Correction and rectification of light fields

Ke Deng[a],[*], Lifeng Wang[b], Zhouchen Lin[b], Tao Feng[b], Zhidong Deng[a]

[a] *Computer Science Department, Tsinghua University, Beijing 100084, China*
[b] *Microsoft Research Asia, No. 49 Zhichun Road, Beijing 100080, China*

Abstract

The light field is a well-known image-based rendering technology. The traditional way to capture a light field is to move a camera on a plane and take images at every gird points. However, due to device defects it is hard to ensure that the captured light field is ideal. For example, the camera's imaging plane is not on the image plane of the light field and the camera is not in the right position. To solve this problem we propose a correction and rectification framework that uses only four images. This framework involves un-distortion, feature point detection, homography computation to correct the orientation of the camera, and positional error correction. It is the first to take positional error into consideration. Our experiments show that our method is effective.
© 2003 Elsevier Science Ltd. All rights reserved.

1. Introduction

Image-based rendering techniques use a collection of sample images to render novel views. It is regarded as a powerful alternative to traditional geometry-based techniques for image synthesis [1,2]. The light field [3] is a well-known image-based rendering technique. It uses a two-plane parameterization to index the light rays in regions of space free of occluders. Every light ray can be defined by connecting a point on the camera plane to another point on the image plane (Fig. 1(a)). With a light field, new views can be easily synthesized by interpolating among appropriate rays. A light field is usually captured by moving a camera on a camera plane and taking pictures at every grid points of the camera plane (Fig. 1(b)). However, in real cases due to device defects the captured light field usually not ideal. For example, the camera is not moving on a plane, or the

camera's optical axis is not perpendicular to the camera plane, or the camera is not in the right position precisely (Fig. 1(c)). Moreover, lens distortion can also make the ray query incorrect. On the other hand, correcting the light field and rectifying it, i.e., making the epi-polar lines of the images parallel to the coordinate axes in the image space, facilitate further treatment on the light field, such as rendering and stereo matching among the images. Therefore, correction and rectification is necessary so that the light field is close to ideal.

1.1. Previous work

There has been some work on the rectification between stereo images [4–6], where the number of images is usually two or the images are on a line. In [5], the epipolar lines are aligned by estimating two projective parameters. The results of this algorithm have much shearing distortion. Similar distortion exists in [6] because it only aligns the epipolar lines and does not consider the visual effect of overall image. In [4], the estimation of shear transform is proposed and thus the results are much more satisfactory. It recovers the fundamental matrix first and then estimates the

---

*Corresponding author.

E-mail addresses: dengke99@mails.tsinghua.edu.cn (K. Deng), lfwang@microsoft.com (L. Wang), zhoulin@microsoft.com (Z. Lin), tfeng@microsoft.com (T. Feng), michael@cs.wustl.edu (Z. Deng).

rectifying homography (see Section 2.2.1), which relates the coordinates of a planar object in two views. But calculating the fundamental matrix is sensitive to correspondence accuracy, so this two-step algorithm causes error to accumulate.

Though there are commons between the rectification of light fields and that of stereo images, light field rectification needs more constraints and conventional methods used in stereo image rectification cannot be applied directly, because the light fields are taken by a camera moving in a plane. The requirement that the epipolar lines are aligned is not enough [4]. Therefore, more constraints must be imposed. There is little previous work on the problem of rectifying a light field. In [3], warping to a common plane is mentioned as camera pans and tilts, but there are few words about how to warp and the precision of the warping, which are critical for the application of light fields. The warping is a rectifying process. We find that the pose of the camera is usually fixed when moving, which means recovering a single rectifying homography is enough for a whole light field. In our algorithm, we adopt a novel method by decomposition and iteration that insures the globally optimal solution and also avoids error accumulation. It is also noteworthy that our framework is the first to take positional error into consideration.

For the computation of homographies and positional error correction, feature points are indispensable.

Therefore, it is important to design markers to facilitate the detection of feature points. In [7], markers consisting of concentric rings were used for tracking from a hand-held camera. Youngkwan Cho proposed a multi-ring color fiducial system for augmented reality in [8]. Concentric rings were also used in [9]. We also choose to use such markers for the detection of independent feature points. What is different is that we also use dots between the concentric-rings as dependent feature points. The main advantage of using dependent features is that the pattern of all markers is smaller so that it is visible in more views simultaneously.

