traditional stereo image pairs, LFs extend disparity to a continuous space [11]. This advantage is apparent when considering epipolar plane images (EPIs) [16]. Due to a dense sampling in the angular direction (e.g., 11 × 11 subviews in Lytro Illum), corresponding pixels of a scene point in subaperture images (SAIs) can be projected onto a slope line in EPIs, and line parameters can be encoded into dense stereo matching to obtain a more robust depth estimation [11]. However, a single LF has a weakness that cannot be ignored: the extremely small baseline between SAIs. Under normal settings, the SAI baseline can be as small as 0.5 mm. With this structural imperfection, even if LF depth estimation methods can achieve a high disparity resolution, a relative low depth resolution is still inevitable, which generally leads to a layered 3-D reconstruction, as shown in Fig. 1. To obtain an LF with a larger baseline, on one hand, using the camera array [17], [18] is a candidate choice. However, it is complicated to align all cameras in the camera array and this solution is also high in cost to reach the identical angular resolution as that of an LF camera. On the other hand, manually aligning two hand-held
LF cameras is another straightforward solution, whereas it is also hard to make the image planes reside in the same plane and the rows of corresponding SAIs completely aligned. Hence, it is critical to find a solution which mathematically aligns two hand-held LF cameras with a random relative pose to obtain an LF with a large baseline.

The methods designed for traditional rectification of stereo images can be used to rectify the central SAIs. However, LF cameras have multiple SAIs, and finding accurate translations for other subaperture pairs is difficult and computationally expensive. Even if pairwise SAIs are rectified, there is no guarantee that collinear subapertures in the original LF configuration will remain collinear after rectification, impacting reliability of depth estimation. Therefore, a specialized and efficient rectification approach is needed to generate row-aligned SAIs for LF cameras. In this article, we propose LF rectification to align two LFs captured by two hand-held LF cameras with a random relative pose. A pairwise rays warping function is established, and the rays of LFs captured by two hand-held LF cameras are warped into a common two-plane parameterized (TPP), so that an LF with a large baseline can be obtained. A subsequent quadrilinear interpolation among warped rays is implemented to acquire the row-aligned SAI sets. A commonly used depth estimation algorithm designed for a single LF is performed on the rectified LF to output the depth in a high depth resolution.

For an accurate rectification, a method for pose estimation is also proposed, where the estimated relative rotation and translation are used to compose the rays warping function. LF images provide rich angular information, for example, images from $11 \times 11$ subviews in Lytro Illum can be extracted, i.e., the SAIs. Based on these SAIs, depths of scenes can be recovered which are usually embedded in rays or 3-D points outside the LF cameras. With these ray or 3-D point features, many relative pose estimation methods are proposed, such as [19], [20], [21]. However, Due to the impact of the narrow baseline of LF camera, rays or 3-D scene points transformed from the feature points in LF images usually cannot achieve high accuracy, which significantly penalize the performance of relative pose estimation. Therefore, a reasonable algorithm should utilize the depth information provided by the high angular resolution and simultaneously avoid the noises brought by the narrow baseline. Utilizing LF features extracted in LF images to estimate the relative pose is gradually becoming the popular approach [21], [22]. In prior state-of-the-art method [22], correspondence between matched LF features, i.e., LF-point-LF-point correspondence is established. Based on this model, the relative pose is decoupled and represented by the components of LF-points and the intrinsic parameters of an LF camera. However, the LF-points are still possibly influenced by the intrinsic parameters, just such as rays or 3-D-points influenced by the ultra-small baseline of the LF camera. In this article, we explicitly solve the LF-point-LF-point correspondence in a linear way as a whole while maintaining the minimum degree of freedom (DoF) of this model. The relative pose is then recovered from the model parameters. Only LF-points are input in solving the correspondence, thus the influence of the intrinsic parameters to LF-points is avoided and the complete information of LF-points are retained when compared to [22]. To validate the proposed algorithm, we compare it with classical as well as state-of-the-art pose estimation algorithms for LF camera in LF-SfM [19], LF-DLT [21], and LFP [22]. Experiments on simulated and real scene data are performed to illustrate the superiority of the proposed algorithm.

The main contributions of this article include as follows.

1) We propose a linear approach to solve the LF-point-LF-point correspondence model as a whole while maintaining the minimum DoF of this model. The original information of LF images is well exploited to avoid the influence of the small baseline, from which a more accurate relative pose can be extracted. Experimental results demonstrate that, compared with other state-of-the-art relative pose estimation algorithms, the proposed algorithm is accurate and robust, which prepares for the proposed LF rectification. Besides, the degeneracy of the LF-point-LF-point correspondence model is given to make the model complete.

2) To the best of our knowledge, our work is the first to rectify two 4-D LFs based on the estimated relative pose of LF cameras. The rectified LF not only has the enlarged baseline but also holds the characteristics of LF data such as regular sampling pattern and dense angular sampling, which make the depth estimation on a pair of LF achievable and the depth resolution is significantly improved. Experiments show the effectiveness of the proposed LF rectification.

3) The proposed LF rectification supports the modular design. The relative pose estimated by other existing or future methods can be used to achieve the LF rectification, and the rectified LF supports the existing or future depth estimation methods designed for the LF data to provide a flexible solution of the 3-D reconstruction.

The remainder of the article is organized as follows. Section II summarizes related prior works. Section III first discusses the disadvantage of estimating relative pose by reconstructed rays and points and then introduces the adopted feature representation (i.e., LF-point) and related LF-point-LF-point correspondence model, which are considered as backgrounds of the proposed algorithm. Based on these backgrounds, we derive the proposed relative pose estimation
method in Section IV. Section V describes the proposed LF rectification method. Section VI presents the experimental results on simulated and real scene data, demonstrating the accuracy and robustness of the proposed relative pose estimation algorithm compared with state-of-the-art methods, as well as the effectiveness of the proposed LF rectification. Finally, Section VII concludes the article.

II. RELATED WORK

A. LF Feature

Different LF features are useful in various LF applications. For LF camera calibration, Bok et al. [23] proposed to detect “line features” in raw images of LF, which is the 2-D projections of the 3-D line segment connecting two adjacent corners in the checkerboard. O’brien et al. [24] defined the “plenoptic disk feature” by combining 2-D pixel coordinates of corners in the central SAI and the normalized disparity. O’brien et al. [24] utilized speeded up robust features (SURFs) [30] of scene points extracted from RGB images and designed slope filters to decrease the mismatches between adjacent images, paving the way for pose estimation of LF cameras. Dansereau et al. [25] proposed a scale invariant LF feature detector and descriptor, referred to as “LiFe,” which detects LF features from the focal stack composed of LF images. Jin et al. [26] combined 2-D pixel coordinates in the central SAI and the disparity in adjacent SAIs to represent the feature of a 3-D scene point, referred to as the “LF-point” (detailed in Section III). Projected image points in all SAIs of the 3-D scene point can be calculated from the corresponding LF-point, which indicates that the LF-point contains complete LF structure of 3-D scene points.

B. LF Relative Pose Estimation

Since an LF camera can be considered a pinhole camera array, the pose estimation for traditional pinhole cameras is useful for estimating the relative pose between LF cameras. The most direct way is to use the methods for epipolar geometry estimation such as [27], [28], [29] with two central SAIs. However, applying only one SAI for each LF camera cannot make good use of the rich angular information provided by LF cameras. Since LF cameras can also give depth information of scenes in a single exposure, iterative closed point and perspective-n-point based methods such as [29], [30], [31] can be employed to estimate relative poses between LF cameras. Nevertheless, the depths of 3-D scene points are difficult to recover with high accuracy due to the ultra-small baseline in this case, and the spatial resolution of SAIs is extremely low, which brings great challenges to the relative pose estimation for traditional cameras.

