

Research Article

A Flexible Online Camera Calibration Using Line Segments

Yueqiang Zhang,^{1,2} Langming Zhou,^{1,2} Haibo Liu,^{1,2} and Yang Shang^{1,2}

¹College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, Hunan 410073, China

²Hunan Provincial Key Laboratory of Image Measurement and Vision Navigation, Changsha, Hunan 410073, China

Correspondence should be addressed to Haibo Liu; liuhaibo@nudt.edu.cn

Received 1 June 2015; Revised 23 September 2015; Accepted 27 September 2015

Academic Editor: Eugenio Martinelli

Copyright © 2016 Yueqiang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to make the general user take vision tasks more flexibly and easily, this paper proposes a new solution for the problem of camera calibration from correspondences between model lines and their noisy image lines in multiple images. In the proposed method the common planar items in hand with the standard size and structure are utilized as the calibration objects. The proposed method consists of a closed-form solution based on homography optimization, followed by a nonlinear refinement based on the maximum likelihood approach. To automatically recover the camera parameters linearly, we present a robust homography optimization method based on the edge model by redesigning the classic 3D tracking approach. In the nonlinear refinement procedure, the uncertainty of the image line segment is encoded in the error model, taking the finite nature of the observations into account. By developing the new error model between the model line and image line segment, the problem of the camera calibration is expressed in the probabilistic formulation. Simulation data is used to compare this method with the widely used planar pattern based method. Actual image sequences are also utilized to demonstrate the effectiveness and flexibility of the proposed method.

1. Introduction

Camera calibration has always been an important issue in the field of computer vision, since it is a necessary step to extract metric information from 2D images. The goal of the camera calibration is to recover the mapping between the 3D space and the image plane, which can be separated into two sets of transformations. The first transformation is mapping of the 3D points in the scene to the 3D coordinates in the camera frame, which is described by the extrinsic parameters of the camera model. The second one involves mapping of the 3D points in the camera frame to the 2D coordinates in the image plane. This mapping is described by the intrinsic parameters which models the geometry and optical features of the camera. In general case, these two transformations can be expressed by the ideal pin-hole camera model.

Up to now, much work for camera calibration has been done to accommodate various applications. Those approaches can be roughly grouped into two categories according to whether requiring a calibration object. This first type of camera calibration methods is named as metric calibration, which resolves the camera model with the help of metric information of a reference object. Camera

calibration is performed by observing a calibration object whose geometry dimension is known with very high precision. The calibration object can be 3D object with several planes orthogonal to each other [1, 2]. Sometimes a 2D plane undergoing a precisely known translation [3] or free movement [4] is utilized. Recently, a 1D temple [5–8] is used with three or more markers for camera calibration. In [6], it was proved that the 1D object undergoing a planar motion was essentially equivalent to the 2D planar object. For such type of methods, calibration can be done very efficiently and accurately. However, a calibration pattern also needs to be prepared, though in [4] the setup is very easy and only a planar object attached with the chessboard is utilized. Another type of camera calibration methods is called self-calibration which does not use any metric information from the scene or any calibration object. Such methods are also considered as 0D approach for only image feature correspondences are required. Since two constraints on the intrinsic parameter of the camera can be provided by using image information alone [9], three images are sufficient to recover the camera parameters including the internal and external parameters and reconstruct the 3D structure of the scene up to similarity [10, 11]. The problem of such methods is that a

large number of parameters need to be estimated, resulting in very unstable solution. If the camera rotation is known, more stable and accurate results can be obtained [12, 13]. However, it is not always easy to get the camera rotation with very high accuracy. In general, metric calibration methods can provide better results than self-calibration methods [14]. Our current research is focused on smartphone vision system since the potential for using such system is large. Smartphones are now becoming ubiquitous and popular in our daily life. To make the general public who are not experts in computer vision do vision tasks easily, the setup of camera calibrate should be flexible enough. The method developed in [4] is considered as the most flexible technique; however, when the orientation of the model plane with respect to image plane is increasing, foreshortening will make the corner detection less precise and even fail. Moreover, the planer pattern should be prepared, which is still inconvenient for general user of smartphone. Therefore, it would be best to utilize the handy item as the calibration object. The camera calibration technique described in this paper was designed with these considerations in mind. Compared with the classical techniques, the proposed technique does not need to prepare the planer pattern and is considerably more flexible. The calibration objects employed by the proposed method are common and handy in our daily life such as an A4 paper or even a standard IC card.

Our approach exploits the line/edge features of the handy objects to calibrate both the internal and external parameters of the camera, since they provide a large degree of stability to illumination and viewpoint changes and offer some resilience to hash imaging conditions such as noise and blur. A first challenge of the solution proposed in this paper is to automatically estimate the homography and establish the correspondences between model and image features. In this sense, we redesigned the model based tracking method [15–18] to robustly estimate homography for the common planar object in the clutter scene. An advantage of such methods is handling the occlusion, large illumination, and viewpoint change. With a series of homography from the planar object to the image plane, the initial camera parameters can be solved linearly. A second challenge is to optimize the camera parameters by developing effective object function and by making full use of the finite nature of the observation extracted in the images. In this paper, the error function for the model and image line, which encodes the length of the image line segment and the information of the midpoint, is derived from the noisy image edge points in the least square approach.

The remainder of the paper is organized as follows. Section 2 gives the procedure of the proposed camera calibration algorithm. Section 3 presents an overview of the redesigned homography tracking method based on edge model. Section 4 derives the error model between image and model lines and expresses the problem of the camera calibration in the probabilistic formulation by the maximum likelihood approach. Section 5 details how to solve the problem of camera calibration by the nonlinear technique. Some experiment results are given in Section 6.

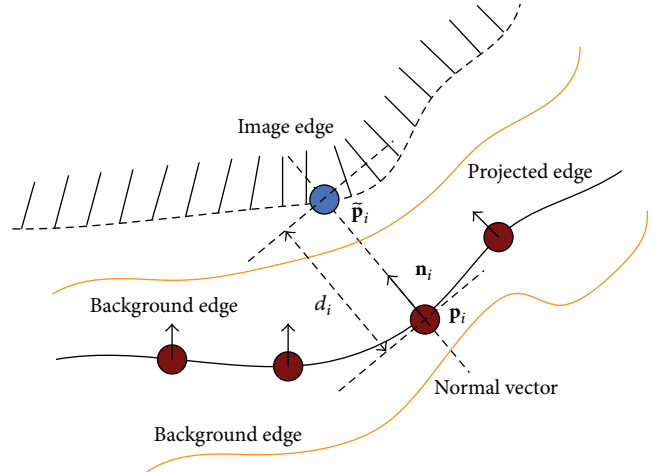


FIGURE 1: 1D search from the model line to the image line.

2. Algorithm

The proposed algorithm is summarized in this section.

Step 1. Optimize the homography between the model plane and image plane according to our model based homography tracking approach.