1.2. Overview of our framework

Our framework consists of positional error correction and rectification of the light field, where the correction also depends on the homography computed in the rectification process. It involves the following steps (Fig. 2):

(1) un-distorting every image in the light field,
(2) detecting the independent and dependent feature points of each marker,
(3) calculating the intermediate homography that transforms the quadrangle of feature points into rectangle,
(4) computing the affine transform that scales the image transformed by the intermediate homography appropriately,
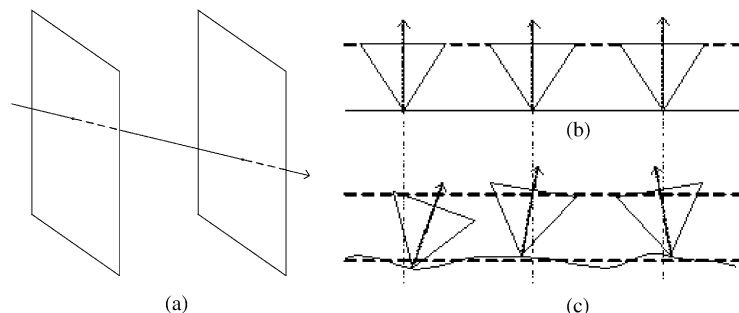

Fig. 1. The concept of light field and the correction and rectification problem. (a) The two-plane parameterization of the light field. (b) In the ideal light field, the imaging plane of the camera should keep on the image plane of the light field and the camera should be in the right position. (c) In practice, it is hard to make the imaging plane of the camera on a fixed plane and the camera may not be on the camera plane or at the grids precisely. In (b) and (c), the triangles represent the cameras and the dashed arrows are the optical axes of the cameras. The dashed lines in (c) illustrate the correct positions of the camera.
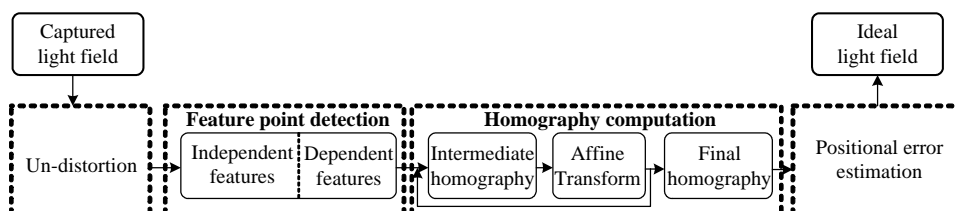

Fig. 2. The outline of our framework.

(5) iterating steps 3 and 4 to refine the homography,

(6) combining the iterative results to compute the final homography and rectifying the whole light field, and

(7) correcting the positional error after estimating the error.

The rest of this paper is organized as follows. In Section 2 we present the details of our framework, including feature design and detection, the computation of the homography, and the correction of positional error. Finally we show our experiments in Section 3 and conclude in Section 4.

2. Our framework

2.1. Feature design and detection

Our framework is based on positions of several feature points in different views. To simplify the process of feature detection and matching, a fiducial system is used. The fiducial system consists of detecting two kinds of features: independent ones and dependent ones. The positions of both kinds of features can be detected independently and the features are printed on a planar board. However, the identity of an independent feature can also be detected independently, while that of a dependent feature cannot until its neighbors are recognized.

2.1.1. Independent features

We choose concentric rings as markers since they can be detected accurately and stably. A typical marker and its image under projective projection are shown in Figs. 3(a) and (b). The centers of the markers are the independent features. The detection for such feature points consists of the following steps:

(1) All elliptic contours are recognized and fit by ellipses (Fig. 3(c)) using the Open Source Computer Vision Library by Intel Corporation [10].

(2) Since the centers of ellipses and the projected center of concentric circles are collinear [11], by fitting this line, the relative radii of the circles that correspond to the smaller ellipses can be computed from the preservation of cross ratio (Fig. 3(d)). These relative radii are used as features to identify which marker it is.

2.1.2. Dependent features

As shown in the above figures, an independent feature should be relatively large in a single view. Otherwise its details will merge together. Moreover it needs to be carefully designed. Otherwise there may be similar
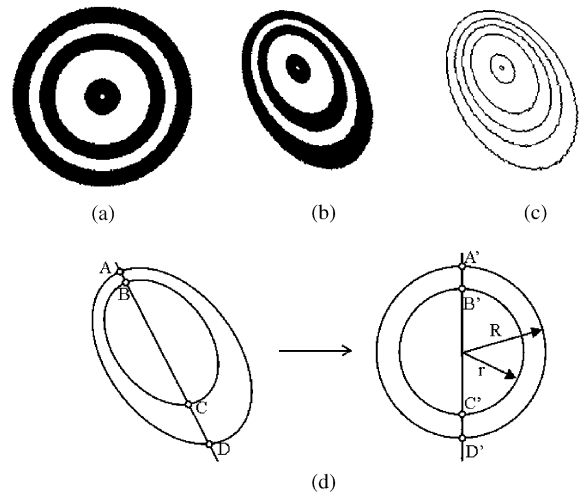

Fig. 3. The detection of independent feature points. (a) The original multi-ring marker. (b) The marker after a projective transform. (c) The contours are detected and be fit by ellipses. (d) The relative radii r/R of smaller circles can be computed by exploiting the preservation of cross ratio, where ellipse AD is the outmost ellipse.

markers that can cause trouble in recognition. Therefore, we cannot use too many independent features. In most cases, the number of independent features in a single view is less than 9. So we make use of dependent features that exploit the relation between the features.