In this situation, designing pose estimation algorithms tailored to LF cameras can achieve better performance by considering the characteristics of LF cameras. Pless [32] defines a generalized camera model as a collection of rays, and establishes the correspondences between rays intersecting the same 3-D point, denoted as the generalized epipolar constraint (GEC). Taking all kinds of degenerated camera configurations into consideration, Li et al. [33] used a linear algorithm to solve the relative pose from the GEC efficiently without ambiguities, which was also applicable to LF cameras. Johannsen et al. [19] first introduced the GEC into the LF camera and considered the geometric constraints between projections within a single LF, which effectively utilized the characteristics of the LF camera. They further introduced the bundle adjustment to refine the estimation in their more recent work [20], and the refined results were adopted to make the comparison in this article. Zhang et al. [34] study how ray manifolds associated with geometric features including points, lines, and planes transform when the relative pose changes and exploit these transformations to recover the pose. The point-ray manifold is identical to that in [19]. Nousias et al. [35] proposed a complete pipeline of LF structure from motion (SfM) where the approach of pose estimation is the same as [33]. Zhang et al. [36] established the ray-space homography between two LF cameras and used rays captured by the two LF cameras to solve the relative pose. With an assumption that the LF camera is uncalibrated, the translation can be determined up to a scale factor.

However, the rays reconstructed from the LF images by the small baseline are very dense and easy to be corrupted by noises in the above methods [21]. To avoid the disadvantages of rays, Nousias et al. [21] proposed an LF projection matrix to represent the correspondence between the LF features and 3-D points, and used the direct linear transformation (DLT) to estimate the absolute pose of LF cameras. The LF features they used are directly extracted from LF images. However, the accuracy of recovered 3-D points can still be affected by the narrow baseline. In the state-of-the-art work [22], the LF-point-LF-point correspondence model was established, which described the correspondence between LF features (LF-points) extracted from a pair of LF raw images. The use of 3-D points is avoided. In the solving process of this model, the rotation and translation were decoupled from the model and solved separately. The relative pose is represented by the components of LF-points and the intrinsic parameters of an LF camera.

In this article, we also adopted LF-points, but explicitly solved the LF-point-LF-point correspondence in a linear way, while maintaining the minimum DoF of this model. The relative pose is then recovered from the model parameters. By doing so, the influence of the intrinsics on LF-points is avoided and the complete information on F-points are retained.

C. Traditional Rectification Method

Based on the extracted relative pose, the alignment can be implemented. The rectification methods [37], [38], [39] for the traditional camera can generate row-aligned images from the pair of stereo images. Given the relative pose that relates the stereo images, the traditional rectification method constructs two rotation matrices for the left and right cameras, respectively. The two matrices rotate two cameras around their centers of projection so that the two image planes reside in a common plane, epipolar lines become horizontal and the epipoles are at infinity. Then the row-aligned images can be interpolated from the original images. This traditional rectification method can be used for the central SAI pair because the
rotation and translation of two center subapertures are the same as those of two LF cameras. However, the translations of the other subapertures pairs that can be considered as the pinholes of traditional cameras are unknown and hard to accurately estimate. Massive pairwise combinations for subapertures also bring a huge burden to the traditional rectification method. Even if the pairwise SAIs are rectified, it cannot guarantee that the collinear subapertures in the original LFs still exist. If not, the depth estimation for an LF camera will be unreliable. Therefore, there is a need for a specialized and efficient LF camera rectification method that can generate row-aligned SAIs for all subapertures, ensuring accurate depth estimation and maintaining the characteristics of the LF data. In this article, we propose to construct a third TPP and warp the rays captured by the LF cameras into this TPP using the derived pairwise rays warping function. To maximize the number of captured by the LF cameras into this TPP using the derived article, we propose to construct a third TPP and warp the rays and maintaining the characteristics of the LF data. In this design, the challenges posed by the traditional rectification methods can be overcome.

III. Background on Relative Pose Estimation for LFs

For an effective LF rectification, the prerequisite is to obtain the accurate relative pose of the two LF cameras. If the points or rays outside the LF camera are used to solve the poses, the feature points in the LF images must be used to reconstruct these points or rays through the intrinsic parameters of the LF camera. However, the small baseline, which belongs to intrinsic parameters [13], significantly penalizes the reconstruction accuracy of rays and points. For rays, the sampling interval is usually only one-hundredth of a pixel, which makes it difficult to distinguish them, resulting in significant errors in reconstruction accuracy. More analysis can be found in [21]. For 3-D points, the small baseline will amplify the detection noise of features (e.g., 2-D SIFT feature). The depth $Z$ of points can be calculated by the depth formula as

$$Z = \frac{B f}{D}$$

(1)

where $f$ is the focal length of the LF camera, and $D$ is the disparity of features between the adjacent SAIs. The differential change of $Z$ is calculated as

$$dZ = -\frac{Z^2}{Bf} dD.$$  

(2)

For a baseline $B = 0.5$ mm, a small disparity error $dD$ can make for a large depth error $dZ$. Here the disparity is expressed by the coordinate difference of detected feature points in two adjacent SAIs. Therefore, the reconstructed rays and points provide limited information.

LF-point $(x_c, y_c, \lambda)$ is an LF feature extracted from LF images and is first defined in [26] which has a one-to-one correlation with a 3-D scene point. The mathematical expression is composed of the projection coordinates $(x_c, y_c)$ of the 3-D point in the central SAI and its disparity $\lambda$ between any two adjacent SAIs. According to [23], the SAIs are evenly arranged in the image plane, thus the disparities of the 3-D point obtained from different adjacent SAIs are assumed to be equal. It indicates that this triplet can completely contain the inherent information about the LF structure of 3-D scene points.

LF-point-LF-point correspondence is a model defined in the work [22], describing the mapping between LF-points and established as (3), shown at the bottom of the next page, where $[x_c, y_c, \lambda, 1]^T$ and $[x'_c, y'_c, \lambda', 1]^T$ are the homogeneous coordinates of LF-points (in a pair of LFs) corresponding to the same 3-D point. $Z$ and $Z'$ are the depth of the 3-D point in the two LF camera coordinate systems, respectively. $H$ and $H'$ are the intrinsic parameter matrices of the two LF cameras which are already known after calibration. $R_{3 \times 3}$ and $T_{3 \times 1}$ are the unknown extrinsic parameters that need to be estimated. This correspondence indicates that the relative pose can be solved from the LF-points, thus the difficulties and loss of accuracy in reconstructing rays or 3-D points can be avoided. Details can be found in [22]. This article explicitly solves the LF-point-LF-point correspondence as a whole and then recovers the relative pose of LF cameras from the model parameters.

IV. Relative Pose Estimation

In this section, we derive the proposed relative pose estimation method. Fig. 2 shows the pipeline of the proposed LF rectification. The LFs of the checkerboard and real scenes are captured by two internal-calibrated LF cameras with a fixed relative pose. The green block shows the steps of relative pose estimation. As in the work [22], the calibration is performed using the method introduced in [26]. To avoid degeneracy in the LF-point-LF-point correspondence model, the checkerboard should be placed in different poses. In the relative pose estimation, the LF-points of different corners in two cameras are extracted, respectively. The detailed extraction method is shown in Section VI-B. Then these LF-points pairs are normalized based on the invariance of image coordinate transformation. The parameters in the LF-point-LF-point correspondence model are estimated by the proposed linear approach using these LF-point pairs. Finally, the initial relative pose is extracted from these parameters and refined by the Levenberg–Marquardt-on-Manifold algorithm. The commonly used notation of symbols in this article is shown in Nomenclature.