Step 2. Fit the image line segment from the image edge points obtained by 1D search along the normal direction of the corresponding model line.

Step 3. Calculate the initial camera parameters linearly with a series of homography matrices (more than three orientations).

Step 4. Estimate the camera parameters by minimizing the sum of the distance between the finite image line segments and the model lines in the maximum likelihood approach.

3. Model Based Homography Tracking

As can be seen in Figure 1, the 2D model edge is projected to the image plane using the prior homography of the planar object. Instead of tackling the line segment itself, we sample the projected line segment (black solid line in Figure 1) with a series of points (brown points in Figure 1). Then the visibility test for each of the sample points is performed, since some of these sample points may be out of the camera's view field. For each of the visible sample points, 1D search along the normal direction of the projected model line is employed to find the edge point with the strongest gradient or closest location as its correspondence. Finally, the sum of the errors between the sample points and their corresponding image points is minimized to solve for the homography between frames subsequently.

3.1. Probabilistic Formulation for Homography Tracking. The relationship between a model point \mathbf{P} and its image point \mathbf{p} can be given as

$$\begin{aligned}\mathbf{p} &= \Psi(\mathbf{P}, \mathbf{H}) = \left[\begin{array}{c} (\mathbf{HP})_x \\ (\mathbf{HP})_y \\ (\mathbf{HP})_z \end{array} \frac{(\mathbf{HP})_y}{(\mathbf{HP})_z} \right]^T, \\ \mathbf{P} &= (P_x \ P_y \ 1)^T, \\ \mathbf{p} &= (p_x \ p_y \ 1)^T,\end{aligned}\quad (1)$$

where \mathbf{H} is the homography between the model plane and the image plane.

Suppose $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_S$ is the set of projected sample points and $\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \dots, \tilde{\mathbf{p}}_S$ is their corresponding image points with the presence of the observation noise along the normal direction. Then we can define a function to measure the normal distance between a projected sample point \mathbf{p}_i and its noisy observation $\tilde{\mathbf{p}}_i$:

$$d_i = \mathbf{n}_i^T (\mathbf{p}_i - \tilde{\mathbf{p}}_i), \quad (2)$$

where \mathbf{n}_i is the unit normal vector of the projected sample point \mathbf{p}_i .

Assuming a Gaussian distribution for d_i , then we have

$$p(\tilde{\mathbf{p}}_i | \mathbf{p}_i) = N(d_i; 0, \sigma^2). \quad (3)$$

The conditional density of \mathbf{p}_i given $\tilde{\mathbf{p}}_i$ can be given by

$$p(\mathbf{p}_i | \tilde{\mathbf{p}}_i) = p(\tilde{\mathbf{p}}_i | \mathbf{p}_i) \frac{p(\mathbf{p}_i)}{p(\tilde{\mathbf{p}}_i)} = p(\tilde{\mathbf{p}}_i | \mathbf{p}_i). \quad (4)$$

Therefore, with the assumption that the observation errors for different sample points are statistically independent, a maximum likelihood estimation of the homography is

$$\begin{aligned}\hat{\mathbf{H}} &= \arg \min_{\mathbf{H}} \left(-\log \prod_{i \in [1, S]} p(\Psi(\mathbf{P}_i, \mathbf{H}) | \tilde{\mathbf{p}}_i) \right) \\ &= \arg \min_{\mathbf{H}} \sum_{i=1}^S (d_i)^2,\end{aligned}\quad (5)$$

where S is the number of 3D mode points.

It is clear that proposed approach can obtain the maximum likelihood estimation of the homography by minimizing the sum of the square of normal distances.

3.2. Interaction Matrix-Distance between Points. The derivation of the interaction matrix for the proposed approach is based on the distance between the projection of sample point \mathbf{p}_i and its projected image point $\tilde{\mathbf{p}}_i$. The motion velocity of the object is then related to the velocity of these distances in the image.

Assume that we have a current estimation of the homography \mathbf{H}_t . The posterior homography \mathbf{H}_{t+1} can be computed from the prior homography \mathbf{H}_t given the incremental motion $\Delta\mathbf{H}$:

$$\mathbf{H}_{t+1} = \Delta\mathbf{H}\mathbf{H}_t. \quad (6)$$

$\Delta\mathbf{H}$ can be represented as follows:

$$\Delta\mathbf{H} = \mathbf{I}_{3 \times 3} + \begin{pmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{pmatrix}. \quad (7)$$

The motion in the image is related to the twist in model space by computing the partial derivative of the normal distance with respect to j th generating motion at current homography:

$$\begin{aligned}J_{ij} &= \frac{\partial d_i}{\partial h_j} = \mathbf{n}_i^T \frac{\partial}{\partial h_j} (\Psi(\mathbf{P}_i, \Delta\mathbf{H}\mathbf{H}_t)) \\ &= \mathbf{n}_i^T \left\{ \frac{\partial \mathbf{P}_i}{\partial \mathbf{q}_i} \right\}_{2 \times 3} \left\{ \frac{\partial \mathbf{q}_i}{\partial h_j} \right\}_{3 \times 1} = \mathbf{n}_i^T \mathbf{J}_i^M \tilde{\mathbf{J}}_{ij},\end{aligned}\quad (8)$$

where $\mathbf{p}_i = \mathbf{q}_i/q_z^i$, $\mathbf{q}_i = [q_x^i, q_y^i, q_z^i]^T = \Delta\mathbf{H}\mathbf{H}_t\mathbf{P}_i$.

Then the corresponding Jacobian matrices can be obtained by

$$\begin{aligned}\mathbf{J}_i^M &= \begin{pmatrix} \frac{1}{q_z^i} & 0 & -\frac{q_x^i}{(q_z^i)^2} \\ 0 & \frac{1}{q_z^i} & -\frac{q_y^i}{(q_z^i)^2} \end{pmatrix}, \\ \tilde{\mathbf{J}}_{ij} &= \begin{pmatrix} \mathbf{P}_{i,t}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{P}_{i,t}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{P}_{i,t}^T \end{pmatrix} \mathbf{g}_j,\end{aligned}\quad (9)$$

where \mathbf{g}_j is a 9×1 unit vector with the j th item equal to 1 and $\mathbf{P}_{i,t} = \mathbf{H}_t\mathbf{P}_i$.

3.3. Robust Minimization. The error vector \mathbf{d} is obtained by stacking all of the normal distances of each sample point as follows:

$$\mathbf{d} = (d_1 \ d_2 \ d_3 \ \dots \ d_i \ \dots \ d_S)^T. \quad (10)$$

The optimization problem for (5) can be solved according to the following equation:

$$\mathbf{W}\mathbf{J}\mathbf{h} = \mathbf{e} = \mathbf{W}\mathbf{d}, \quad (11)$$

where \mathbf{h} is the motion vector, \mathbf{J} is Jacobian matrix which links \mathbf{d} to \mathbf{h} , and $\mathbf{W} = \text{diag}(\omega_1, \omega_2, \dots, \omega_i, \dots, \omega_S)$ is the weight matrix (refer to [17]).