The position of each feature in the original pattern is known. After a projective transform, if three collinear features are visible in the image, the positions of other features on the same line can be computed using the cross ratio. If four visible features form a quadrangle, the positions of all features on the plane can be estimated from the homography. Then the positions of previously found markers are compared with these estimated positions. If the distance between the two is within a threshold (typically 1–2 pixels), these markers are identified. In our system, we use the pattern in Fig. 4, which has 9 independent features (concentric-rings) and 16 dependent features (dots).

2.1.3. Error analysis

Our feature detection algorithm is tested with different poses of the camera. To obtain correct feature positions as the ground truth for error analysis, we use a virtual camera so that both intrinsic and extrinsic parameters are known. The camera is located on a hemisphere, with the optical axis pointing to the center of the hemisphere (Fig. 5(a)). The pattern is placed on the center of the base of the hemisphere and images are captured as camera moves on the hemisphere
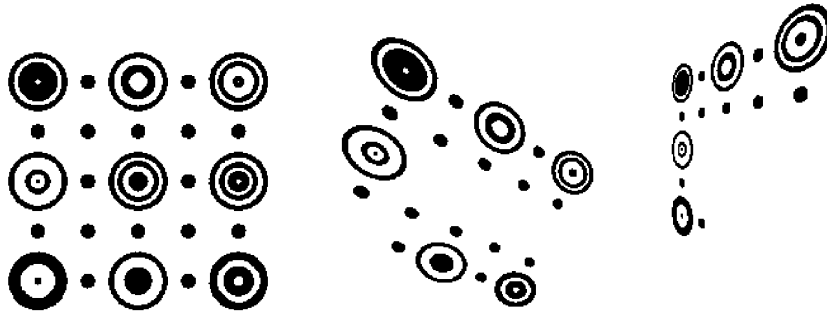
Fig. 4. The pattern with independent features and dependent features. On the left is the original pattern. All features in this pattern are numbered 0–24, from left to right, top to bottom. On the center is the pattern after a perspective transform. It is partially visible due to occlusion. Because more than four non-degenerate independent features are visible, all positions of visible features can be determined. On the right is the pattern after another transform. The features 0, 2 and 4 are visible independent features. Then features 1 and 3 can be found using the cross ratio. Similarly is feature 5 and 15. But the positions of features 6, 7, 8, 9 and 21 cannot be determined.

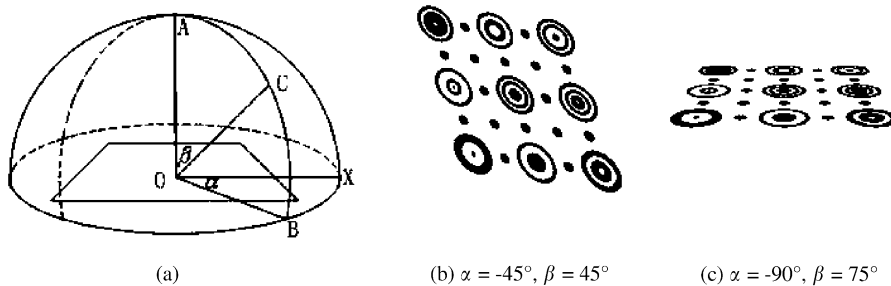   (a)   (b) α = -45°, β = 45°   (c) α = -90°, β = 75°

Fig. 5. The synthetic experiment to test our feature detection algorithm. (a) The virtual camera translates on the hemisphere with its optical axis pointing to the center of the hemisphere. (b) and (c) Patterns taken at different positions of the camera. The definitions of α and β can be found in (a). (b) $\alpha = -45°$, $\beta = -45°$; (c) $\alpha = -90°$, $\beta = -75°$.

(Figs. 5(b)) and (c)). Fig. 6 shows the relation of the errors in pixel with the position of the camera. We can see that our feature detect algorithm is robust and accurate with the maximal error less than 1 pixel and the average error less than 0.2 pixels.

