Geometric Calibration of Micro-Lens-Based Light-Field Cameras Using Line Features

Yunsu Bok, Hae-Gon Jeon, and In So Kweon

KAIST, Korea

Abstract. We present a novel method of geometric calibration of micro-lens-based light-field cameras. Accurate geometric calibration is a basis of various applications. Instead of using sub-aperture images, we utilize raw images directly for calibration. We select proper regions in raw images and extract line features from micro-lens images in those regions. For the whole process, we formulate a new projection model of micro-lens-based light-field cameras. It is transformed into a linear form using line features. We compute an initial solution of both intrinsic and extrinsic parameters by a linear computation, and refine it via a non-linear optimization. Experimental results show the accuracy of the correspondences between rays and pixels in raw images, estimated by the proposed method.

Keywords: Calibration, plenoptic, light-field cameras.

1 Introduction

A light-field or plenoptic camera captures angular and spatial information on the distribution of light rays in space, which obtains a multi-view of a scene in a single photographic exposure. The concept of a light-field camera was proposed by Adelson and Wang [1], and light-field photography has been an emerging technology in recent years.

For light-field acquisition, Wilburn *et al.* [17] presented a bulky camera array. The system can obtain a light-field image with high spatial and angular resolution, but is very expensive. Liang *et al.* [9] encoded angular information of light rays using programmable aperture patterns. Veeraraghavan *et al.* [15] presented a simple light-field acquisition technique based on a transparent mask attached at the front of a camera's sensor. Taguchi *et al.* [12] used hemispherical mirrors for light-field rendering with a wide field of view. However, these approaches have some impediments to commercialization due to manufacturing cost or low quality of the light-field image.

Ng [10] proposed a hand-held light field camera using a micro-lens array. Ng augmented a camera sensor by placing a micro-lens array in front of it. Each micro-lens plays a role in a tiny sharp image of the lens aperture, estimating the directional distribution of incoming rays through it. Georgiev and Lumsdaine [7] presented a modified version of Ng's model that interprets the micro-lens array as an imaging system focused on the focal plane of the main camera lens. Their system is able to capture a light-field image with higher spatial resolution

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 47–61, 2014.
© Springer International Publishing Switzerland 2014

than Ng's model, but its angular resolution is decreased. Based on these approaches using a micro-lens array, commercial light-field cameras such as Lytro and Raytrix have been released.

In the computer vision and graphics field, research on applications using hand-held light-field cameras has garnered interest. A representative application of light-field cameras is refocusing [10], which changes the in-focus region of an image after capturing it. A major drawback of such hand-held systems is low spatial resolution. In order to overcome this limitation, several light-field super-resolution methods have been developed [3,11]. Light-field panorama [2] and disparity estimation [13,16] are also interesting applications. In the robotics field, light-field cameras composed of a camera array have shown their usefulness for visual odometry and visual SLAM [6,4]. As is widely known, the performance of these applications can be enhanced if the geometric information of hand-held light-field cameras is available. However, there has been few works on the geometric calibration of hand-held light-field cameras.

An earlier work [14] dealing with a camera array system is based on combining plane and parallax methods. Johannsen *et al.* [8] present a metric calibration method using a dot pattern with a known grid size and a depth distortion correction for focused light-field cameras [7]. The most similar previous works to the present study is that of Dansereau *et al.* [5], who proposed a geometry calibration approach for commercial light-field cameras. They modeled pixel-to-ray correspondences of commercial light-field cameras in 3D space, and presented a 4D intrinsic matrix from a conventional pinhole and thin-lens model. However, it has remaining issues such as estimating the initial values of the cameras physical parameters.

In this paper, we present a novel geometric calibration method for micro-lens-based light-field cameras. Instead of using sub-aperture images, we utilize raw images of light-field cameras directly. Since conventional methods of generating sub-aperture images are based on assumptions on light-field cameras, they must be generated 'after' geometric calibration of raw images. We present a new formulation of the projection model of micro-lens-based light-field cameras. We extract line features from raw images and compute an initial solution of both intrinsic parameters and extrinsic parameters of light-field cameras by a linear method. The initial solution is then refined via a non-linear optimization.

2 Line Features from Raw Image

The most important data for calibration of any sensor are correspondences between known environments and sensor measurements. For example, we usually detect corners of a checkerboard with known size for conventional camera calibration. Figure 1 shows examples of raw images captured by Lytro, a popular micro-lens-based camera. It is difficult to extract precise locations of checkerboard corners in small micro-lens images. We have observed that border lines of black and white regions are more visible than corners in micro-lens images. In this paper, we extract line features from raw images and utilize them to calibrate micro-lens-based light-field cameras.