Then, the solution of (11) can be given by

$$\hat{\mathbf{h}} = (\mathbf{W}\mathbf{J})^+ \mathbf{W}\mathbf{d}, \quad (\mathbf{W}\mathbf{J})^+ = ((\mathbf{W}\mathbf{J})^T (\mathbf{W}\mathbf{J}))^{-1} (\mathbf{W}\mathbf{J})^T. \quad (12)$$

Finally, the new homography can be computed according to (7) as follows:

$$\hat{\mathbf{H}}_{t+1} = \Delta\hat{\mathbf{H}}\mathbf{H}_t, \quad \Delta\hat{\mathbf{H}} = \begin{pmatrix} 1 + \hat{h}_0 & \hat{h}_1 & \hat{h}_2 \\ \hat{h}_3 & 1 + \hat{h}_4 & \hat{h}_5 \\ \hat{h}_6 & \hat{h}_7 & 1 + \hat{h}_8 \end{pmatrix}. \quad (13)$$

With a series of homography matrices (more than three orientations), the camera parameters can be solved linearly by method [4].

4. Maximum Likelihood Estimation of the Camera Parameters

In this paper, the camera calibration problem can be formulated in terms of a conditional density function that measures the probability of the image observations predicted from the camera parameters given the actual image observations. This section describes how to construct this conditional density function.

4.1. Probabilistic Formulation for the Camera Calibration. Consider the case where there are N images of a static scene containing M straight line segments. Let $\{(\mathbf{L}_i^j, \tilde{\mathbf{I}}_i^j)\}_{i=1}^M$ be the matched set of 3D model and 2D image lines in the image j , which can be established automatically according to homography optimization in this paper. With the assumption that the observation errors for different line segments are statistically independent, the conditional density function \mathbf{P} of the camera parameters can now be defined as follows:

$$\mathbf{P} = \prod_{j \in [1, N]} \prod_{i \in [1, M]} p(\Gamma(\mathbf{L}_i^j, \boldsymbol{\kappa}, \boldsymbol{\mu}_j) | \tilde{\mathbf{I}}_i^j), \quad (14)$$

where $\Gamma(\cdot)$ is the projection function which takes the camera parameters and the 3D line segment and returns the corresponding edge in the image. $\boldsymbol{\kappa}$, $\boldsymbol{\mu}_j$ are the intrinsic and extrinsic parameters of the camera in the image j , respectively. $p(\cdot)$ denotes the conditional density.

Then, the maximum likelihood estimation of the camera parameters $\boldsymbol{\kappa}$, $\boldsymbol{\mu}_j$ is maximizing the conditional density function \mathbf{P} , which is given by

$$\left(\hat{\boldsymbol{\kappa}}, \{\hat{\boldsymbol{\mu}}_j\}_{j=1}^N\right) = \arg \max_{\boldsymbol{\kappa}, \{\boldsymbol{\mu}_j\}_{j=1}^N} \mathbf{P}. \quad (15)$$

By taking the negative logarithm, the problem of maximize a product is converted into a minimization of a sum, which is given as follows:

$$\begin{aligned} & \left(\hat{\boldsymbol{\kappa}}, \{\hat{\boldsymbol{\mu}}_j\}_{j=1}^N\right) \\ & = \arg \min_{\boldsymbol{\kappa}, \{\boldsymbol{\mu}_j\}_{j=1}^N} \sum_{j \in [1, N]} \sum_{i \in [1, M]} -\ln \left(p(\Gamma(\mathbf{L}_i^j, \boldsymbol{\kappa}, \boldsymbol{\mu}_j) | \tilde{\mathbf{I}}_i^j) \right). \end{aligned} \quad (16)$$

The intrinsic parameters of the camera $\boldsymbol{\kappa}$ are represented as $[f_x, f_y, c_x, c_y, k]^T$, where f_x and f_y are the equivalent focal length, (c_x, c_y) is the principal point of the camera, and k is the radial distortion coefficient. The extrinsic parameters of the camera $\boldsymbol{\mu}_j$ in the image j are presented in the usual manner by a translation vector $\mathbf{T}_j \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R}_j \in \text{SO}(3)$. In the remainder of this section, the elements of (16) are discussed in more detail.

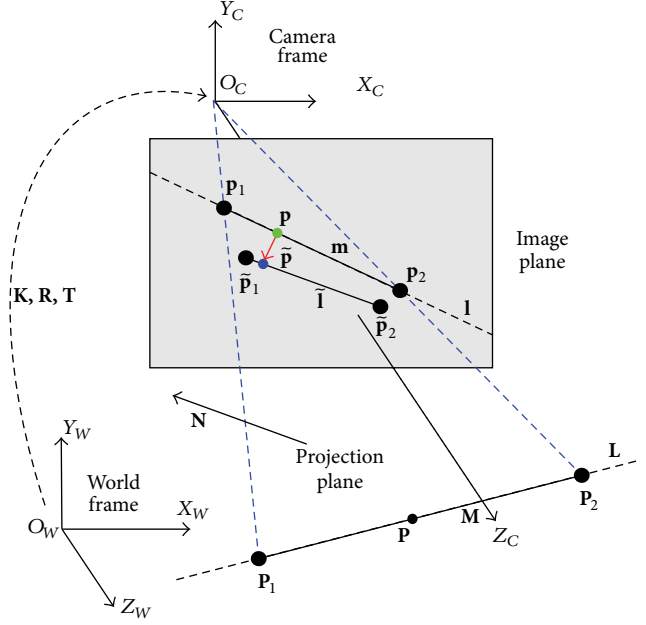


FIGURE 2: Perspective projection of 3D line and point.

4.2. Perspective Projection of 3D Model Line Segment. Throughout this paper, the perspective projection model is utilized. The relationship between a 3D world \mathbf{P} and 2D image point \mathbf{p} can be given as

$$\begin{aligned} \mathbf{p} &= [u \ v \ 1]^T = \Gamma(\mathbf{P}, \boldsymbol{\kappa}, \boldsymbol{\mu}), \\ u &= f_x \frac{(\mathbf{P}_C)_x}{(\mathbf{P}_C)_z} + c_x = p_x + c_x, \\ v &= f_y \frac{(\mathbf{P}_C)_y}{(\mathbf{P}_C)_z} + c_y = p_y + c_y, \end{aligned} \quad (17)$$

where $\mathbf{P}_C = \mathbf{R}\mathbf{P} + \mathbf{T}$ is the coordinate in the camera frame for \mathbf{P} and $(\cdot)_z$ is the z -coordinate. f_x and f_y are the equivalent focal length. c_x and c_y are the principal point. δ_x and δ_y are the radial distortion, which is modeled as one-order polynomial model:

$$\begin{aligned} \delta_x &= f_x \bar{p}_x (\bar{p}_x^2 + \bar{p}_y^2) k, \\ \delta_y &= f_y \bar{p}_y (\bar{p}_x^2 + \bar{p}_y^2) k, \\ \bar{p}_x &= \frac{(u - c_x)}{f_x}, \\ \bar{p}_y &= \frac{(v - c_y)}{f_y}, \end{aligned} \quad (18)$$

where $\bar{\mathbf{p}} = (\bar{p}_x, \bar{p}_y)^T$ is corresponding to the projection ray from the focal point to the image point \mathbf{p} .