2.2. Homography computation

2.2.1. The homography

A planar object in two views is related by a perspective transform. The homography determines the relation between the coordinates of the planar object in two views by a matrix:

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}, \tag{1}$$

where homogeneous coordinates are used for 2D points in the image space:

$$X = \begin{pmatrix} x & y & w \end{pmatrix}^{\mathrm{T}}.$$

For $\lambda \neq 0$, $\begin{pmatrix} x & y & w \end{pmatrix}^{\mathrm{T}}$ and $\begin{pmatrix} \lambda x & \lambda y & \lambda w \end{pmatrix}^{\mathrm{T}}$ represent the same point in the projective space (In the sequel, λ stands for an appropriate scaling parameter.). Therefore, for $w \neq 0$, $\begin{pmatrix} x & y & w \end{pmatrix}^{\mathrm{T}}$ is the ordinary point $\begin{pmatrix} x/w & y/w \end{pmatrix}^{\mathrm{T}}$. As a result, only 8 parameters are independent in (1) and we may set $h_9 = 1$. Then the transform of X under the homography H can be written as

$$X' = HX.$$

To determine the homography, only four pairs of corresponding points with non-degenerate configuration (i.e., they form a convex quadrangle) are enough. Due to the fact that the camera's pose is usually fixed when moving, a single homography is enough to rectify all images. So we capture four images of the markers using our light field capturing device (Fig. 7(a)) to estimate this homography. All images are undistorted using the algorithm in [12] before processing by taking images of the un-distortion pattern shown in Fig. 7(b).

Let us investigate one feature point captured at different camera positions. The feature point appears
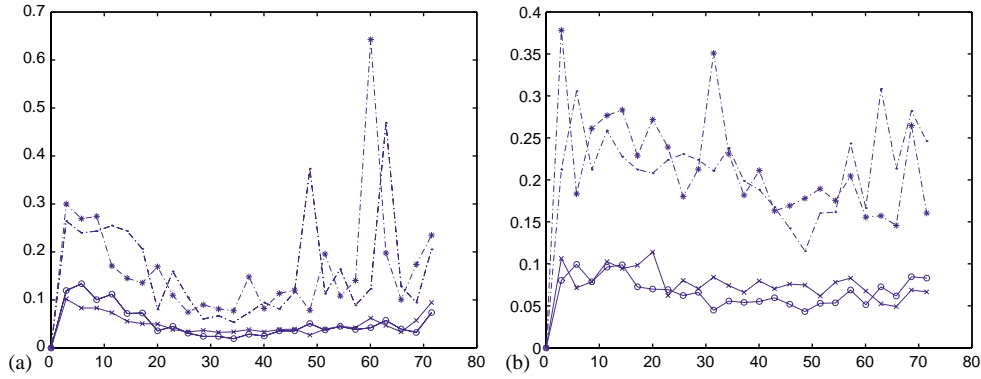
Fig. 6. The relation between the errors (in pixels) and the position of the camera. Curves with circles denote the average errors of all detected feature points in x. Those with crosses are the average errors in y. Dashed ones with stars are the maximal errors of all detected features in x, and with dots are the maximal error in y. All 25 feature points are detected in every image tested here. (a) $\alpha = -45°$; (b) $\alpha = -90°$.
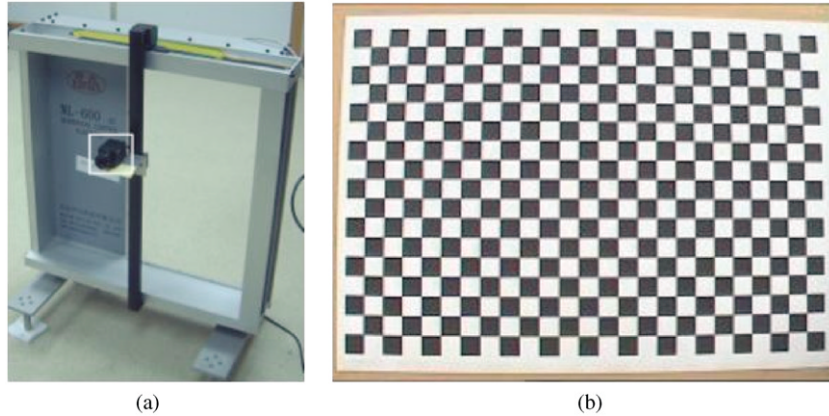

Fig. 7. (a) The light field capturing device. It is a vertical XY-table. The object inside the white box is a CCD camera. (b) The pattern used for un-distortion. The images are undistorted using this chessboard image before rectification.

in the upper-left, upper-right, bottom-left and bottom-right images as pixels P_0, P_1, P_2, P_3, respectively. And the coordinates of P_i are $\begin{pmatrix} x_i & y_i & 1 \end{pmatrix}^T (i = 0, \cdots, 3)$, respectively. Then the homography H must satisfy:

$$H \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} \quad (i = 0, \cdots, 3), \tag{2}$$

where $\begin{pmatrix} x'_i & y'_i & 1 \end{pmatrix}^T$ are the rectified coordinates, and

$$x'_0 = x'_2, \quad x'_1 = x'_3, \quad y'_0 = y'_1, \quad y'_2 = y'_3. \tag{3}$$