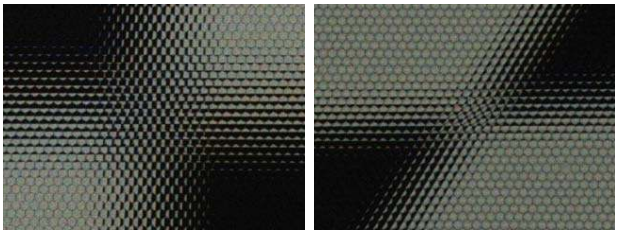
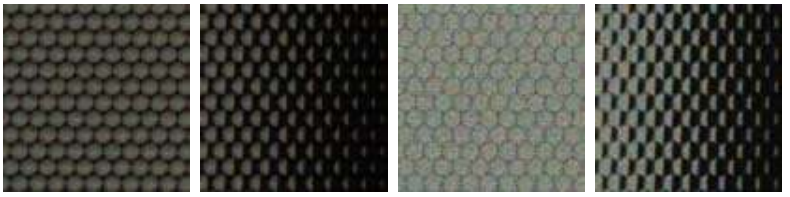


Fig. 1. Examples of raw images captured by Lytro. It is difficult to extract precise locations of checkerboard corners from raw images. However, lines are relatively visible in small micro-lens images.



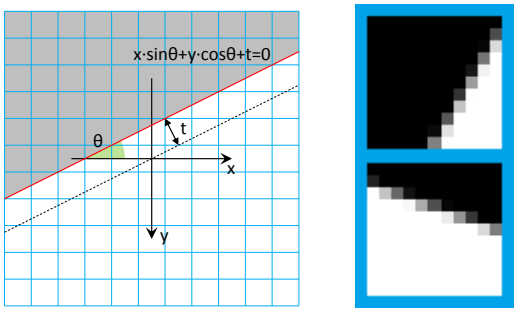
(a) (b) (c) (d)

Fig. 2. Vignetting effect of micro lenses. Left two figures (a, b) and right two figures (c, d) are examples of raw images with vignetting and without vignetting, respectively. Raw images are divided by white-plane images to remove the vignetting effect. Checkerboard images without vignetting (d) are better than those with vignetting (b) for line feature extraction.

Extraction of line features consists of two steps: (1) selection of micro-lenses to extract lines and (2) computation of line parameters for every micro-lens. We assume that the centers of micro-lenses are already known by any means, such as using data given by manufacturers or computing centers from white-plane images. White-plane images with vignetting are used for two purposes: estimating micro-lens centers and removing the vignetting effect from raw images. As shown in Fig. 2(b), the vignetting effect of micro-lenses makes it difficult to estimate the accurate location of line features. We simply divide raw images by white-plane images to generate images without vignetting. In this paper, we use raw images from which the vignetting effect of micro lenses is removed (see Fig. 2(d)).

2.1 Line Feature Extraction

Micro-lens images are usually too small (10×10 pixels for Lytro) to be applied to conventional line fitting techniques. In this paper, we propose an indirect line fitting method. Instead of extracting lines directly from micro-lens images, we generate a number of samples with known line parameters and compare them to actual micro-lens images.



(a) (b)

Fig. 3. Template generation for line feature extraction. (a) Line equation is determined by an angle θ and a translation t. (b) Example of template. Top: $\theta = 60$(deg), $t = -2.5$(pixel), bottom: $\theta = -20$(deg), $t = 1.7$(pixel).

Let (u_c, v_c) be the center of a micro-lens image. First we prepare a small rectangular template whose side length is equal to the diameter of the micro-lenses (11 pixels in our implementation). The center pixel of the template corresponds to the pixel of the micro-lens image closest to its actual center. Let (u_r, v_r) be the round-off result of (u_c, v_c), and (u'_r, v'_r) the center pixel of the template ($u'_r = v'_r = 6$ in our implementation). The micro-lens center (u'_c, v'_c) in the template is defined as

$$\begin{bmatrix} u'_c \\ v'_c \end{bmatrix} = \begin{bmatrix} u'_r \\ v'_r \end{bmatrix} + \left(\begin{bmatrix} u_c \\ v_c \end{bmatrix} - \begin{bmatrix} u_r \\ v_r \end{bmatrix} \right). \tag{1}$$

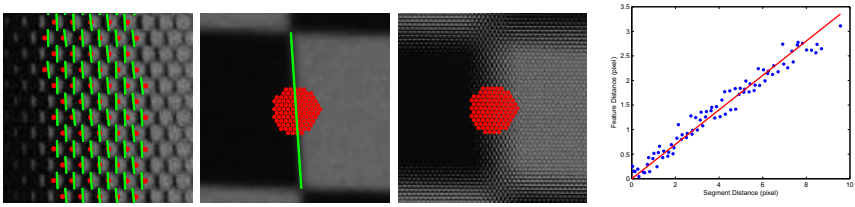
Templates are generated by varying the rotation angle θ and translation t of a line (see Fig. 3(a)). Setting the center pixel of the template as $(0, 0)$, a line equation is defined as

$$x \cdot \sin\theta + y \cdot \cos\theta + t = 0. \tag{2}$$

Examples of template are shown in Fig. 3(b).