As shown in Figure 2, the line segment \mathbf{M} and its projection \mathbf{m} in the image plane are represented by their endpoints $(\mathbf{P}_1, \mathbf{P}_2)$ and $(\mathbf{p}_1, \mathbf{p}_2)$. The line segments \mathbf{M} and \mathbf{m} lie on the

infinite lines \mathbf{L} and \mathbf{I} , respectively. The perspective projection of 3D line segment can be given by the projection of its two endpoints:

$$\mathbf{m} = \Gamma(\mathbf{M}, \boldsymbol{\kappa}, \boldsymbol{\mu}) = (\Gamma(\mathbf{P}_1, \boldsymbol{\kappa}, \boldsymbol{\mu}), \Gamma(\mathbf{P}_2, \boldsymbol{\kappa}, \boldsymbol{\mu})). \quad (19)$$

When noise is present in the measuring data, we denote $\tilde{\mathbf{p}}$ as the noisy observation of the projection of the 3D points \mathbf{P} and $\tilde{\mathbf{I}}$ as the noisy observation of the projection of 3D model line \mathbf{L} .

4.3. Error Model for the Observation of the Line Segment. Let $\tilde{\mathbf{p}}_1 = (x_1, y_1)$, $\tilde{\mathbf{p}}_2 = (x_2, y_2), \dots, \tilde{\mathbf{p}}_n = (x_n, y_n)$ be a series of image edge points with the presence of the observation noise perpendicular to the line. For convenience, we assume that the true position of the line is parallel to the horizontal axis. Then we have

$$y_i = ax_i + b + e_i, \quad i = 1, 2, \dots, n, \quad (20)$$

where $e_1, e_2, \dots, e_i, \dots, e_n$ are Gaussian random variables with $Ee_i = 0$, $De_i = \sigma^2$ and they are mutually independent.

Let the noises for the endpoints along the vertical direction be s_1, s_2 , respectively, and $\mathbf{s} = (s_1, s_2)^T$. It can be easily derived that

$$D(s_1) = \frac{1}{nC_{xx}} \sigma^2 \sum_{i=1}^n (x_i - x_1)(x_i - x_1),$$

$$D(s_2) = \frac{1}{nC_{xx}} \sigma^2 \sum_{i=1}^n (x_i - x_n)(x_i - x_n), \quad (21)$$

$$\text{Cov}(s_1, s_2) = \frac{1}{nC_{xx}} \sigma^2 \sum_{i=1}^n (x_i - x_1)(x_i - x_n),$$

where $\bar{x} = (1/n) \sum_{i=1}^n x_i$, $C_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n (x_i)^2 - n\bar{x}^2$.

It is clear that these two noises are negatively correlative. Since the observation noises conform to Gaussian random variables, the joint density for the random variables s_1 and s_2 is a Gaussian PDF, which can be given by

$$p(\mathbf{s}) = \frac{1}{2\pi |\mathbf{V}|^{1/2}} \exp(-\mathbf{s}^T \mathbf{V}^{-1} \mathbf{s}). \quad (22)$$

Supposing that the length of the line segment is l and the intervals of the edge points are all l_0 , then we have $x_1 = l_0$, $x_2 = 2l_0, \dots, x_n = nl_0$. Therefore, we obtain

$$D(s_1) = D(s_2) = \frac{2l_0\sigma^2 * (2l + l_0)}{(l + l_0)(l + 2l_0)},$$

$$\text{Cov}(s_1, s_2) = -\frac{2l_0\sigma^2 * (l - l_0)}{(l + l_0)(l + 2l_0)}. \quad (23)$$

When the number n is large enough, that is, $l_0 \ll l$, it is easy to obtain

$$\mathbf{V}^{-1} = \frac{4}{3\sigma^2} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} = \frac{l}{3l_0\sigma^2} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}. \quad (24)$$

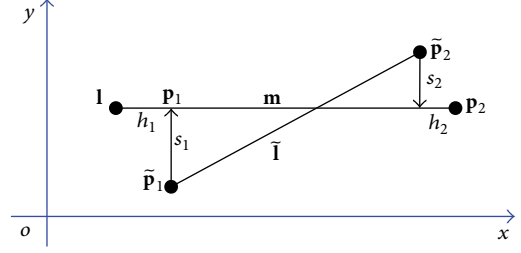


FIGURE 3: Relation between projection of the 3D line segment and its noisy observation.

Substituting (24) into (22) yields

$$p(\mathbf{s} | \mathbf{I}) = \frac{1}{2\pi |\mathbf{V}|^{1/2}} \cdot \exp\left(-\frac{1}{l_0\sigma^2} \left(\frac{l}{3} ((s_1)^2 + s_1s_2 + (s_2)^2)\right)\right). \quad (25)$$

From (25), it can be seen that the error model allows us to encode the measurement error for image edge point (σ) explicitly and obtain the intuitive impact of image line length. Moreover, long line segments produce more accurate location than shorter ones and small σ produces higher confidence about the line location.

4.4. Maximum Likelihood (ML) Estimation. The measurement noise for the localization of the 2D line segments can be decomposed into two components: noise perpendicular to the line and noise along the length of the line. The first noise is modelled as a Gaussian random variable related to orientation error and the noise model has been derived in the last section, whilst the second one is assumed to conform to any distribution (not necessarily Gaussian) related to line fragmentation.

As can be seen in Figure 3, both the projection of the 3D line segment \mathbf{m} and its noisy observation $\tilde{\mathbf{I}}$ are represented by their endpoints, $(\mathbf{p}_1, \mathbf{p}_2)$ and $(\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2)$, respectively. The noise vector \mathbf{s} perpendicular to the line and the noise vector \mathbf{h} along the line are expressed as follows:

$$\mathbf{s} = [s_1 \ s_2]^T, \quad (26)$$

$$\mathbf{h} = [h_1 \ h_2]^T,$$

where the components of \mathbf{s} are the distances between the endpoints of $\tilde{\mathbf{I}}$ and \mathbf{m} along the direction perpendicular to \mathbf{m} . The components of \mathbf{h} are the distances between the endpoints of the two line segments along the direction of \mathbf{m} .