A. Initial Solution for $R$ and $T$

In the data-preprocessing step, the LF-points are normalized based on the invariance to image coordinate transformation [40]. The normalization matrix $N$ is introduced as

$$N = \begin{bmatrix} v_1 & 0 & 0 & x_1 \\ 0 & v_2 & 0 & x_2 \\ 0 & 0 & v_3 & x_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4)

where $v_1$, $v_2$, and $v_3$ are scaling factors for the image and disparity coordinate. $[x_1, x_2, x_3]^T$ are the translation vector.
Normalization translates the set of LF-points so as to bring the centroid of them to the origin and scale them to make the two principal moments both equal to unity. An independent normalization is also applied to another set of LF-points, and \( N' \) has a similar form.

Equation (3) shows that the matrix \( M \) can map an LF-point to its corresponding LF-point. For different scene points, their LF-points can be mapped to the corresponding LF-points by the same projective transformation \( W \) under homogeneous coordinates, since \( W \) only relates to the intrinsic and extrinsic parameters of these two cameras.

Based on the above analysis, the projective transformation can be expressed in the matrix form as

\[
P' \simeq WP \tag{5}
\]

where \( P \) and \( P' \) are a pair of LF-points, and \( \simeq \) denotes equality up to a scale factor.

Based on the DLT algorithm described in [40], \( W \) in (5) can be determined up to a nonzero scale factor, referred to as \( W' = \mu W \). Further, (5) is expanded by taking the cross product of the left- and right-hand sides as

\[
A \text{ vec}(W'_{16}) = 0 \tag{6}
\]

where the matrix elements of \( A \) are quadratic in the known coordinates of the LF-points, and \( \text{vec}(W'_{16}) \) is a 16 \( \times \) 1 vector consisting of the entries of the matrix \( W' \). Equation (7) shows the details of \( A \) as

\[
A = \begin{bmatrix}
P & 0_{1 \times 4} & 0_{1 \times 4} & -x_i' P \\
0_{1 \times 4} & P & 0_{1 \times 4} & -y_i' P \\
0_{1 \times 4} & 0 & P & -\lambda' P
\end{bmatrix}. \tag{7}
\]

\( W' \) can be solved from (6) by performing singular value decomposition (SVD), and then the rotation and translation can be extracted from \( W' \). On the other hand, \( W' \) represents a 3-D projective transformation that can only be defined up to scale. The total number of DoF of it is 15. The Rotation (considered as nine independent variables) and translation take 12 DoF, thus there are still redundant variables in (6). Here, we give three linear constraints to minimize the DoF as

\[
\begin{align*}
w_9' &= -K_1'w_{13}' \\
w_{10}' &= -K_1'w_{14}' \tag{8} \\
w_{12}' &= -K_1'w_{16}' + K_1w_0'
\end{align*}
\]

where \( w_{13}' = -w_{13}' + K_1w_{15}' \). The derivation can be found in Section I of the supplemental document. Then \( \text{vec}(W'_{16}) \) can
be represented as

\[
\begin{pmatrix}
    \mathbf{w}_1' \\
    \mathbf{w}_2' \\
    \vdots \\
    \mathbf{w}_{16}'
\end{pmatrix} =
\begin{bmatrix}
    I_{8\times 8} & 0_{8\times 5} \\
    0_{1\times 8} & -K_1 & 0 & 0 & 0 \\
    0_{1\times 8} & 0 & -K'_1 & 0 & 0 \\
    0_{1\times 8} & 0 & 0 & -K'_1 & 0 & -1 \\
    0_{4\times 8} & I_{4\times 4} & 0_{4\times 1}
\end{bmatrix}
\begin{pmatrix}
    \mathbf{w}_1' \\
    \mathbf{w}_2' \\
    \vdots \\
    \mathbf{w}_{16}'
\end{pmatrix}.
\]

(9)

The coefficient matrix in (9) is represented by \(Q\), and variables on the right side of the equation are represented by \(\text{vec}(W_{13}')\). \(I\) is the identity matrix, and the subscript indicates its size. Substituting (6) to (9), we obtain

\[AQ\text{vec}(W_{13}') = 0.\]

(10)

By performing SVD on \(AQ\), \(\text{vec}(W_{13}')\) is solved subject to \(\|\text{vec}(W_{13}')\| = 1\). Then we calculate \(\text{vec}(W_{16}')\), reform it to matrix \(W'\) and remove the intrinsic parameter matrices to calculate the rotation and translation. This process is expressed through rearranging \(W'\) as

\[\mu \begin{bmatrix}
    R_{3\times 3} \\
    T_{3\times 1}
\end{bmatrix} = H^{-1}W'H.\]

(11)

\(R_{3\times 3}\) and \(T_{3\times 1}\) can be obtained through scaling the matrix on the left-hand side. Considering the orthogonality of the rotation matrix, \(R\) is projected into the space of rotation matrices [SO(3)] using the method in [41] to get its initial solution.

In order to obtain the linear solution of translation, (3) is restated to a linear form as

\[\begin{bmatrix} A_R & A_T \end{bmatrix} \begin{bmatrix} \text{vec}(R_{3\times 3}) \\
    \text{vec}(T_{3\times 1}) \end{bmatrix} = 0.\]

(12)

where \(A_R\) and \(A_T\) are the known coefficient matrix of \(\text{vec}(R_{3\times 3})\) and \(\text{vec}(T_{3\times 1})\), respectively. A detailed derivation of (12) can be found in Section II of the supplemental document. \(\text{vec}(R_{3\times 3})\) is the \(9\times 1\) vector formed by stacking the columns of \(R\), and \(\text{vec}(T_{3\times 1})\) is exactly \(T\). After solving the rotation matrix, we can calculate the translation vector using the method discussed in [40]. Here

\[\text{vec}(T_{3\times 1}) = -A_T^+A_R\text{vec}(R_{3\times 3})\]

(13)

where \(A_T^+\) represents the pseudo-inverse of \(A_T\).

B. Nonlinear Optimization

The initial solution computed by the linear method is refined via nonlinear optimization. The variables to be optimized are pose parameters \(R\) and \(T\). The reprojection cost function measuring the geometric distance between the LF-point is defined as follows:

\[\sum_{i=1}^{N} d\left(\mathbf{x}_{iL}^{\text{est}}, \mathbf{y}_{iL}^{\text{est}}; \mathbf{x}_{iL}^{\text{est}}, \mathbf{y}_{iL}^{\text{est}}\right)^2\]

(14)

where \(N\) is the number of LF-points, and \(d(\cdot, \cdot)\) is the Euclidean distance. \([x_{iL}^{\text{est}}, y_{iL}^{\text{est}}, \lambda_{iL}^{\text{est}}]^T\) is the LF-point of the \(i\)th 3-D point in the second LF. \([x_{iL}^{\text{est}}, y_{iL}^{\text{est}}, \lambda_{iL}^{\text{est}}]^T\) is the corresponding estimated LF-point calculated by (3). Equation (14) is minimized to determine the final pose parameters of the camera setup. The minimization is performed by the Levenberg-Marquardt-on-Manifold Algorithm used in [22], [42], [43], and [44].