We generate templates with varying parameters $-90 \leq \theta \leq 90$(deg) and $-$(radius) $\leq$ t $\leq$ (radius) (radius=5(pixels) in our implementation using Lytro). They are compared to micro-lens images via normalized cross-correlation (NCC). We compute the 'absolute value' of 'weighted' NCC using Gaussian weight to consider inverted templates and ignore the outside region of micro-lens images. After selecting a template with the maximum NCC value, we adjust the constant term of its line equation to set the actual micro-lens center in the template (1) as the origin:

$$\begin{aligned} &(x + (u_c - u_r)) \cdot \sin\theta + (y + (v_c - v_r)) \cdot \cos\theta + t \\ &= x \cdot \sin\theta + y \cdot \cos\theta + t + (u_c - u_r)\sin\theta + (v_c - v_r)\cos\theta \\ &= x \cdot \sin\theta + y \cdot \cos\theta + t'. \end{aligned} \tag{3}$$



(a) (b) (c) (d)

Fig. 4. Upper-limit distance estimation of line segment. (a) 'Feature distance' refers to the distance between the micro-lens center (red dot) and the line feature (green line) in the raw image. (b) 'Segment distance' is the distance between the micro-lens center (red dot) and line segment (green line) connecting two adjacent corners in the sub-aperture image. (c) An example of N nearest micro-lens centers from the center of a line segment. (d) Relation between feature distance (vertical axis) and segment distance (horizontal axis) is estimated via line fitting.

2.2 Micro Lens Selection

In a raw image of a checkerboard, micro-lenses are classified into three categories: corner, line, and homogeneous. We extract line features from only micro-lens images that contain border lines of black and white regions, not corners or homogeneous regions. In order to identify the class of each micro-lens, we utilize a sub-aperture image at the center (i.e., a collection of center pixels of micro-lens images). This is based on the conventional assumption on micro-lens-based light-field cameras, but is sufficient to provide useful information for the selection.

The distance of a line feature from the micro-lens center in a raw image (feature distance, see Fig. 4(a)) is nearly proportional to that of a line segment from the micro-lens center in a sub-aperture image (segment distance, see Fig. 4(b)). For each line segment connecting adjacent corners of checkerboard, we measure the upper limit of the segment distance that guarantees the existence of a line feature in the micro-lens. This must be measured for each line segment because it depends on the distance from the camera to the checkerboard in the real world.

We compute the center of a line segment in a sub-aperture image, and then select N nearest micro-lens centers from it (see Fig. 4(c)). A line feature is extracted from each micro-lens center. We compute the feature distance and estimate the relation between the feature distance and the segment distance (see Fig. 4(d)). We consider a segment distance corresponding to a user-defined feature distance as the upper limit of the segment distance of a line segment.

Each line segment in a sub-aperture image has a small region where line features exist in the raw image. This is shown in Fig. 5. For each line segment, there are at most four neighboring line segments perpendicular to it. Each perpendicular line has its own upper-limit distance. We classify regions near perpendicular segments within their upper-limit distances multiplied by a user-defined constant

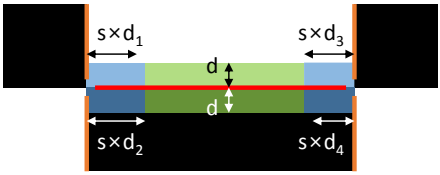


Fig. 5. Line features exist only near lines, not homogeneous regions or near corners. For each line segment (a red line with upper-limit distance of d), there are four neighboring line segments (orange lines with upper-limit distances of $d_1 \sim d_4$) perpendicular to it. Blue regions near perpendicular segments within their upper-limit distances multiplied by s are classified as 'corner regions'. The remaining region (green region) near the line segment within its upper-limit distance is classified as a 'line region'. We extract line features from only micro-lenses whose centers are in line regions of a sub-aperture image.

s as 'corner regions'. Since the upper-limit distances of neighboring perpendicular lines may be different ($d_1 \neq d_2$, $d_3 \neq d_4$ in Fig. 5), we choose a larger one between two distances. The region near each line segment within its upper-limit distance except corner regions is classified as a 'line region'. We extract line features from micro-lenses whose centers are in line regions of the sub-aperture image.

3 Projection Model of Micro-Lens-Based Light-Field Cameras

Micro-lens-based light-field cameras contains two layers of lenses: the main lens and a micro-lens array. We apply the 'thin lens model' to the main lens and the 'pinhole model' to the micro-lenses, similar to [5].

Let F be the focal length of a thin lens. All rays from an arbitrary point with distance a from a lens pass through it and head to a common point called an 'image (see Fig. 6). That is why we may consider the location of an image as that of a point. The word 'image' refers to 'a common point of rays' only in this section while it refers to 'data captured by a camera' in the other sections.