It is assumed that the two random vectors \mathbf{s} and \mathbf{h} are statistically independent. And then we can approximate the conditional density of $\tilde{\mathbf{I}}$ given \mathbf{m} as

$$p(\tilde{\mathbf{I}} | \mathbf{m}) = p(\mathbf{s} | \mathbf{m}) p(\mathbf{h} | \mathbf{m}). \quad (27)$$

In the literature [19], it is proved that the conditional density of the projection of the 3D model line \mathbf{I} given its

observed noisy image line segment $\tilde{\mathbf{I}}$ is only dependent on the noise perpendicular to line \mathbf{l} :

$$p(\mathbf{I} | \tilde{\mathbf{I}}) = p(\mathbf{s} | \mathbf{I}). \quad (28)$$

Therefore, with the assumption that the observation errors for different line segments are statistically independent, (16) can be converted into the following formation:

$$\begin{aligned} (\hat{\boldsymbol{\kappa}}, \{\hat{\boldsymbol{\mu}}_j\}_{j=1}^N) &= \arg \min_{\boldsymbol{\kappa}, \{\boldsymbol{\mu}_j\}_{j=1}^N} O, \\ O &= \sum_{j \in [1, N]} \sum_{i \in [1, M]} (\mathbf{s}_i^j)^T (\mathbf{V}_i^j)^{-1} \mathbf{s}_i^j, \end{aligned} \quad (29)$$

where O is the objective function that measures the disparity between the actual image observations and their corresponding predicted ones by the current camera parameters. \mathbf{s} corresponds to the distances from the endpoints of the image line segment to the projected model line.

If the image line segment is fitted by LST and the intervals of the edge points are fixed for all of the image line segments, then we have

$$\begin{aligned} O &= \sum_{j \in [1, N]} \sum_{i \in [1, M]} \left(\frac{1}{l_0 \sigma^2} \left(\frac{l_i^j}{3} \left((s_{1i}^j)^2 + s_{1i}^j s_{2i}^j + (s_{2i}^j)^2 \right) \right) \right) \\ &\approx \sum_{j \in [1, N]} \sum_{i \in [1, M]} \frac{l_i^j}{6} \left((s_{1i}^j)^2 + 4 * (s_{0i}^j)^2 + (s_{2i}^j)^2 \right), \end{aligned} \quad (30)$$

where s_{1i}^j , s_{2i}^j , and s_{0i}^j correspond to the distances from the two endpoints and midpoint of the image line segment $\tilde{\mathbf{I}}_i^j$ to the projected model line \mathbf{l}_i^j . It is clear that the error function between 3D model line and 2D image line is weighted by the length of the image line segment.

5. Nonlinear Technique for the Optimization of Camera Parameters

In this section, we will describe how to employ the nonlinear technique to solve the problem of camera calibration defined in the previous section. In the initial case, the camera parameters can be provided by the method which is similar to [4] except that the homography matrices are calculated by the method discussed in Section 2, rather than the chessboard corners. At each iteration, the linearized error function is minimized to obtain the interframe motion vector for the intrinsic and extrinsic parameters. Then the camera parameters are updated until the objective function converges to a minimum.

The distance from the point of the image line segment to the projection of the model line is given by

$$s = \frac{1}{\lambda} \bar{\mathbf{p}} \cdot ((\mathbf{R}\mathbf{P}_1 + \mathbf{T}) \times (\mathbf{R}\mathbf{P}_2 + \mathbf{T})), \quad (31)$$

where $\lambda = \sqrt{(A_x/f_x)^2 + (A_y/f_y)^2}$ (refer to [20]).

Assume that we have a current estimation of the rotation \mathbf{R}_t at the time of t . The posterior rotation \mathbf{R}_{t+1} can be computed from the prior rotation \mathbf{R}_t given the incremental rotation $\exp(\hat{\boldsymbol{\omega}})$:

$$\mathbf{R}_{t+1} = \exp(\hat{\boldsymbol{\omega}}) \mathbf{R}_t, \quad (32)$$

where $\hat{\boldsymbol{\omega}}$ is the corresponding skew-symmetric matrix of vector $\boldsymbol{\omega}$:

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (33)$$

The transformation from the reference frame to the camera frame can be rewritten as

$$\mathbf{P}_C = \mathbf{R}(\mathbf{P} - \mathbf{T}^W), \quad (34)$$

where $\mathbf{T}^W = -\mathbf{R}^T \mathbf{T}$ denotes the location of the origin of the camera frame in the world frame.

Let $\boldsymbol{\xi} \in \mathbb{R}^3$ represent the motion velocities corresponding to translation in the x , y , and z directions between the prior translations \mathbf{T}_t^w and the posterior translation \mathbf{T}_{t+1}^w . Equation (31) can be rewritten as

$$\begin{aligned} s_{i,m}^j &= \frac{1}{\lambda_{i,t}^j} (\mathbf{R}_{t+1}^T \bar{\mathbf{P}}_{i,m}^j) \\ &\cdot ((\mathbf{P}_{i,1}^j - \mathbf{T}_t^W - \boldsymbol{\xi}) \times (\mathbf{P}_{i,2}^j - \mathbf{T}_t^W - \boldsymbol{\xi})). \end{aligned} \quad (35)$$

Then, the partial derivative of the error function $s_{i,m}^j(\mathbf{r})$ with respect to the n th motion velocities can be computed as

$$\begin{aligned} \frac{\partial s_{i,m}^j}{\partial \omega_n^j} &= -\frac{1}{\lambda_{i,t}^j} \left((\mathbf{R}_t^j)^T \hat{\mathbf{g}}_n \bar{\mathbf{P}}_{i,m}^j \right) \\ &\cdot ((\mathbf{P}_{i,1}^j - \mathbf{T}_t^{j,W}) \times (\mathbf{P}_{i,2}^j - \mathbf{T}_t^{j,W})), \end{aligned} \quad (36)$$

$$\frac{\partial s_{i,m}^j}{\partial \xi_n^j} = \frac{1}{\lambda_{i,t}^j} \left((\mathbf{R}_t^j)^T \bar{\mathbf{P}}_{i,m}^j \right) \cdot (\hat{\mathbf{g}}_n (\mathbf{P}_{i,1}^j - \mathbf{P}_{i,2}^j)),$$

where $\mathbf{g}_1 = (1, 0, 0)^T$, $\mathbf{g}_2 = (0, 1, 0)^T$, and $\mathbf{g}_3 = (0, 0, 1)^T$.

The partial derivative of the error function $s_{i,m}^j(\mathbf{r})$ with respect to $\boldsymbol{\kappa}$ can be given by

$$\begin{aligned} \frac{\partial s_{i,m}^j}{\partial \boldsymbol{\kappa}} &= \frac{1}{\lambda_{i,t}^j} \left((\mathbf{R}_t^j)^T \frac{\partial \bar{\mathbf{q}}_{i,m}^j}{\partial \boldsymbol{\kappa}} \right) \\ &\cdot ((\mathbf{P}_{i,1}^j - \mathbf{T}_t^{j,W}) \times (\mathbf{P}_{i,2}^j - \mathbf{T}_t^{j,W})), \end{aligned} \quad (37)$$

where $\bar{\mathbf{q}} = \bar{\mathbf{p}} + \bar{\boldsymbol{\delta}}$, $\bar{\boldsymbol{\delta}} = (\bar{p}_x(\bar{p}_x^2 + \bar{p}_y^2)k, \bar{p}_y(\bar{p}_x^2 + \bar{p}_y^2)k, 0)^T$.