A direct computation of H will lead to a system of non-linear equations like $(h_1 x_0 + h_2 y_0 + h_3)/(h_7 x_0 + h_8 y_0 + h_9) = (h_1 x_2 + h_2 y_2 + h_3)(h_7 x_2 + h_8 y_2 + h_9)$, by eliminating x'_i's and y'_i's in (2) using relation (3). Such

method is computationally intensive and inaccurate. So we choose an indirect way as in [4]. On the other hand, the constraints in (3) are too weak to compute all the entries in H because they only require the rectified quadrangle is a rectangle. The size of the rectangle is not specified. So we add two constraints on H so that it keeps the image center invariant and avoids clipping. Consequently, the computation of rectifying homography consists of two parts:

(1) Computation of the intermediate homography $\bar{H}$ that maps the quadrangle $P_0 P_1 P_2 P_3$ into a rectangle.
(2) Computation of the affine transform A that keeps the image center invariant and avoids clipping.

For the computation of $\bar{H}$, we decompose it into an affine transform and a perspective transform as in [4]. Then the estimated homography is

$$H \approx A\bar{H}.$$

As estimating H only once may not be accurate enough, we may iterate the above steps to make it more accurate.

2.2.2. Estimation of $\bar{H}$

Since [4]:

$$\bar{H} = \begin{pmatrix} \bar{h} & \bar{h}_2 & \bar{h}_3 \\ \bar{h}_4 & \bar{h}_5 & \bar{h}_6 \\ \bar{h}_7 & \bar{h}_8 & \bar{h}_9 \end{pmatrix} = \tilde{A}\tilde{P}, \tag{4}$$

where

$$\tilde{A} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\equiv \begin{pmatrix} \bar{h}_1 - \bar{h}_3\bar{h}_7 & \bar{h}_2 - \bar{h}_3\bar{h}_8 & \bar{h}_3 \\ \bar{h}_4 - \bar{h}_6\bar{h}_7 & \bar{h}_5 - \bar{h}_6\bar{h}_8 & \bar{h}_6 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\tilde{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \bar{h}_7 & \bar{h}_8 & 1 \end{pmatrix}, \tag{5}$$

we may estimate the affine transform $\tilde{A}$ and the perspective transform $\tilde{P}$ separately.

Let $\tilde{A}$ satisfy (2) and (3), we get:

$$a_1 x_0 + a_2 y_0 + a_3 = a_1 x_2 + a_2 y_2 + a_3,$$
$$a_1 x_1 + a_2 y_1 + a_3 = a_1 x_3 + a_2 y_3 + a_3,$$
$$a_4 x_0 + a_5 y_0 + a_6 = a_4 x_1 + a_5 y_1 + a_6,$$
$$a_4 x_2 + a_5 y_2 + a_6 = a_4 x_3 + a_5 y_3 + a_6.$$

Hence, we can solve a_1/a_2 and a_4/a_5 by the least-square method. As we can see later, only the ratios are necessary. The two parameters a_3 and a_6, which are related to the translation of the image, are still not solved yet. However, they are not critical because the translation will be solved in another affine transform A.

On the other hand, binding (2)–(5), the two parameters $\bar{h}_7$ and $\bar{h}_8$ can be calculated from the following linear system:

$$(x_1 y_0 - x_0 y_1)\bar{h}_7 - (a_4/a_5)(x_1 y_0 - x_0 y_1)\bar{h}_8$$
$$= (a_4/a_5)(x_1 - x_0) + (y_1 - y_0),$$
$$(x_2 y_0 - x_0 y_2)\bar{h}_7 - (a_1/a_2)(x_2 y_0 - x_0 y_2)\bar{h}_8$$
$$= (x_2 - x_0) + (a_1/a_2)(y_2 - y_0),$$

$$(x_3 y_2 - x_2 y_3)\bar{h}_7 - (a_4/a_5)(x_3 y_2 - x_2 y_3)\bar{h}_8$$
$$= (a_4/a_5)(x_3 - x_2) + (y_3 - y_2),$$
$$(x_3 y_1 - x_1 y_3)\bar{h}_7 - (a_1/a_2)(x_3 y_1 - x_1 y_3)\bar{h}_8$$
$$= (x_3 - x_1) + (a_1/a_2)(y_3 - y_1).$$

2.2.3. Estimation of A

Till now, four constraints on the homography are obtained, namely, a_1/a_2, a_4/a_5, $\bar{h}_7$ and $\bar{h}_8$. Additional four constraints must be applied to solve all entries of H. There are many ways to impose constraints, as mentioned in [4]. Those we choose are to keep both the image center and the size of image invariant. Let

$$A = \begin{pmatrix} s_x & 0 & d_x \\ 0 & s_y & d_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{6}$$

where s_x and s_y account for the horizontal and vertical scaling, respectively, d_x and d_y account for the translation.