The image, point, and lens center are collinear because a ray passing through them (red line in Fig. 6) is not refracted by the lens. If the lens center is set as the origin, the coordinates of the image can therefore be computed using those of the point and a distance ratio b/a. The distance b of the image from the lens is computed using a and F as follows:

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{F} \tag{4}$$

$$b = \frac{aF}{a - F} = \frac{F}{a - F} \cdot a. \tag{5}$$

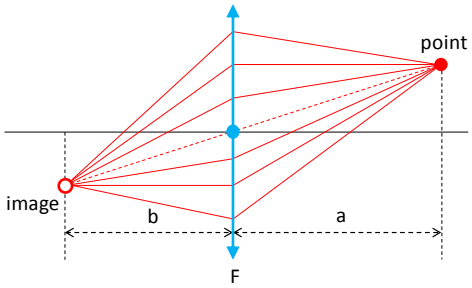


Fig. 6. Thin lens model. All rays from an arbitrary point pass through a common point called an 'image'. Observing a point through a lens, we may treat the location of an image as that of the point. If the lens center is set as the origin, the coordinates of the image are computed using those of the point and a distance ratio b/a because a ray (dotted red line) passing through them and the lens center is not refracted by the lens.

Figure 7 shows the projection model of micro-lens-based light-field cameras. The projected location of an arbitrary point is the intersection of the CCD array and a ray that passes through an image of the point and a micro-lens center (see Fig. 7(a)). The actual path of the ray is indicated by a red line, but we do not have to be concerned with this because we consider the location of the image as that of the point.

Since there are many micro-lenses, a point is projected onto multiple locations (one location for each micro-lens). In Fig. 7(b), the left part of the main lens in Fig. 7(a) is rotated 180 degrees. Without loss of generality, we set the center and optical axis of the main lens as the origin and the z-axis of the camera coordinate system, respectively. The relation between the coordinates of point (X_c, Y_c, Z_c) and image (X, Y, Z) is described as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{F}{Z_c - F} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}. \tag{6}$$

The ratio among elements is not changed because both of them lie on a line that passes through the origin of the camera coordinate system.

Let L and l be the distances from the micro-lens array to the main lens and the CCD array, respectively. The physical center of the micro-lens is computed using its projected location (x_c, y_c):

$$(\text{physical center of micro lens}) = L \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}. \tag{7}$$

Note that projected locations are expressed in a normalized coordinate system $(z = 1)$.

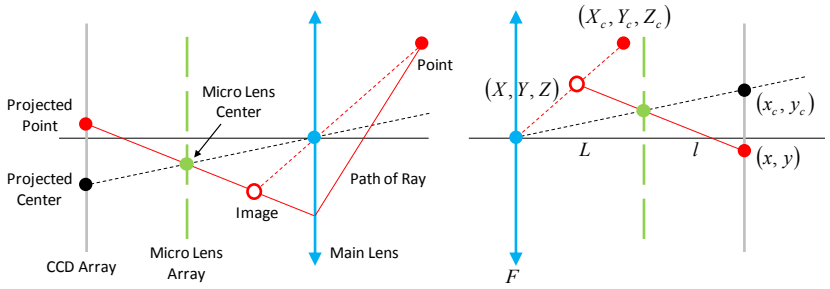


Fig. 7. Projection model of micro-lens-based light-field cameras. (a) All rays from an arbitrary point pass through the image of the point. For each micro-lens, a ray from the image passes through its center and meets with the CCD array. (b) The left part of the main lens is rotated 180 degrees to make further computation easier, similar to the conventional pinhole model.

The projected location (x, y) of an image is computed by extending the line connecting the image and the micro-lens center as follows:

$$(L+l)\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \frac{L+l-Z}{L-Z}\left(L\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) \tag{8}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\frac{l}{(L-Z)(L+l)}\begin{bmatrix} X \\ Y \end{bmatrix} + \frac{L(L+l-Z)}{(L-Z)(L+l)}\begin{bmatrix} x_c \\ y_c \end{bmatrix}. \tag{9}$$

Equation (9) is simplified by subtracting the projected micro-lens center (x_c, y_c) from it:

$$\begin{bmatrix} x-x_c \\ y-y_c \end{bmatrix} = \frac{l}{(L-Z)(L+l)}\left(-\begin{bmatrix} X \\ Y \end{bmatrix} + Z\begin{bmatrix} x_c \\ y_c \end{bmatrix}\right). \tag{10}$$

Substituting (6) into (10),

$$\begin{aligned} \begin{bmatrix} x-x_c \\ y-y_c \end{bmatrix} &= \frac{l}{(L-\frac{F}{Z_c-F}Z_c)(L+l)}\left(-\frac{F}{Z_c-F}\begin{bmatrix} X_c \\ Y_c \end{bmatrix} + \frac{F}{Z_c-F}Z_c\begin{bmatrix} x_c \\ y_c \end{bmatrix}\right) \\ &= \frac{1}{K_2(K_1Z_c-1)}\left(-\begin{bmatrix} X_c \\ Y_c \end{bmatrix} + Z_c\begin{bmatrix} x_c \\ y_c \end{bmatrix}\right), \end{aligned} \tag{11}$$

where $K_1 \equiv 1/F - 1/L$ and $K_2 \equiv L(L/l+1)$.