The error vector \mathbf{d} is obtained by stacking all of the normal distances of each image point as follows:

$$\mathbf{d} = \left((s_1^1)^T, (s_2^1)^T, \dots, (s_M^1)^T, \dots, (s_1^j)^T, \dots, (s_M^j)^T \right)^T, \quad (38)$$

where $\mathbf{s}_i^T = (s_0^i \ s_1^i \ s_2^i)$ is the distance vector from midpoint and endpoints of the image line segment to the projected model line.

The optimization problem for (30) can be solved according to the following equation:

$$\mathbf{W}\mathbf{J}\Phi = \mathbf{e} = \mathbf{W}\mathbf{d}, \quad (39)$$

where $\Phi = (\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \dots, \boldsymbol{\mu}_N^T, \Delta f_x, \Delta f_y, \Delta c_x, \Delta c_y, \Delta k)^T$ is the motion vector and $\boldsymbol{\mu} = (\boldsymbol{\omega}^T, \boldsymbol{\xi}^T)^T$. \mathbf{J} is Jacobian matrix which links \mathbf{s} to Φ and $\mathbf{W} = \text{diag}(w_1, w_2, \dots, w_m, \dots, w_{3*N*M})$ is the weight matrix (refer to [17]).

If the incremental motion vector has been calculated, the new camera parameters can be computed as follows:

$$\begin{aligned} \mathbf{R}_{t+1}^j &= (\mathbf{I} + \widehat{\boldsymbol{\omega}}^j) \mathbf{R}_t^j, \\ \mathbf{T}_{t+1}^j &= -\mathbf{R}_{t+1}^j (\mathbf{T}_t^{j,W} + \boldsymbol{\xi}^j), \\ f_{x,t+1} &= f_{x,t} + \Delta f_x, \\ f_{y,t+1} &= f_{y,t} + \Delta f_y, \\ c_{x,t+1} &= c_{x,t} + \Delta c_x, \\ c_{y,t+1} &= c_{y,t} + \Delta c_y, \\ k_{t+1} &= k_t + \Delta k. \end{aligned} \quad (40)$$

6. Experimental Results

The proposed algorithm has been tested on simulated data generated by the computer and real image data captured from our smartphone. The closed-form solution is yielded by the approach [4] except that the homography matrices are estimated by the proposed method. The nonlinear refinement within the IRLS algorithm takes 5 to 8 iterations to converge.

6.1. Computer Simulations. The simulated perspective camera is supposed to be 2 m from the plane object. The resolution of the virtual camera is 640×640 . The simulated camera has the following property: $f_x = f_y = 1814.8$, $c_x = c_y = 300$. The model plane is a checker pattern printed on the A4 paper (210 mm \times 297 mm) with 11×14 corners. The images are taken from different orientations in front of the virtual camera. The normal vector of the plane is parallel to the rotation axis represented by a 3D vector \mathbf{r} , whose magnitude is equal to the rotation angle. The position of the plane is represented by a 3D vector \mathbf{t} (unit in millimetres). In the experiment, the proposed method is compared with the widely used chessboard corners based method [4] (referred to as corners based method and the implementation is according to the related camera calibration function of OpenCV [21]). For the corners based method, 154 corners are utilized. In our method, we use 25 lines fitted from the noisy corners by the LST. The reprojection error indicated by RMS is expressed by the root of mean squared distances in pixels, between the detected image corners and the projected ones. When only four edges of the plane pattern are utilized, the proposed method is referred to as 4-line based method.

6.1.1. Performance with respect to the Noise Level. In this experiment, three planes with $\mathbf{r}_1 = [20, 0, 0]^T$, $\mathbf{t}_1 = [-105, -149, 2000]^T$, $\mathbf{t}_2 = [-105, -149, 2100]^T$, $\mathbf{r}_3 = [-14, -14, -7]^T$, and $\mathbf{t}_3 = [-105, -149, 2250]^T$ are used (the three orientations are chosen according to [4]). Zero mean Gaussian noise is added to the projected image points with the standard deviation σ ranging from 0.1 pixels to 2.0 pixels in steps of 0.1 pixels. At each noise level, 100 independent trials are generated. The estimated camera parameters are then compared with the ground truth and RMS errors are measured. Moreover, for 154 points with real projections and the recovered projections, the RMS reprojection error is also calculated. Figures 4(a) and 4(b) display the relative errors of the intrinsic parameters which are measured with respect to f_x , while Figure 4(c) shows the reprojection errors of the two methods.

From Figure 4, we can see that both the relative errors of the intrinsic parameters and the reprojection errors increase almost linearly with the noise level. The proposed method can produce the equivalent performance with the corners based methods since the image lines are fitted from the noisy image corners. When 4 lines (the smallest set for homography estimation) are utilized, the errors of the proposed method are larger than the corners based method. For $\sigma < 0.5$, there is little difference between the 4-line based method and the corners based method.

In addition, we vary the number of sample points that are utilized to fit the line segment to validate the performance of the 4-line based method with $\sigma = 0.5$. From the results in Figure 5, we can see that the errors decrease significantly when more sample points are utilized. When the number is above 40 where more than 160 are utilized to fit 4 line segments, the performance of the 4-line based method is almost similar to that of the 154-corner based method.

6.1.2. Performance with respect to the Number of Planes. In this experiment, we investigate the performance of the proposed method with respect to the number of the images of the model planes. In the first three images, we use the same orientation and position of the model plane as those used in the last subsection. For the following images, the rotation axes are randomly chosen in a uniform sphere with the rotation angle fixed to 30° and the positions are randomly selected around $[-105, -145, 2000]$. The number of the model plane images ranges from 3 to 17. At each number of the images, 100 independent trials of independent plane orientations are generated with the noise level for the image points fixed to 0.5 pixels. The errors including the relative errors in camera intrinsic parameters and the reprojection errors for the two methods are shown in Figure 6. The errors decrease when more images are used. From 3 to 7, the errors decrease significantly. Moreover, the reprojection errors of the proposed method are around 0.7, when the number of the images is varying.