Suppose the image is of size $w \times h$ pixels, and on the image plane, the x-coordinate ranges from 0 to $w - 1$, and the y-coordinate ranges from 0 to $h - 1$, then the image center is $((w - 1)/2, (h - 1)/2)$. Let Q be the quadrangle of four corners of the original image, and $Q' = \bar{H}Q$. The constraint of keeping the image center invariant gives:

$$A\bar{H}\begin{pmatrix} (w - 1)/2 \\ (h - 1)/2 \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} (w - 1)/2 \\ (h - 1)/2 \\ 1 \end{pmatrix}. \tag{7}$$

To avoid clipping of the rectified image, let $(x, y, 1)^{\mathrm{T}}$ be the center of the image after applying the homography $\bar{H}$:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \bar{H}\begin{pmatrix} (w - 1)/2 \\ (h - 1)/2 \\ 1 \end{pmatrix}. \tag{8}$$

As the image center is invariant, s_x and s_y should be chosen as:

$$s_x = \frac{\max(x - \text{left}, \text{right} - x)}{w/2},$$

$$s_y = \frac{\max(y - \text{top}, \text{bottom} - y)}{h/2},$$

so that the scaled image does not beyond the area $[\text{left}, \text{right}] \times [\text{bottom}, \text{top}]$ on the rectified image, where left is the minimum of the x-coordinates of Q', right, top, and bottom are defined alike.

Combining (6)–(8), we get d_x and d_y from:

$$\begin{pmatrix} s_x & 0 & d_x \\ 0 & s_y & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix} = \begin{pmatrix} (w-1)/2 \\ (h-1)/2 \\ 1 \end{pmatrix}.$$

Finally, $H = A\tilde{A}\tilde{P}$ is computed.

2.2.4. Homography estimation by iteration

A single linear approximation of a non-linear problem is not accurate enough, but each estimated H can be used to transform all coordinates of feature points. Then we may estimate H again by the transformed coordinates. When the errors are below a preset threshold, the iteration stops and all homographies are concatenated together to get the final homography, i.e.,

$$H = H^{(n)} \cdots H^{(3)} H^{(2)} H^{(1)},$$

where $H^{(i)}$ is the homography computed in ith iteration.

Fig. 8 shows the reduction of errors with the iteration, where the errors are defined as follows:

$$e_x = \frac{1}{2N} \sum_{i=1}^{N} (|x_2^{(i)\prime} - x_0^{(i)\prime}| + |x_3^{(i)\prime} - x_1^{(i)\prime}|),$$

$$e_y = \frac{1}{2N} \sum_{i=1}^{N} (|y_1^{(i)\prime} - y_0^{(i)\prime}| + |y_3^{(i)\prime} - y_2^{(i)\prime}|)$$

in which N is the number of images in the light field and $\begin{pmatrix} x_j^{(i)\prime} & y_j^{(i)\prime} & 1 \end{pmatrix}^{\mathrm{T}}$ $(i = 0, \ldots, 3)$ are the transformed feature points after applying $H^{(i)}$.

2.3. Rectification and the correction of positional error

The markers shown in Fig. 3 can be put in the scene if the user does not care the presence of the markers or the object of interest is relatively small such that it does not
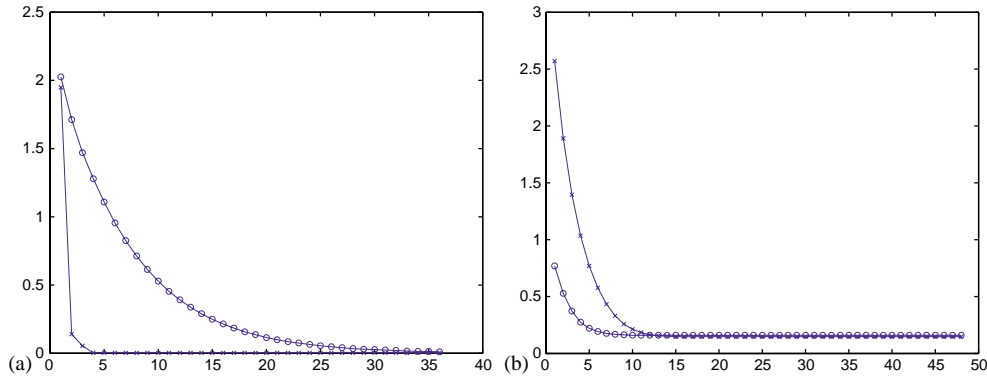