C. Degeneracy

It should be noted that when all 3-D points are coplanar, there may be degeneracy in the LF-point-LF-point correspondence model. When the scene being captured lies in a single plane, we can see the translation part becomes zeros, thus the matrix loses the DoF of the translation. As a result, the coefficient matrix \(AQ\) becomes rank deficient, making the estimation of the homography matrix unstable. In such a case, the proposed estimation method may not be effective. This situation is given as

\[\begin{bmatrix}
    x' \\
    y' \\
    \lambda'
\end{bmatrix} = Z'H \begin{bmatrix}
    R_{3\times 3} - T_{3\times 1}H' \\
    0_{1\times 3}
\end{bmatrix} \begin{bmatrix}
    0_{1\times 1} \\
    1
\end{bmatrix}^{-1} \begin{bmatrix}
    x_c \\
    y_c \\
    \lambda_c
\end{bmatrix}\]

where \(n\) denotes the normal vector of this plane, and \(d\) is the distance from the plane to the origin. Equation (15) shows that \(U\) is another solution of the homography matrix. The derivation of (15) can be found in Section III of the supplemental document. The traditional method of solving homography of 2-D features in images [45] (i.e., using the image coordinates \((x_c, y_c)\) of LF-points) and extracting the relative pose from the homograph can be an alternative solution.

V. LF Rectification

In this section, we describe the proposed LF rectification method. As shown in the blue block in Fig. 2, the LF pairs of real scenes are processed. In the preparation phase, for a pair of LFs, SAIs sets of both LFs are acquired, respectively, through the extraction algorithm in [23]. An LF camera can be regarded as a set of pinhole cameras arranged in a grid and the SAIs of an LF camera are equivalent to those images captured by these pinhole cameras. The coordinates and luminance of rays are converted from SAIs through calibrated intrinsic parameters. After obtaining the solved relative pose, rectification starts by constructing the rectification matrices and vectors of the two LFs. Then, based on the warping function represented by the elements of the rectification matrix and vector, the rays of two LFs are warped into the constructed TPP. Finally, considering that the LF under TPP is a 4-D signal, the row-aligned SAIs are obtained through quadrilinear interpolation. These extended SAIs are applied in the subsequent depth estimation and metric reconstruction as shown in the orange block.

After obtaining the relative pose, one LF can be warped into the TPP parameterization of another LF. However, the direct warping result usually cannot enlarge the baseline effectively. According to the classical work of LF camera calibration [23], subapertures are located on the main plane of the LF camera. The same row of subapertures should have the same vertical
coordinates, and the same column of subapertures should have the same horizontal coordinates as shown in Fig. 3. Unfortunately, it is hard for users to manually place the two cameras in a proper position so that their main planes can be perfectly coplanar and subapertures distribute appropriately. The proposed LF rectification allows users to freely rotate and translate the LF cameras and align the two LFs into a common TPP parameterization. Fig. 3 shows the goal of LF rectification: constructing the third TPP and warping rays of two LFs into the common TPP parameterization with subapertures aligned.

A. LF Warping

We use the conventional TPP to parameterize the LF, which contains the $ST$ plane (i.e., the main plane) and the $UV$ plane. In this parameterization, each ray in the LF is described by coordinates $[s, t, u, v]^T$. Here $[s, t]^T$ is the intersection point coordinate of the ray and the $ST$ plane, indicating the position of the ray and the subaperture to which the ray belongs. $[u, v]^T$ is the intersection point coordinate of the ray and the $UV$ plane relative to $[s, t]^T$, indicating the direction of the ray. In this model, we place the $ST$ plane in the main lens plane and the $UV$ plane one unit in front of the $ST$ plane.

To warp the LF, all the rays need to be warped. Given the rotation $R$ and translation $T$, we start with deriving the pairwise rays warping function. Without loss of generality, one LF $L$ is set as the global LF which can be represented by the set of each ray $r$ in it. For each ray, its corresponding coordinate $[s', t', u', v']^T$ in the second LF $L'$ is computed. This is done by intersecting $r$ with the TPP of $L'$ as shown in Fig. 4. The calculation is followed as:

$$
\begin{bmatrix}
X_{s1} \\
Y_{s1} \\
Z_{s1}
\end{bmatrix}
= R
\begin{bmatrix}
s \\
t \\
0
\end{bmatrix}
+ T
$$

(16)

and

$$
\begin{bmatrix}
X_{s2} \\
Y_{s2} \\
Z_{s2}
\end{bmatrix}
= R
\begin{bmatrix}
s + u \\
t + v \\
1
\end{bmatrix}
+ T
$$

(17)

where $[s, t, 0]^T$ is the coordinate of $P_1$, i.e., the intersection of $r$ and the $ST$ plane in $L$. $[s + u, t + v, 1]^T$ is the coordinate of $P_2$, i.e., the intersection of $r$ and the $UV$ plane. $[X_{s1}, Y_{s1}, Z_{s1}]^T$ and $[X_{s2}, Y_{s2}, Z_{s2}]^T$ are used to represent the coordinates of $P_1$ and $P_2$ in $L'$. Then we calculate the coordinates of $P_3$ and $P_4$, which are the intersections of $r$ with the TPP of $L'$, represented by $[s', t', 0]^T$ and $[s' + u', t' + v', 1]^T$, respectively. The coordinates are calculated as

$$
\begin{bmatrix}
s' \\
t'
\end{bmatrix}
= \frac{1}{\lambda_1}
\begin{bmatrix}
X_{s1} \\
Y_{s1} \\
Z_{s1}
\end{bmatrix}
+ \lambda_3
\begin{bmatrix}
X_{s2} \\
Y_{s2} \\
Z_{s2}
\end{bmatrix}
$$

(18)

and

$$
\begin{bmatrix}
s' + u' \\
t' + v'
\end{bmatrix}
= \frac{1}{\lambda_2}
\begin{bmatrix}
X_{s1} \\
Y_{s1} \\
Z_{s1}
\end{bmatrix}
+ \lambda_2
\begin{bmatrix}
X_{s2} \\
Y_{s2} \\
Z_{s2}
\end{bmatrix}
$$

(19)

where $\lambda_1 = (Z_{s1} - Z_{s2}) / (Z_{s1} - Z_{s2})$ and $\lambda_2 = (Z_{s1} - 1) / (Z_{s1} - Z_{s2})$. Finally, we substitute (16) and (17) into (18) and (19) and obtain (20), as shown at the bottom of the next page, where subscripts indicate the components of the rotation matrix or translation vector.

Due to the randomness of the relative pose, the directly warped LF is still not effective in enlarging the baseline. To illustrate the underlying problem, we set two cameras that only have a 15 mm vertical spacing and no rotation. Fig. 5 shows the warping result of two LFs with 11 subapertures each, where the red dots represent the subapertures of $L$, and the blue dots represent the subapertures of $L'$. As shown in Fig. 5(a), none of the subapertures from the LF pair can be located on a horizontal line, which implies that the baseline cannot be effectively enlarged. To solve this problem, we construct the third TPP and warp the rays of two LFs onto it.

B. Rectification Matrix

Rectification first attempts to maximize the number of subapertures that can be located on horizontal lines. Specifically, we construct a rotation matrix $R_{rect}$ that can map the two center-view subapertures onto the third $ST$ plane and align them horizontally. To obtain the largest baseline, $R_{rect}$ is created by starting with the translation vector $T$. The first row $e_1^T$ of the matrix is calculated as

$$
e_1 = \frac{T}{\|T\|}.
$$

(21)

To make sure the center-view subaperture only has horizontal displacement, the next vector $e_2$ must be orthogonal to $e_1$. This can be accomplished by using the cross-product
of $e_1$ with another direction. Selecting the vectorial sum of two principal rays can reduce reprojection distortion brought by warping two LFs. After normalization, another unit vector is calculated as

$$e_2 = \text{norm}\left(\begin{bmatrix} -r_2 s t_3 + t_2 (r_3 s + 1) \\ r_1 s t_3 - t_1 (r_3 s + 1) \\ -r_1 t_2 + r_2 s t_1 \end{bmatrix}\right)$$

(22)

where $\text{norm}(\cdot)$ represents the two-norm of a vector. In order to maintain the orthogonality of the rotation matrix, the third vector $e_3$ orthogonal to $e_1$ and $e_2$ can be computed by using the cross-product operation as

$$e_3 = e_1 \times e_2.$$  

Then the rectification matrix is calculated as

$$R_{\text{rect}} = \begin{bmatrix} e_1^\top \\ e_2^\top \\ e_3^\top \end{bmatrix}.$$  

(24)

The matrix rotates two $ST$ planes around their central subapertures, so that the two $ST$ planes can be coplanar, and the two central subapertures can have the same vertical coordinate in one coordinate system. The warping of two LFs into the third TPP parameterization is implemented by setting $R_l = R_{\text{rect}}$ and $R_r = R_{\text{rect}} R$. $R_l$ and $R_r$ are the rotation matrices of the left camera and the right camera, respectively. Since there is no translation between the TPP of the left camera and the third TPP, its translation vector $T_l = [0, 0, 0]^\top$. The translation vector between the TPP of the right camera and the third TPP is calculated as $T_r = R_{\text{rect}} T$.