6.1.3. Performance with respect to the Number of Lines. This experiment examines the performance of the proposed method with respect to the number of the lines utilized to recover the camera parameters. For our method, more than

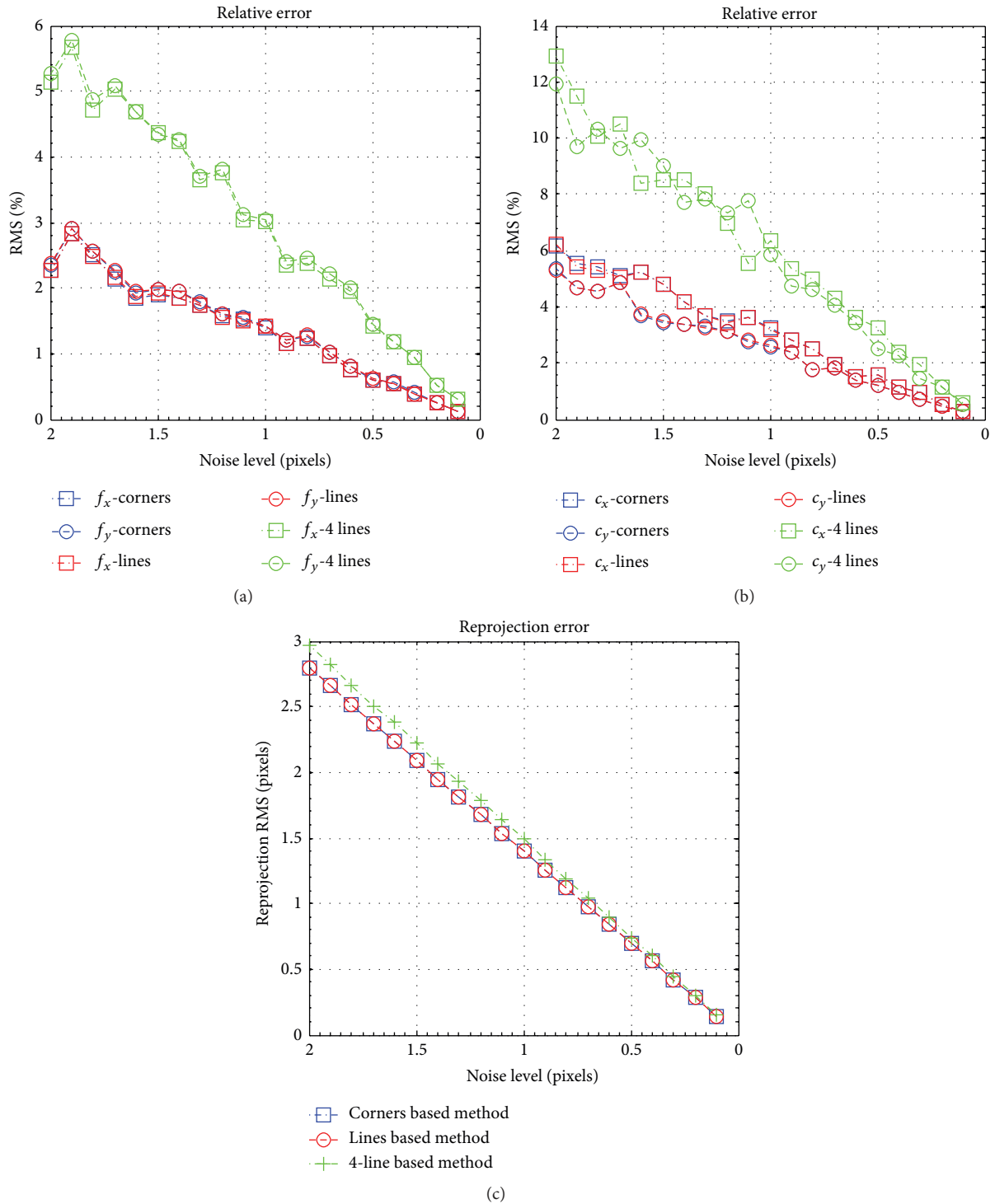


FIGURE 4: Errors versus the noise level of the image points.

4 lines should be employed. We vary the number of lines from 4 to 25. Three images of the model plane are also used with the same orientation and position as last subsection. 100 independent trials are conducted with the noise level fixed to 0.5 pixels for each number of the lines. The results are shown in Figure 7. When more lines are used, the errors decrease. In particular, from 4 to 15, the errors decrease significantly.

6.1.4. Performance with respect to the Orientation of the Model Plane. This subsection investigates the influence of the orientation of the model plane with respect to the image plane. In the experiment, three images are used with two of them similar to the last two planes in Section 6.1.1. The initial rotation axis of the third plane is parallel to the image plane, and the orientation of the planes is randomly chosen from