Fig. 8. Errors over iteration. The vertical axes are the error in pixel, and horizontal ones are the times of iteration. Curves with circles denote the errors in x-coordinate, while those with crosses denote the errors in y-coordinate. The left figure is from synthetic data, which are not acquired via feature detection in Section 2.1. The right figure is from real images after feature detection. Due to the error of feature detection, the errors do not converge to zero.
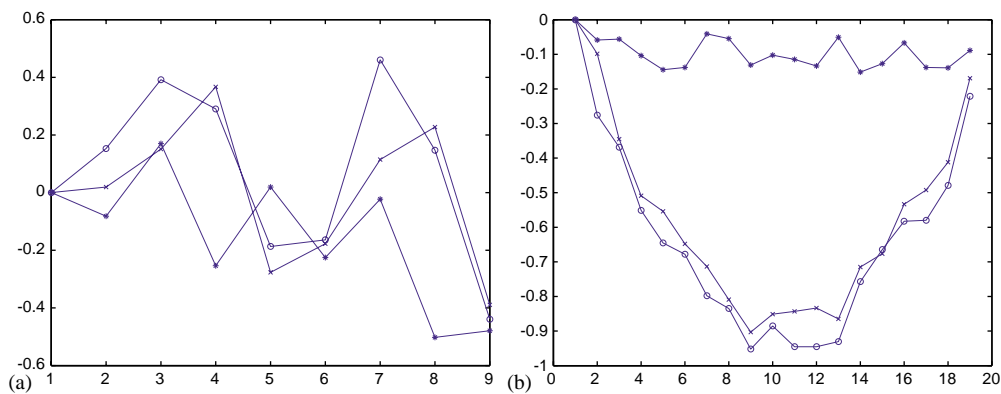

Fig. 9. Positional error correction. The vertical axes are the error in pixel, and the horizontal ones are the columns or rows of the image. Curves with circles denote the errors in the first light field. Those with crosses denote the errors in the second light field. Those with stars denote the errors after subtraction. (a) Average error along x-axis and (b) average error along y-axis.
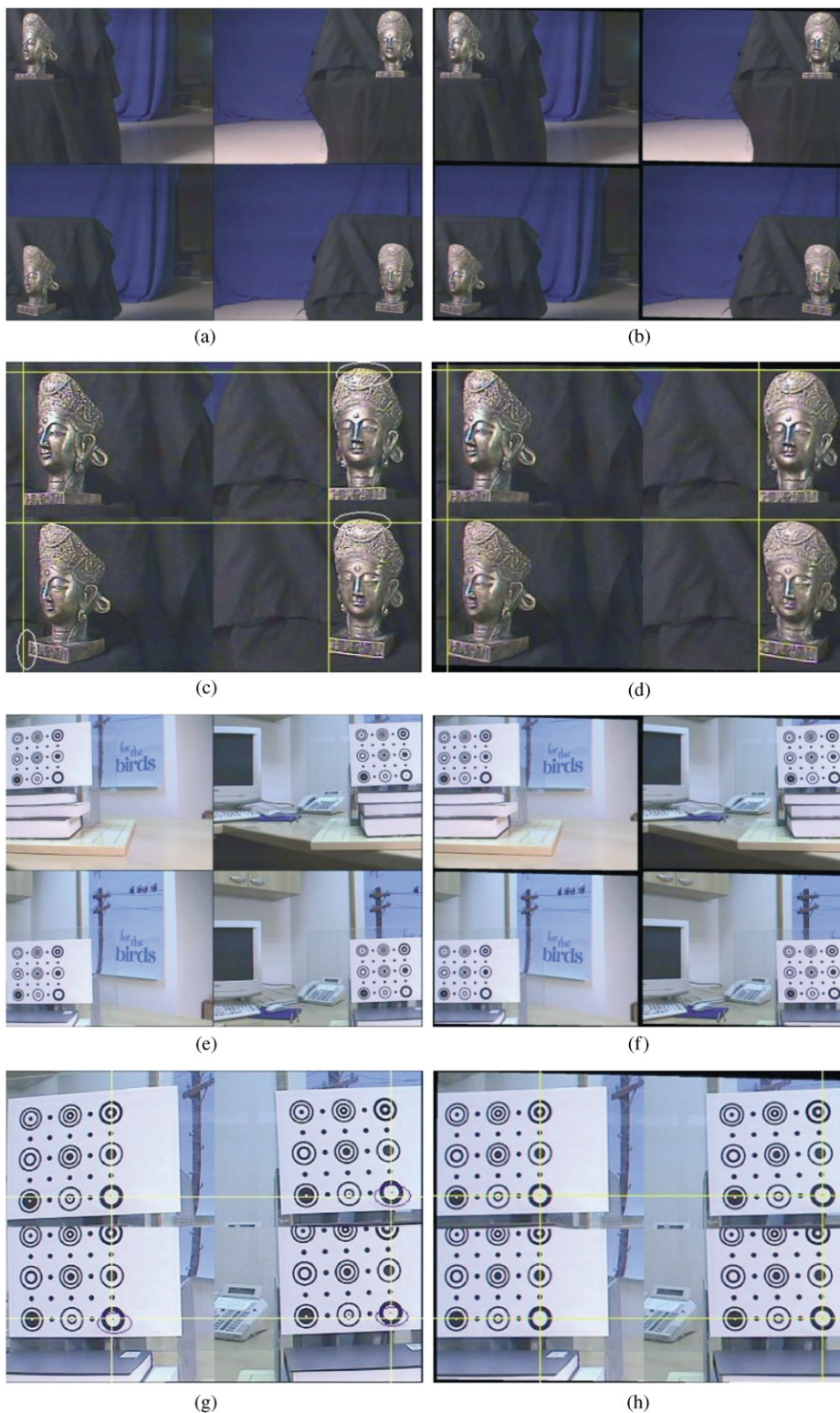
Fig. 10. The experimental results. The images in (a) and (e) are light fields after un-distortion and before rectification. Those in (b) and (f) are after rectification. (c), (d), (g) and (h) are the blown-up of the light fields before and after rectification. The horizontal and vertical lines are added to indicate the horizontal and vertical errors marked by ellipses.