### C. Interpolation of Row-Aligned SAI

After transforming the rays of the two LFs onto the third TPP by the method described in Section V-A, the distribution of subapertures in the third TPP are shown in Fig. 5(b). Apparently, the number of subapertures from the LFs that can be located on the horizontal line increases significantly. In order to obtain the images of aligned subapertures, a ray interpolation as shown in Fig. 6 is required. The $[s, t, u, v]^\top$ is the TPP parameters of the ray. For each synthetic subaperture that needs to be aligned, such as $(s_0, t_0)$, the pixel coordinates of SAI will be transformed to the ray coordinates $[s_0, t_0, u, v]^\top$ by the user-defined intrinsic parameters. Then the luminance of each ray is acquired by calculating the ray coordinate in its original TPP and implementing the quadratic interpolation. This means that the surrounding apertures

![Fig. 6. Ray interpolation on the regular grid. For each ray $[s_0, t_0, u, v]^\top$ in the row-aligned synthetic subaperture $(s_0, t_0)$, the rays surrounding four apertures $(s_i, t_i)$ with $i = 3, 4, 5, 6$ are used for interpolation.](image-url)
Fig. 7. LF rectification. (a) Original left and right SAI pairs. (b) Rectified left and right SAI pairs; note that the scan lines become aligned in the rectified images.

Fig. 8. EPI extracted from the rectified LF.

$(s_i, t_i)$ with $i = 3, 4, 5, 6$ are located, and in each of them, four rays are used to interpolate the ray $[s_i, t_i, u, v]^\top$. As shown in Fig. 7, since the subapertures are aligned horizontally, corners in these SAIs are aligned in rows. Not only the central SAIs are row-aligned, but also other view SAIs are row-aligned.

The corresponding EPI can also be extracted from the rectified LF as shown in Fig. 8. A row of pixels in an EPI image corresponds to a row of pixels in a SAI. All pixel rows are extracted from SAIs in the same row in Fig. 7(b). Since the baseline is enlarged, the line on EPI becomes much longer. Although the two LFs span a long distance, the lines on EPI still match well, which is attributed to the accurate relative pose estimation.

VI. EXPERIMENT AND ANALYSIS

A. Performance of Pose Estimation

In order to evaluate the performance of the proposed algorithm for pose estimation, an MLA-based conventional LF camera is simulated, whose intrinsic parameters are listed in Table I. These parameters come from the setting of a Lytro Illum camera so as to obtain plausible input.

For comparison purposes, we generate LF-points matches of checkerboard corners in two LFs based on different relative pose sets. LF-points of $7 \times 11$ corners with a spacing of 22.5 mm under different checkerboard poses are generated. The geometry of generating is based on the correspondence between the LF-point and its 3-D point. The correspondence model in [26] is used

$$
\begin{bmatrix}
  x_c \\
  y_c \\
  \lambda \\
  1
\end{bmatrix} =
\frac{1}{Z}
\begin{bmatrix}
  f_x & 0 & c_x & 0 \\
  0 & f_y & c_y & 0 \\
  0 & 0 & -K_1 & -K_2 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}.
$$

According to the known corner coordinates $[X, Y, Z]^\top$ and the preset relative poses, the LF-points in the two LFs are generated, respectively, by (25).

As described in [23], the SAIs are evenly arranged in the image plane, which means the disparities of the 3-D point obtained from different adjacent SAIs are assumed to be equal. Therefore, given the ground truth of LF-points, we can utilize the corner coordinate $(x_c, y_c)$ extracted in the central SAI and the disparity $\lambda$ to generate corresponding corners in $11 \times 11$ SAIs. To simulate the realistic detection errors, independent Gaussian noises with mean zero and varying standard deviations $\sigma$ are added on the locations of these projected corners. For each noise level or pose configuration, 100 independent trials are performed and the average result is considered as the final performance. For a fair comparison, suitable input for compared algorithms is computed using these projected points.
in SAI. The relative poses are set considering actual scenarios so that the corners can be observed in both LF cameras.

As the ground truth of relative pose is known, the performance of relative pose estimation algorithms can be evaluated. The angular errors of relative poses which have been widely used such as in [19], [20], [22], and [33] are adopted for measurement, represented as

\[
\text{Err}_R = \frac{180}{\pi} \arccos \left( 0.5 \text{trace} \left( (R_{\text{est}}^\top R) - 1 \right) \right)
\]

and

\[
\text{Err}_T = \frac{180}{\pi} \arccos \left( \frac{\text{dot}(T, T_{\text{est}})}{\|T\| \|T_{\text{est}}\|} \right)
\]

where \(\text{Err}_R\) and \(\text{Err}_T\) are angular errors of the rotation matrix and translation vector, respectively. \(\arccos(\cdot)\) represents an arccosine function. \(\text{trace}(\cdot)\) calculates the trace of a matrix. \(\text{dot}(\cdot, \cdot)\) represents the dot product operation. \(R\) and \(T\) are the actual rotation matrix and translation vector, respectively. \(R_{\text{est}}\) and \(T_{\text{est}}\) are the estimated rotation matrix and translation vector. \(R_{\text{est}}^\top\) is the transpose of \(R_{\text{est}}\). In addition, the Euclidean distance which calculates the displacement between the true camera position and the estimated camera position is also used. Under these commonly used error metrics, the performance of our proposed algorithm (Proposed) is compared with pose estimation algorithms in [19] and [20] (LF-SfM), [21] (LF-DLT), and the work [22] (LFP).

1) Errors With Respect to Different Noise Levels: In this experiment, the measurements of rotation angles set at \((5^\circ, 20^\circ, 5^\circ)\) and translation set at \([150, 5, 5]^\top\) are employed to verify the robustness of estimation algorithms. Gaussian noises with the standard deviation \(\sigma\) varying from 0.1 to 0.5 pixels with a step of 0.1 pixel are added to all projected corners to represent the increasing noise levels. In addition, the extreme cases of \(\sigma = 2, 3\) are also applied. As shown in Table II, a higher noise level makes for larger errors on the rotation matrix and translation vector for all algorithms. When the noise is at a low level, the errors of rotation and translation estimated by LF-SfM are relatively small. However, the errors grow rapidly and exceed those estimated by other algorithms when the noise increases gradually, which indicates the ray-based algorithm is not robust to noise. For LF-DLT, the error of translation estimated is almost always the largest and the error of rotation estimated increases relatively fast, since LF-DLT will be disturbed by the inaccurate 3-D points reconstructed through the small baseline. In most cases, the proposed algorithm achieves the best performance and LFP ranks second to the proposed algorithm. In extreme cases \((\sigma = 2, 3)\), the errors of the proposed algorithm are significantly smaller than those estimated by the other algorithms, which indicates the robustness of the proposed algorithm.