4 Calibration of Micro-Lens-Based Light-Field Cameras

The projection model of (11) contains normalized coordinates of projected points. They can be computed if and only if we know the transformation between the

normalized coordinates and the image coordinates. In the case of conventional pinhole cameras, the transformation is defined by a 3×3 matrix called an intrinsic parameter. We simplify it by assuming zero skew, single focal length, and zero center (coordinate of image center is set to $(0, 0)$). The image coordinates (u, v) are then computed simply by scaling the normalized coordinates(x, y):

$$\begin{bmatrix} u \\ v \end{bmatrix} = f \begin{bmatrix} x \\ y \end{bmatrix}, \tag{12}$$

where f is the focal length of intrinsic parameters. Actually f indicates the number of pixels in one measurement unit (millimeter in this paper). Adopting (12) to the projection model (11),

$$\begin{bmatrix} u - u_c \\ v - v_c \end{bmatrix} = \frac{1}{K_2(K_1 Z_c - 1)} \left(-f \begin{bmatrix} X_c \\ Y_c \end{bmatrix} + Z_c \begin{bmatrix} u_c \\ v_c \end{bmatrix} \right). \tag{13}$$

The simplified model (12) will be restored to a 3×3 upper triangular matrix in Sect. 5.

We apply (13) to line features extracted from raw images of a checkerboard pattern. Unfortunately, projections of two adjacent corners do not lie exactly on their corresponding line feature because of nonlinear terms such as the left multiplier term $1/K_2(K_1 Z_c - 1)$ of (13) and radial distortion of the main lens. We do not know which part of a line segment is projected onto micro-lens images and extracted as line features. However, projections of corners are close enough to corresponding line features to use an approximation that they lie on the features. Let $(a,\ b,\ c)$ be the parameters of a line feature:

$$a(u - u_c) + b(v - v_c) + c = 0 \qquad (a^2 + b^2 = 1). \tag{14}$$

Substituting corner projections (13), the line equation (14) becomes

$$a(-fX_c + Z_c u_c) + b(-fY_c + Z_c v_c) + cK_2(K_1 Z_c - 1) = 0. \tag{15}$$

Let (X_w, Y_w, Z_w) be one of two corners that define a line segment in the world coordinate system (i.e., the checkerboard coordinate system). It must be transformed into the camera coordinate system by an unknown transformation matrix with a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation vector $\mathbf{t}$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t} = \begin{bmatrix} r_{11} X_w + r_{12} Y_w + t_1 \\ r_{21} X_w + r_{22} Y_w + t_2 \\ r_{31} X_w + r_{32} Y_w + t_3 \end{bmatrix}, \tag{16}$$

where r_{ij} and t_i are the elements of $\mathbf{R}$ and $\mathbf{t}$ at the i-th row and the j-th column, respectively. Without loss of generality, the z-coordinate of the checkerboard

pattern is set to zero ($Z_w = 0$ for all corners). Substituting (X_c, Y_c, Z_c) by (16), (15) becomes an $\mathbf{Ax} = \mathbf{0}$ form:

$$\begin{bmatrix} -aX_w \\ -aY_w \\ -a \\ -bX_w \\ -bY_w \\ -b \\ (au_c + bv_c)X_w \\ (au_c + bv_c)Y_w \\ (au_c + bv_c) \\ cX_w \\ cY_w \\ c \end{bmatrix}^\top \begin{bmatrix} fr_{11} \\ fr_{12} \\ ft_1 \\ fr_{21} \\ fr_{22} \\ ft_2 \\ r_{31} \\ r_{32} \\ t_3 \\ K_1K_2r_{31} \\ K_1K_2r_{32} \\ K_2(K_1t_3 - 1) \end{bmatrix} = 0. \tag{17}$$

For each micro-lens whose center is in line regions shown in Fig. 5, two equations are derived from two corners that define a line segment corresponding to a line feature in the lens. Stacking all line features to the matrix $\mathbf{A}$, its right singular vector $\mathbf{v}$ corresponding to its smallest singular value is selected as an initial solution multiplied by an unknown scale λ ($\mathbf{v} = \lambda\mathbf{x}$). Let v_n be the n-th element of $\mathbf{v}$. Initial values of unknown parameters are computed using $\mathbf{v}$ as follows:

$$f = \sqrt{\frac{-v_1v_2 - v_4v_5}{v_7v_8}} \quad (\because r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} = 0), \tag{18}$$

$$\lambda = \sqrt{(v_1/f)^2 + (v_4/f)^2 + v_7^2} \quad (\because r_{11}^2 + r_{21}^2 + r_{31}^2 = 1), \tag{19}$$

$$\mathbf{r}_1 = \frac{1}{\lambda f}\begin{bmatrix} v_1 & v_4 & fv_7 \end{bmatrix}^\top, \quad \mathbf{r}_3 = \frac{\mathbf{r}_1 \times \begin{bmatrix} v_2 & v_5 & fv_8 \end{bmatrix}^\top}{\left\| \mathbf{r}_1 \times \begin{bmatrix} v_2 & v_5 & fv_8 \end{bmatrix}^\top \right\|}, \quad \mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1, \tag{20}$$