occlude the markers a lot. Otherwise, we may capture the light field of the markers and compute the homography. This homography are also used to rectify the light field of interested object.

Till now, we only investigate the problem when the optical axis of the camera is not perpendicular to the camera plane, there is possibility that the camera is not at the desired position. The use of markers can also solve this problem. The positional error can be computed because the positions of the feature points are computable. Then we again use the positional error measured from the light field of markers to correct the interested light field. The error that the camera is not on the camera plane is negligible.

To demonstrate, our capturing device captures two light fields with the same configuration. We show the average error of the two light fields in Fig. 9. For one image in the light field, the error in x direction is defined as the average deviation in x of each feature point from the topmost image in the column; the error in y direction is defined as the average deviation in y of each feature point from the leftmost image in the row. The average errors in x and y are the mean error of each column and row, respectively. From this figure, we can see that the errors are unbiased. The errors of one light field are used as the benchmark. After subtraction, the errors reduce in both x and y directions, especially the maximal errors (Fig. 9).

3. Experimental results

Parts of some light fields before and after rectification are shown in Fig. 10. Before rectification, positions of correspondences in light field are not aligned vertically and horizontally. Take Fig. 10(a) for example, a feature point in the status on the top-left image is 5 pixels lower than its correspondence on the top-right one, and moves 3 pixels right compared to its correspondence on the bottom-left one (Fig. 10(c)). After rectification, the deviation is less than 1 pixel (due to the error of manually selecting feature points) without positional error correction (Figs. 10(b) and (d)). After correcting the positional error, the average error over the entire light field drops to less than 0.5 pixels. The results after positional error correction are not shown here because the further improvement is too small to be visually detected. These examples show that our algorithm is rather effective. The run time for detecting markers in four images and calculating the homography is 400–500 ms on a Pentium III 800 with 256 M RAM.

4. Conclusion

We have proposed an effective framework that both corrects the positional error of the light field and rectifies the light field, in which a light field of a pattern is taken to compute the rectification homography and positional error. Other light fields taken at the same positions of the camera use the same homography and the positional error. As the core of our framework, the rectifying homography is computed in a divide-and-conquer manner, and the iteration refined the homography. Our experiments show that our feature detection, positional error correction and rectification are all robust and accurate.

References

[1] Shum H-Y, Kang SB. A review of Image-based rendering techniques. IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000:2–13.

[2] Xu D, Pan Z, Shi J. Image-based rendering in virtual reality. China Journal of Image and Graphics 1998;3(12):1005–10.

[3] Levoy M, Hanrahan P. Light field rendering. In Proceedings of SIGGRAPH'96, 1996.

[4] Loop C, Zhang Z. Computing rectifying homographies for stereo vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999.

[5] Isgro F, Trucco E. Projective rectification without epipolar geometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999

[6] Fusiello A, Trucco E, Verri A. Rectification with unconstrained stereo geometry. In Proceedings of British Machine Vision Conference 1997. p. 400–9.

[7] Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF. The lumigraph. In Proceedings SIGGRAPH '96, 1996 p. 43–54.

[8] Cho Y, Lee J, Neumann U. A multi-ring color fiducial system and a rule-based detection method for scalable fiducial-tracking augmented reality. Proceedings of the First International Workshop on Augmented Reality, San Francisco, Nov. 1998.

[9] Hoff WA, Nguyen K. Computer vision-based registration techniques for augmented reality. Proceedings of Intelligent Robots and Computer Vision XV, SPIE Vol. 2904, Nov. 18–22, 1996, Boston, MA. p. 538–48.

[10] Intel Corporation, Open Source Computer Vision Library Reference Manual, 1999–2001.

[11] Jun-Sik K, In-So K. Camera calibration using projective invariance of concentric circles. Proceedings of Workshop on Image Processing and Image Understanding (IPIU), January, 2001 [In Korean].

[12] Xu G, Zhang Z. Epipolar geometry in stereo, motion and object recognition: a unified approach. Dordrecht: Kluwer Academic Publishers, 1996.