2) Errors With Respect to Different Poses: This experiment investigates the performance with respect to different relative poses. The rotation angles are \((5^\circ, 15^\circ, 5^\circ)(R_1)\) and \((5^\circ, 40^\circ, 5^\circ)(R_2)\). The translation along x-axis are 100 mm \((T_x)\) and 200 mm \((T_y)\). For each combination of rotation and translation, the independent Gaussian noises with zero mean and standard deviation \(\delta\) of 0.1, 0.3, 0.5, and 3 pixels are added to all projected points, respectively. The results of performance are shown in Table III. When the rotation is fixed and translation increases, the errors of rotation and translation estimated with LF-SfM also increase, which indicates the ray-based algorithm is more suitable for a small translation. The angular errors and distance errors of translation estimated with LF-DLT are always the largest. Especially when the translation is small, the errors of LF-DLT are significantly larger than the other algorithms. The performance variation of rotation of LFP is the same as that of the proposed algorithm. When the rotation is fixed and translation increases, the errors of rotation increase significantly. When the translation is fixed and rotation increases, the errors of rotation change slightly, which indicates the proposed algorithm is not sensitive to rotation. As for the performance of translation, when the translation is fixed, the errors of LFP are always the largest. Especially when the translation is small, the errors of LF-DLT are significantly larger than the other algorithms. The performance variation of rotation of LFP is the same as that of the proposed algorithm.

| Algorithm | LF-SfM | LF-DLT | LFP | Proposed |
|-----------|--------|--------|-----|----------|
| Angular error of \(R\) |
| 0.1 | 0.0535 | 0.0893 | 0.0487 | 0.0323 |
| 0.2 | 0.1322 | 0.3060 | 0.1094 | 0.0658 |
| 0.3 | 0.2731 | 0.6835 | 0.2466 | 0.1119 |
| 0.4 | 0.4233 | 1.2773 | 0.2893 | 0.1583 |
| 0.5 | 0.5013 | 2.0258 | 0.4130 | 0.2014 |
| 2 | 5.0864 | 7.4815 | 3.9349 | 1.0200 |
| 3 | 7.8477 | 6.2613 | 5.9064 | 1.7312 |
| Angular error and relative Euclidean distance error of \(T\) |
| 0.0531/0.2063 | 0.2249/0.6909 | 0.1339/0.4842 | 0.0893/0.3366 |
| 0.1270/0.7084 | 0.6535/2.5285 | 0.2494/0.9425 | 0.1621/0.6563 |
| 0.3057/1.5256 | 1.4914/5.6671 | 0.3750/1.6369 | 0.2084/0.1012 |
| 0.4463/2.7058 | 2.5816/10.0115 | 0.4330/2.5329 | 0.4081/1.4135 |
| 0.6382/4.0791 | 3.6532/16.3061 | 0.3812/3.3244 | 0.4303/1.6978 |

B. Performance of Rectification

In this section, the rectification performance and the scene reconstruction effects on real scenes are evaluated. We installed two calibrated Lytro Illum cameras in proper poses to make sure scenes are in the field of view of both cameras as shown in Fig. 9. Five real scenes with different relative poses are captured by the Lytro Illum cameras. Central SAIs of LFs are shown in Fig. 10. The intrinsic parameters of LF cameras are listed in Table IV. The LF points \((x_c, y_c, \lambda)\) are first extracted in SAIs which are easy to be obtained for a pre-calibrated LF camera [23]. Since the disparity \(\lambda\) calculated in real scenes is not sufficiently accurate to be used to estimate the relative pose that can support effective LF rectification, we use checkerboard to generate
TABLE III

| Angular Error (Degree) of Simulation Under Different Poses |
|------------------------------------------------------------|
| Algorithm        | LF-SIM | LF-DLT | LFP | Proposed |
|------------------|--------|--------|-----|----------|
| $\sigma = 0.1$   |        |        |     |          |
| $R_1, T_1$       | 0.0428 | 0.0860 | 0.0355 | 0.0406 |
|                  | 0.0407/0.1611 | 0.2366/0.7013 | 0.1305/0.3522 | 0.1532/0.3271 |
| $R_1, T_2$       | 0.0722 | 0.0920 | 0.0774 | 0.0523 |
|                  | 0.0485/0.3555 | 0.1407/0.7884 | 0.1335/0.6121 | 0.0945/0.2861 |
| $R_2, T_1$       | 0.0364 | 0.0713 | 0.0362 | 0.0336 |
|                  | 0.0669/0.1287 | 0.2926/0.6200 | 0.1487/0.3044 | 0.1392/0.3500 |
| $R_2, T_2$       | 0.0522 | 0.0983 | 0.0765 | 0.0397 |
|                  | 0.0487/0.2354 | 0.2061/0.7769 | 0.1643/0.5853 | 0.0865/0.2161 |
| $\sigma = 0.3$   |        |        |     |          |
| $R_1, T_1$       | 0.1954 | 0.7039 | 0.1249 | 0.1185 |
|                  | 0.3234/0.9971 | 2.3036/5.6631 | 0.3732/0.1075 | 0.3636/1.1903 |
| $R_1, T_2$       | 0.3526 | 0.8099 | 0.3029 | 0.1302 |
|                  | 0.2386/2.3889 | 1.1084/6.6863 | 0.4600/2.6422 | 0.1993/0.3898 |
| $R_2, T_1$       | 0.1351 | 0.6296 | 0.1239 | 0.0732 |
|                  | 0.3121/0.6951 | 2.5460/5.3323 | 0.3293/0.9957 | 0.2533/0.1071 |
| $R_2, T_2$       | 0.2484 | 0.6223 | 0.2658 | 0.1356 |
|                  | 0.3326/1.7496 | 1.2124/5.1842 | 0.4668/2.0697 | 0.2909/0.9094 |
| $\sigma = 0.5$   |        |        |     |          |
| $R_1, T_1$       | 0.3864 | 2.0946 | 0.2553 | 0.1692 |
|                  | 0.5701/2.5983 | 5.7325/17.2375 | 0.6084/2.4136 | 0.5602/0.2076 |
| $R_1, T_2$       | 0.8199 | 2.3964 | 0.5833 | 0.2547 |
|                  | 0.5930/6.3524 | 3.1856/19.5317 | 0.6751/5.2762 | 0.4175/1.6041 |
| $R_2, T_1$       | 0.2971 | 1.9816 | 0.2568 | 0.1667 |
|                  | 0.7780/1.8075 | 7.4938/15.7218 | 0.7678/1.9411 | 0.7126/0.2691 |
| $R_2, T_2$       | 0.5662 | 1.8009 | 0.4592 | 0.2577 |
|                  | 0.8795/4.4733 | 3.5309/15.1842 | 0.6679/4.1762 | 0.5175/1.5912 |
| $\sigma = 3$     |        |        |     |          |
| $R_1, T_1$       | 5.8400 | 6.3509 | 3.0807 | 2.1729 |
|                  | 14.9525/45.6048 | 19.3517/49.0182 | 11.2466/32.8916 | 4.5121/14.9266 |
| $R_1, T_2$       | 10.8913 | 6.6313 | 9.5658 | 2.0034 |
|                  | 16.9402/9.8082 | 17.0298/9.1613 | 20.2721/7.8692 | 2.9321/10.0756 |
| $R_2, T_1$       | 4.2033 | 6.1505 | 3.1969 | 2.1666 |
|                  | 18.2935/35.1591 | 20.2412/41.7218 | 12.0592/26.7064 | 4.9629/16.4373 |
| $R_2, T_2$       | 8.4184 | 6.0578 | 6.6864 | 2.2518 |
|                  | 18.9851/70.2867 | 17.7494/74.2737 | 15.8789/59.8583 | 3.7074/12.2454 |