$$\mathbf{t} = \frac{1}{\lambda f}\begin{bmatrix} v_3 & v_6 & fv_9 \end{bmatrix}^\top, \tag{21}$$

$$K_2 = \frac{1}{\lambda}\left(\frac{(v_{10} + v_{11})t_3}{r_{31} + r_{32}} - v_{12}\right), \quad K_1 = \frac{v_{10} + v_{11}}{\lambda K_2(r_{31} + r_{32})}, \tag{22}$$

where $\mathbf{r}_n$ is the n-th column of $\mathbf{R}$. The scale λ is always positive in (19); however, it can be negative in real cases. We determine its sign by checking that of the third term of $\mathbf{t}$ (v_9/λ) because it must be positive (i.e., the planar pattern must be in front of the camera). If it is negative, we change the signs of $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{t}$, K_1 and K_2.

5 Non-linear Optimization

The initial solution computed in Sect. 4 is refined via a non-linear optimization. We have mentioned that checkerboard corners do not lie exactly on corresponding line features. Let (u, v) and (u', v') be projections of two adjacent corners (X_w, Y_w) and (X'_w, Y'_w), respectively, onto a micro-lens image whose center is at (u_c, v_c). We compute a point on the line connecting (u, v) and (u', v') that is closest to (u_c, v_c) by computing a constant k'. In the optimization process, we project $(X_w, Y_w)+k'(X'_w-X_w, Y'_w-Y_w)$ onto the micro-lens image and compute the distance between the projected location and corresponding line feature as follows:

$$k' = \underset{k}{\operatorname{argmin}} \left\| \begin{bmatrix} u \\ v \end{bmatrix} + k \left(\begin{bmatrix} u' \\ v' \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \right) - \begin{bmatrix} u_c \\ v_c \end{bmatrix} \right\|^2 . \tag{23}$$

In Sect. 4, we simplified the transformation between normalized coordinates and image coordinates as (12) and (13). However, the simplified transformation is not suitable for real cameras. We restore it to a generalized model with zero skew as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} u - u_c \\ v - v_c \end{bmatrix} = \frac{1}{K_2(K_1 Z_c - 1)} \left(- \begin{bmatrix} f_x X_c \\ f_y Y_c \end{bmatrix} + Z_c \begin{bmatrix} u_c - c_x \\ v_c - c_y \end{bmatrix} \right) . \tag{25}$$

Initial values of f_x and f_y are set to that of f, and those of c_x and c_y are set to the coordinates of the center pixel of raw images.

Moreover, we also consider radial distortion of the main lens. An arbitrary point (X_c, Y_c, Z_c) in the camera coordinate system is distorted using a popular model,

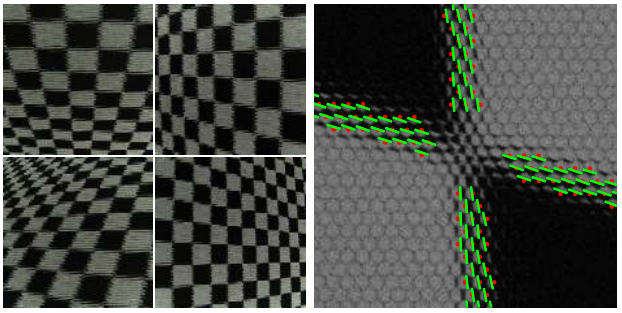
$$\begin{bmatrix} \hat{X}_c \\ \hat{Y}_c \\ \hat{Z}_c \end{bmatrix} = \begin{bmatrix} (1 + k_1 r^2 + k_2 r^4) X_c \\ (1 + k_1 r^2 + k_2 r^4) Y_c \\ Z_c \end{bmatrix} \qquad \left(r^2 = (X_c/Z_c)^2 + (Y_c/Z_c)^2 \right) , \tag{26}$$

mentioned in Zhang's work on camera calibration [18]. The camera coordinate (X_c, Y_c, Z_c) in (25) is substituted by $(\hat{X}_c, \hat{Y}_c, \hat{Z}_c)$.

The final version of the cost function of the non-linear optimization is

$$\begin{aligned} &g(K_1, K_2, \mathbf{R}, \mathbf{t}, f_x, f_y, c_x, c_y, k_1, k_2) \\ &= \sum \left\| a \cdot (u + k'(u' - u) - u_c) + b \cdot (v + k'(v' - v) - v_c) + c \right\|^2 , \end{aligned} \tag{27}$$

which is a combination of the equations mentioned above. First we transform adjacent corners of a checkerboard into the camera coordinate system using (16). They are distorted by (26) and then projected onto micro-lens images using (25). The cost function g is a squared sum of distances between the closest points computed by (23) and corresponding line features (14). Note that just one distance is computed for each line feature, instead of two.



(a) (b)

Fig. 8. (a) A part of raw images used for calibration. (b) Example of line features extracted from raw images. Red dots and green lines indicate micro-lens centers and line features, respectively. No line features are extracted from corner regions.