TABLE IV

| Intrinsic Parameters of LF Cameras (Pixel) |
|-------------------------------------------|
| $f_x, f_y$ | $c_x, c_y$ | $K_1$ | $K_2$ |
|------------|------------|-------|-------|
| LF camera 1 | 639.670  | 637.707 | 258.408 | 187.291 | 0.117 | 273.282 |
| LF camera 2 | 646.080  | 639.792 | 247.304 | 196.522 | 0.166 | 327.966 |

LF-point pairs. Specifically, the corner detection function detectCheckerboardPoints in MATLAB R2019a is applied on the central SAI to obtain the pixel coordinates $(x_c, y_c)$. Then the disparity parameter $\lambda$ is estimated by implementing the corner detection in all pairwise combinations of collinear SAIs, and the final value is the median value of all pixel coordinate differences. Furthermore, the LF-point-LF-point correspondences can be extracted by sorting the pixel coordinates. After the relative pose is estimated, the rectification of real scenes can be implemented. Further, 3-D point clouds can be recovered from the central SAI and its depth. For a single LF, we directly use its central SAI and depth estimated in [13]. For the rectified LF, we choose one of the rectified central SAIs and use depth estimated in [13].

1) Row Alignment of the Rectified SAIs: Due to the fact that the rectification matrices and vectors are only composed of the relative pose, the rectification errors can be used to evaluate the accuracy of pose estimation methods. In this experiment, the rectification under different pose estimation algorithms is implemented. The rectified SAIs are shown to assess the performance of different algorithms on rectification. The depth estimation requires that the projected points of a scene point in the images of the same row of subapertures have the same vertical coordinate. Therefore, the smaller vertical coordinates differences of corresponding feature points mean better rectification performance. The rectified central SAIs of Box (RT1) on each LF are shown in Fig. 11. The vertical coordinates of feature points on each scene are circled on the images. From these results, we can see that the pose estimation based on rays and 3-D points has the largest rectification errors. This is due to the narrow baseline corrupting the rays or points. On the contrary, the LF-point-based method, i.e., LFP and the proposed pose estimation, can achieve smaller errors in the vertical coordinates. Furthermore, the proposed method removes the influence of the intrinsic parameters on LF-points and has the minimum errors. As a result, the proposed relative pose estimation is chosen to continue the subsequent rectification, depth estimation, and reconstruction.

2) Layered Effect: Due to the inherent characteristic of a small baseline in a hand-held LF camera, the disparity between adjacent SAIs is usually less than 1 pixel, which generally leads to a layered effect in reconstructed point clouds. This experiment aims to compare the layered effects in the single LF and the rectified LF to show the advantage of enlarging the baseline. A commonly used depth estimation algorithm [13], which takes the advantages of EPIs extracted from multiview SAIs and is designed for a single LF is performed on the single and rectified LFs to output the respective depths. As shown in Fig. 12, point clouds with or without rectification all have a good outline of the object from the front view. However, when observed from the side and top views, the layered effect of reconstructions in the single LF is very severe [as shown in Fig. 12(e) and (f)]. An observation of Fig. 12(h) and (i) indicates that, even though the layered effect of the reconstruction after subpixel refinement has been alleviated, it is still visible. On the other hand, the rectified LF shows no apparent

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
layered effect [see Fig. 12(b) and (c)], which demonstrates the effectiveness of the proposed rectification. Equation (2) in Section III represents a differential change of depth \( Z \) derived from the formula of disparities to depths. From the equation, we can see that when the minimum disparity interval \( dD \) is fixed and the baseline \( B \) is effectively enlarged, the depth interval \( dZ \) will be reduced. In the single LF captured by the LF camera, the baseline usually does not exceed one millimeter, which is the reason why the layered effect is so obvious. In the rectified LF, the baseline is extended to more than a hundred of millimeters, thus the layered effect has been almost eliminated and the depth becomes dense and continuous. Finally, the effectiveness of the proposed LF rectification is verified.
3) Drastic R and T Settings: In this section, more drastic R and T settings are employed to verify the robustness of the rectification. The relative poses estimated by the proposed algorithm are shown in Fig. 9(b). The red camera represents the reference pose and cameras in other colors represent the relative poses in different R and T settings. RT1 corresponds to location setup shown in Fig. 9(a). From Fig. 9(b), we can see that RT2 has a larger relative displacement than RT1. RT3 and RT4 have larger relative rotations than RT1. In the setting of RT4, the SAIs of the corresponding scene Box-Jar (RT4) have occurred occlusion as shown in Fig. 10(i) and (j).

To further efficiently evaluate the performance of different pose estimation algorithms, we detect corners in rectified 9 × 9 SAIs of the checkerboard of two LFs. The mean vertical coordinate differences of a total of 6237 pairs of corners are calculated and results are shown in Table V. From the results, we can see that the proposed pose estimation has minimum rectification errors and the errors are less than 1 pixel in all cases, so as to achieve the best performance.

In addition to the Box (RT1), four scenes are tested in these settings, and the performance of rectification shown in Fig. 13. For better visualization, only 3 × 3 SAIs are displayed. As the change of R and T increases, the range of rectification also increases, but the scan lines are still aligned. Subsequent reconstructions of scenes are shown in Fig. 14. For scenes Box (RT2), Books (RT3), and Box-Books (RT3), we can see that the layered effect does not appear and the depth distribution is continuous and dense. Especially, from Fig. 10(i) and (j), we can see that part of the Box is occluded by the Jar in...
Fig. 13. Rectification of different scenes. (a) Rectified SAIs of Box (RT2). (b) Rectified SAIs of Books (RT3). (c) Rectified SAIs of Box-Books (RT3). (d) Rectified SAIs of Box-Jar (RT4).
since LF rectification indexes rays from the corresponding original LF, the occlusion still exists in the rectified left SAIs. Consequently, the occluded part can only be matched within the single left rectified LF. In the reconstruction shown in Fig. 14(h), the depths of the occluded part shows a layered effect and depths of other common-view parts are continuous and dense.

In addition, we conducted extended experiments to compare the 3-D reconstruction quality based on the rectified LF with that of the traditional stereo vision method using two conventional cameras. Two Sony α−6400 cameras are used to capture the scenes. The stereo calibration pipeline in MATLAB R2019a is implemented to provide the intrinsic parameters of the two conventional cameras. State-of-the-art algorithms semi-global block matching (SGBM) [46] and graph cut (GC) [47] are chosen to provide the depths. Fig. 15 shows the 3-D reconstructions of Box and Box-Jar. The reconstructions of Books and Box-Books from which the same observations can be found are put in Section IV of the supplemental document. Through the comparison between Figs. 14(h) and (g), and 15(a) and (c), we can see that
TABLE VI

| measured distances | Box (RT1) | Box (RT2) | Books (RT3) | Box-Books (RT3) | Box-Jar (RT4) |
|--------------------|-----------|-----------|-------------|----------------|--------------|
| LFS                | 241.7 (4.85 %) | 240.8 (5.21 %) | 224.2 (4.17 %) | 223.6 (4.44 %) | 182.9 (3.74 %) |
| SPO                | 237.7 (2.49 %) | 232.4 (4.56 %) | 225.9 (3.17 %) | 226.6 (3.15 %) | 184.9 (2.71 %) |
| the proposed method| 251.3 (1.06 %) | 248.7 (2.09 %) | 230.1 (1.67 %) | 228.3 (2.44 %) | 183.5 (0.79 %) |

The relative error is indicated in parentheses.