The proposed algorithm can be applied to a single raw image of a checkerboard. In the case of using multiple frames, we compute an average value of f first and compute initial values of the other parameters using it. Since a small number of frames may provide an imaginary value for f due to noise and mathematical error, we ignore those values and compute the average of only real values. Values of K_1 and K_2 are also averaged to make the final solution converge faster.

6 Experimental Results

We captured a dataset of checkerboard images using a Lytro camera. The size of raw images are 3280×3280 pixels, and the grid size of a checkerboard is 10 mm. Line parameters (θ, t) in Sect. 2.1 are selected among $\theta = 0 : 0.5 : 180$(deg, MATLAB expression) and $t = -4 : 0.05 : 4$(pixels) via a coarse-to-fine search. For the line fitting described in Fig. 4(d), we selected $N = 100$ nearest micro-lens centers from each line segment and discarded the centers whose feature distance is larger than 3 pixels. The upper limit of the segment distance in Sect. 2.2 is set to the value corresponding to the user-defined feature distance of 2 pixels. The value of s is set to 2.5 because the radius of micro-lens images is 5 pixels in the case of using Lytro. We extracted 155,474 line features from 9 raw images shown in Fig. 8(a). Examples of line features extracted from raw images are displayed in Fig. 8(b). Since we did not extract corners, we projected them onto corner regions of raw images to verify the accuracy of the results. Figure 9 shows an example of a corner projected onto a raw image. Our dataset and executables are available online[1].

After calibrating the camera using raw images, we generated a number of sub-aperture images based on the calibration result. Since the average distance between adjacent micro-lens centers is near 10 pixels, the size of the sub-aperture images is set to 328×328 (1/10 of raw images). We extracted corner features from

[1] `https://sites.google.com/site/yunsubok/lf_geo_calib`

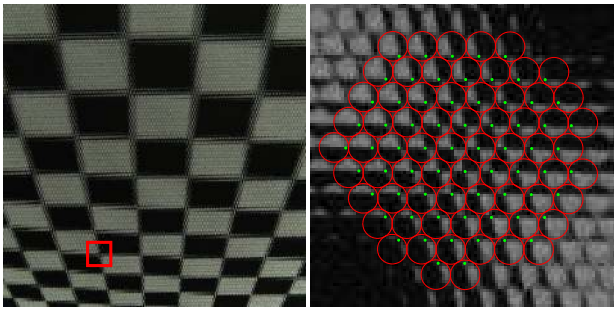


Fig. 9. Corners are projected onto raw images to verify the accuracy of line features extracted from raw images. The right image is a magnified view of the region denoted by a red rectangle in the left image. Red circles and green dots indicate micro-lenses and projected corners, respectively. The projected location in each micro-lens image is very close to its actual location.

Table 1. Projection error of sub-aperture images (unit: pixel)

Distance from center	-3	-2	-1	0	1	2	3
-3	0.9944	0.5913	0.5301	0.5152	0.5253	0.5766	0.7563
-2	0.5587	0.4634	0.4508	0.4451	0.4487	0.4578	0.5357
-1	0.4795	0.4308	0.4163	0.3759	0.3917	0.4285	0.4625
0	0.4451	0.4127	0.3859	0.3368	0.3526	0.4036	0.4253
1	0.4332	0.3944	0.3937	0.3589	0.3675	0.3885	0.4137
2	0.4344	0.3825	0.3615	0.3623	0.3684	0.3830	0.4229
3	0.5331	0.4055	0.3945	0.3885	0.3896	0.3968	0.5079

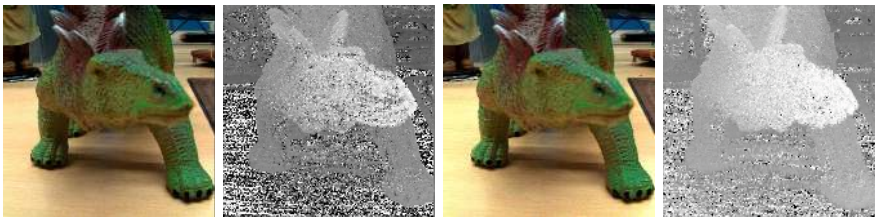
sub-aperture images independently and computed the RMS value of projection errors. Extrinsic parameters of the calibration result are the transformation from the checkerboard coordinate system to the camera coordinate system of the raw images. Transformation from the camera coordinate system to each sub-aperture coordinate system is computed using the calibration result. Projection errors of sub-aperture images are summarized in Table 1. Although extrinsic parameters of sub-aperture images are not estimated independently but 'predicted' using the calibration result, the projection errors are very small. Details of generating sub-aperture images are described in our supplementary material.

For direct comparison of the proposed method with [5], we also measured ray re-projection errors in millimeters using the same datasets[2]. The results are shown in Table 2. Although our RMS ray re-projection errors are almost the same as those of [5], our method has two advantages. The first is that we have obtained similar results using a smaller number of parameters (6 parameters) than [5] (12 parameters). This is verified by reducing the number of images used for calibration, as shown in Table 2. In addition, our sub-aperture images are geometrically closer to images captured by a parallel multi-camera array than those of [5].