Fig. 16. Measurements between specific points. (a) Box: 254.0 mm. (b) Books: 234.0 mm. (c) Jar: 190.0 mm.

the reconstructions based on the rectified LF produces more consistent object contours than the reconstructions based on SGBM. This is particularly evident in boundaries of scenes such as the boundaries of Box and Jar, where the traditional method tends to generate incomplete result. On the other hand, due to the high resolution of the captured images, the details of the reconstruction based on SGBM are better than those based on the proposed method. Additionally, since the traditional stereo also has a large baseline, the depths are denser and more continuous than the reconstructions based on the single LF shown in Fig. 12. GC is a graph cut-based method offering a global optimization framework that can model the depth estimation as an energy minimization problem. The GC method uses smoothness constraints to enforce spatial coherence in the depth map, which produces more complete boundaries as shown in Fig. 15(e) and (g) but simultaneously penalizes changes in depth between neighboring pixels, causing marked steps on the depths as shown in Fig. 15(f) and (h). In summary, the rectified LF takes advantage of the high angular resolution and the large baseline to producing the reconstructions with good contours and dense and continuous depths. However, due to the trade-off of the spatial resolution and angular resolution, the details of the reconstructions are not as good as those of the traditional stereo vision.

4) Distance Measuring: In this experiment, the metric reconstruction based on the rectified LF is implemented. In the metric reconstruction, the depth estimation is implemented under the framework [13], which estimates the depths within a single LF. After rectification, the depth estimation is performed between the row-aligned SAI sets. With a depth of the scene, the point cloud can be reconstructed using the central SAI and intrinsic parameters of the LF camera. To further show the enlarged baseline advantage of the rectified LF, different LF depth estimation methods including LFS [13] and spinning parallelogram operator (SPO) [48] along with corresponding 3-D reconstructions are implemented to make a comparison. The measurement distances between several specific points are shown in Fig. 16, and the reconstruction errors of the estimated distance between reconstructed points are listed in Table VI. From the results, we can see that, with the same intrinsic parameters of the LF cameras, the distances estimated by LFS and SPO are not as accurate as the distance estimated by the proposed pipeline based on the rectified LF.

5) Ablation Study on Direct Warping: Since direct warping can already transform the rays of two LFs into a common TPP parameterization, i.e., warping one LF into the TPP parameterization of another LF, and extract the row-aligned SAI, we present the ablation study to verify the usefulness of the constructed third TPP. Fig. 17(a) and (b) show SAI of Box (RT1) from the two methods, respectively. For direct warping, even in an almost side-by-side camera installation (RT1), most SAI of the direct warped LF (right) cannot find their corresponding SAI (left) in that many warped rays cannot be located on the regular grid on the main plane of the un-warped LF. Furthermore, many parts in the SAI of the warped LF are missing because the rays corresponding to these missing pixels are not captured by the original LF. In drastic R and T settings, since none of the subapertures from the LF pair can be located on a horizontal line, the row-aligned SAI can be generated. In contrast, as shown in Fig. 17(b), the SAI from the complete rectification method do not have such problems, which demonstrates the necessity of introducing the third TPP.

VII. Conclusion

In this article, we propose to rectify LFs captured by two hand-held LF cameras. In the relative pose estimation, we propose a linear approach to estimate the projective
transformation between the LF-point pairs and extract the relative pose. In the rectification part, the constructed rectification matrix and vector are employed to align the two LFs. Row-aligned SAs in all subviews are then generated. These SAs are applied in-depth estimation to achieve depth with high-depth resolutions. The proposed pose estimation algorithm outperforms the classical and state-of-the-art algorithms, which shows the superiority of the proposed algorithm. The effectiveness of the proposed method is also verified in real scenes. In the future, we would like to design proper depth optimization algorithms to exploit the potential of rectified LF and apply it in more visual tasks, such as refocusing and perspective synthesis. We will also conduct further studies on the relative pose estimation in terms of enhancing the ability to extract LF features in natural scenes. Since there is an emerging trend of including plenoptic imaging capabilities in smartphones with regularly arranged rear cameras, designing LF rectification suitable for mobile applications is also an interesting direction. One can utilize the high-resolution images captured by the smartphone to form a high-resolution LF to achieve a high-quality 3-D reconstruction.

REFERENCES

[1] Lytro. (2016). The Lytro Camera. [Online]. Available: http://www.lytro.com/

[2] Raytrix. (2016). 3D Light Field Camera Technology. [Online]. Available: http://www.raytrix.de/

[3] F. Dong, S.-H. Ieng, X. Savatier, R. Etiene-Cummings, and R. Benoist “Plenoptic cameras in real-time robotics,” Int. J. Robot. Res., vol. 32, no. 2, pp. 206–217, Feb. 2013.

[4] N. Zeller, F. Quint, and U. Stilla, “From the calibration of a light-field camera to direct plenoptic odometry,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 7, pp. 1004–1019, Oct. 2017.

[5] C. Zhu, H. Zhang, and L. Yu, “Structure models for image-assisted geometry measurement in plenoptic sampling,” IEEE Trans. Instrum. Meas., vol. 67, no. 1, pp. 150–166, Jan. 2018.

[6] Y. Yang, L. Wu, L. Zeng, T. Yao, and Y. Zhan, “Joint upsampling for refocusing light fields derived with hybrid lenses,” IEEE Trans. Instrum. Meas., vol. 72, pp. 1–12, 2023.

[7] J. Jin, J. Hou, J. Chen, and S. Kwong, “Light field spatial super-resolution via deep combinatorial geometry embedding and structural consistency regularization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2257–2266.

[8] Z. Hu, X. Chen, H. W. F. Yeung, Y. Y. Chung, and Z. Chen, “Enhanced light field super-resolution with spatio-angular decomposition kernels,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–16, 2022.

[9] C. Jia et al., “Semantic segmentation with light field imaging and convolutional neural networks,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–14, 2021.

[10] E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 99–106, 1992.

[11] S. Wanner and B. Goldluecke, “Globally consistent depth labeling of stereo images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 3287–3296.

[12] Q. Zhang, H. Li, X. Wang, and Q. Wang, “3D scene reconstruction with an un-calibrated light field camera,” Int. J. Comput. Vis., vol. 129, no. 11, pp. 3006–3026, Nov. 2021.

[13] D. V. Papadimitriou and T. J. Dennis, “Epipolar line estimation and rectification for stereo image pairs,” IEEE Trans. Image Process., vol. 5, no. 4, pp. 672–676, Apr. 1996.

[14] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” Mach. Vis. Appl., vol. 12, no. 1, pp. 16–22, Jul. 2000.

[15] 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.
Xiao Huo received the B.E. degree from Xidian University, Xi’an, China, in 2019, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Services Networks. His current research interests include light field camera calibration, light field imaging, and 3-D reconstruction.

Dongyang Jin received the B.E., M.E., and Ph.D. degrees from Xidian University, Xi’an, China, in 2009, 2013, and 2020, respectively. He is currently a Lecturer with the School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin, China. His research interests include SLAM, visual navigation, and aerial photogrammetry.

Saiping Zhang (Student Member, IEEE) received the B.E. degree in telecommunication engineering and the Ph.D. degree in telecommunication and information system from Xidian University, Xi’an, China, in 2016 and 2022, respectively. Her current research interests include light field technologies, video coding, and multimedia communication.

Fuzheng Yang (Member, IEEE) received the B.E. degree in telecommunication engineering and the M.E. and Ph.D. degrees in communication and information system from Xidian University, Xi’an, China, in 2000, 2003, and 2005, respectively. He became a Lecturer and an Associate Professor with Xidian University, in 2005 and 2006, respectively, where he has been a Professor of communications engineering, since 2012. From 2006 to 2007, he worked as a Visiting Scholar and Post-Doctoral Researcher with the Department of Electronic Engineering, Queen Mary University of London, London, U.K. His current research interests include light field technologies, video quality assessment, video coding, and multimedia communication.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.