[2] `http://marine.acfr.usyd.edu.au/research/plenoptic-imaging`

Table 2. RMS ray re-projection error of sub-aperture images (unit: mm). The numbers (N) of the proposed method indicate the number of images used for calibration among 18 images for each dataset.

Method	Previous [5]	Proposed	Proposed
Dataset B (grid 3.61)	0.0628	(9) 0.0717	(5) 0.0700
Dataset E (grid 35.1)	0.363	(14) 0.365	(5) 0.331



(a) (b) (c) (d)

Fig. 10. (a) Sub-aperture image using the method in [5]. (b) Disparity map estimated using (a). (c) Sub-aperture image using the proposed method. (d) Disparity map estimated using (c).

High accuracy of geometric calibration is essential for good results in various applications. In order to verify the geometric accuracy of the proposed method, we estimated disparity maps using a state-of-the-art disparity labeling method [16] as an example application. It uses a structure tensor to find correspondences between sub-aperture images and it provides accurate results if corresponding points in sub-aperture images are aligned well. We fixed user-defined parameters for a fair comparison. Sub-aperture images from [5] in Fig. 10(a) lead to outliers while those from the proposed method in Fig. 10(c) provide reliable results.

7 Conclusion

We presented a novel method of geometric calibration of micro-lens-based light-field cameras. Instead of using sub-aperture images, line features are extracted from raw images directly. We formulated a projection model based on the thin-lens model and the pinhole model and applied it to line features. An initial solution of both intrinsic and extrinsic parameters are estimated using a linear computation, and it is refined via a non-linear optimization. Since the correspondences between rays and pixels in raw images are estimated accurately using the proposed method, geometrically well-aligned sub-aperture images provide small projection errors and a disparity map with less noise. Future work will include improving the accuracy of line features, overcoming limitations of thin-lens and pinhole models, and modeling the lens distortion of outer sub-aperture images.

Acknowledgement. This research is supported by the Study on Imaging Systems for the next generation cameras funded by the Samsung Electronics Co., Ltd (DMC R&D center) (IO130806-00717-02).

References

1. Adelson, E.H., Wang, J.Y.A.: Single lens stereo with a plenoptic camera. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 99–106 (1992)
2. Birklbauer, C., Opelt, S., Bimber, O.: Rendering gigaray light fields. Computer Graphics Forum 32(2), 469–478 (2013)
3. Bishop, T.E., Zanetti, S., Favaro, P.: Light field superresolution. In: International Conference on Computational Photography, pp. 1–9 (2009)
4. Dansereau, D.G., Mahon, I., Pizarro, O., Williams, S.B.: Plenoptic flow: Closed-form visual odometry for light field cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4455–4462 (2011)
5. Dansereau, D.G., Pizarro, O., Williams, S.B.: Decoding, calibration and rectification for lenselet-based plenoptic cameras. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1027–1034 (2013)
6. Dong, F., Ieng, S.H., Savatier, X., Etienne-Cummings, R., Benosman, R.: Plenoptic cameras in real-time robotics. The International Journal of Robotics Research 32(2), 206–217 (2013)
7. Georgiev, T., Lumsdaine, A.: Reducing plenoptic camera artifacts. Computer Graphics Forum 29(6), 1955–1968 (2010)
8. Johannsen, O., Heinze, C., Goldluecke, B., Perwaß, C.: On the calibration of focused plenoptic cameras. In: Grzegorzek, M., Theobalt, C., Koch, R., Kolb, A. (eds.) Time-of-Flight and Depth Imaging. LNCS, vol. 8200, pp. 302–317. Springer, Heidelberg (2013)
9. Liang, C.K., Lin, T.H., Wong, B.Y., Liu, C., Chen, H.H.: Programmable aperture photography: multiplexed light field acquisition. ACM Transactions on Graphics 27(3), Article 55 (2008)
10. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Stanford University Computer Science Technical Report CSTR 2(11) (2005)
11. Perez Nava, F., Luke, J.: Simultaneous estimation of super-resolved depth and all-in-focus images from a plenoptic camera. In: 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009, pp. 1–4 (2009)
12. Taguchi, Y., Agrawal, A., Veeraraghavan, A., Ramalingam, S., Raskar, R.: Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. ACM Transactions on Graphics 29(6), Article 172 (2010)
13. Tao, M.W., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: IEEE International Conference on Computer Vision, pp. 673–680 (2013)
14. Vaish, V., Wilburn, B., Joshi, N., Levoy, M.: Using plane+parallax for calibrating dense camera arrays. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. I–2–I–9 (2004)
15. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Transactions on Graphics 26(3), Article 69 (2007)
16. Wanner, S., Goldluecke, B.: Globally consistent depth labeling of 4d lightfields. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 41–48 (2012)
17. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. ACM Transactions on Graphics 24(3), 765–776 (2005)
18. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: IEEE International Conference on Computer Vision, pp. 666–673 (1999)