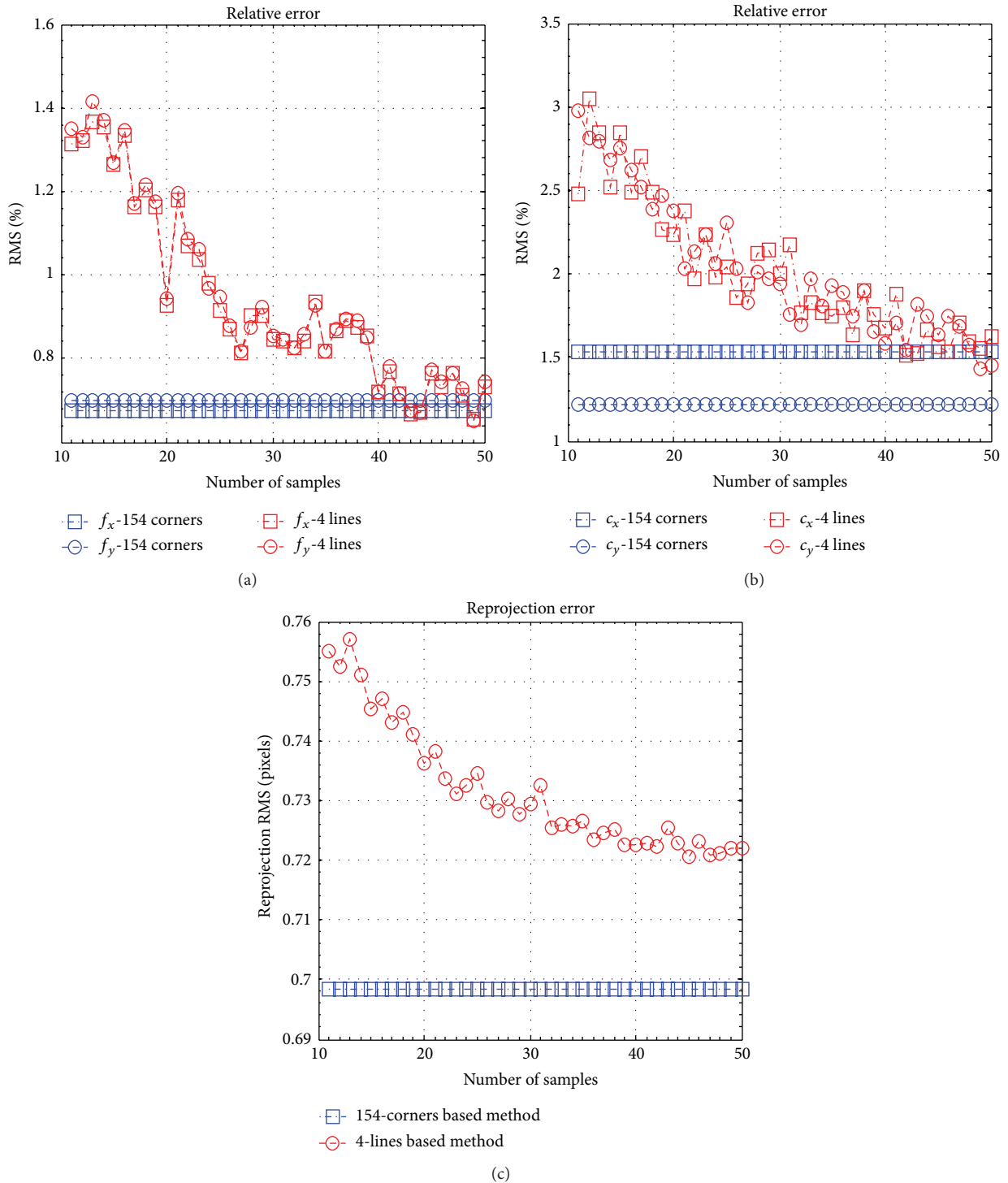


FIGURE 5: Errors versus the number of sample points for 4-line based method.

a uniform sphere with the rotation varying from 5° to 75° . The noise level is fixed to 0.5 pixels. The results are displayed in Figure 8. Best performance seems to be achieved with the angle around 40° .

6.2. Real Images. For the experiment with real data, the proposed algorithm is tested on several image sequences captured from the camera of the smartphone.

6.2.1. Homography Tracking Performance. In the experiment, three image sequences are captured from the smartphone with a resolution of 720×480 . In the first image sequence, a chessboard containing 10×13 interior corners is printed on an A4 paper and put on the desk. About 1500 frames are taken at different orientation. For each image, the homography from the model plane to the image plane is optimized by the proposed method using the four edges of the A4 paper.

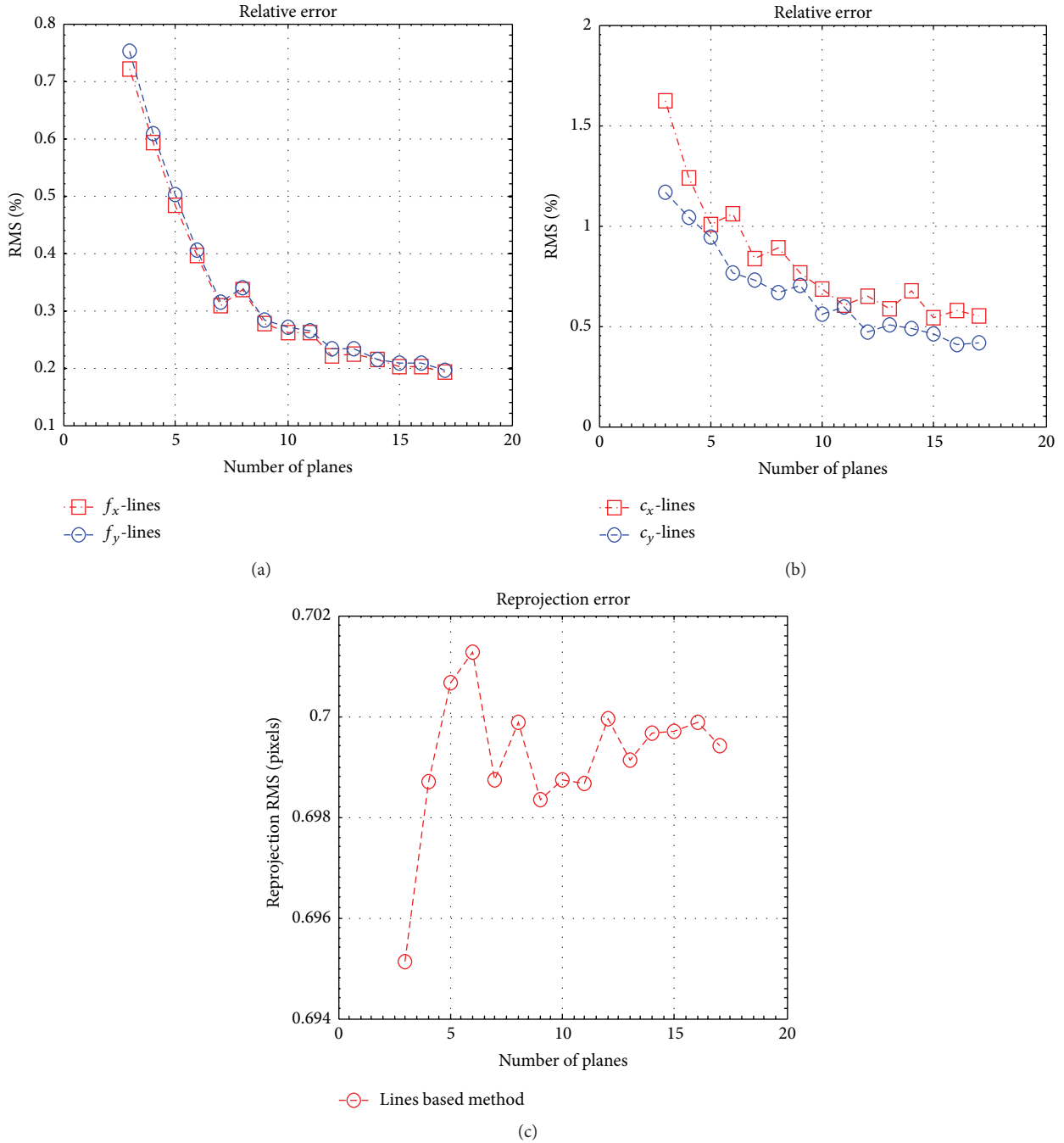


FIGURE 6: Errors versus the number of the images.

The interior corners are extracted by the function of *cvFindChessboardCorners* in OpenCV and refined by the function of *cvFindCornerSubPix*. Figure 9 shows some sampled results from the image sequence.

In the last two image sequences, the covers of two books are chosen as the model planes, respectively. To validate the performance of the proposed homography tracking method, the books are put in the clutter environment with the smartphone undergoing large rotation and translation. Both of the last two sequences contain around 2000 images. Figure 10 exhibits some sampled results.

6.2.2. Camera Calibration Performance. In this subsection, we applied our calibration technique and the corners based method to the four images sampled from the video captured by our smartphone (shown in Figure 11). The image resolution is 960×540 . In the experiment, the chessboard plane contains 10×13 interior corners and 23 lines. The results are shown in Table 1. We can see that there is little difference between the proposed method and the corners based method. When only four edges of the plane pattern are utilized, the proposed method can provide the very consistent results with the corners based method and the offset of the camera

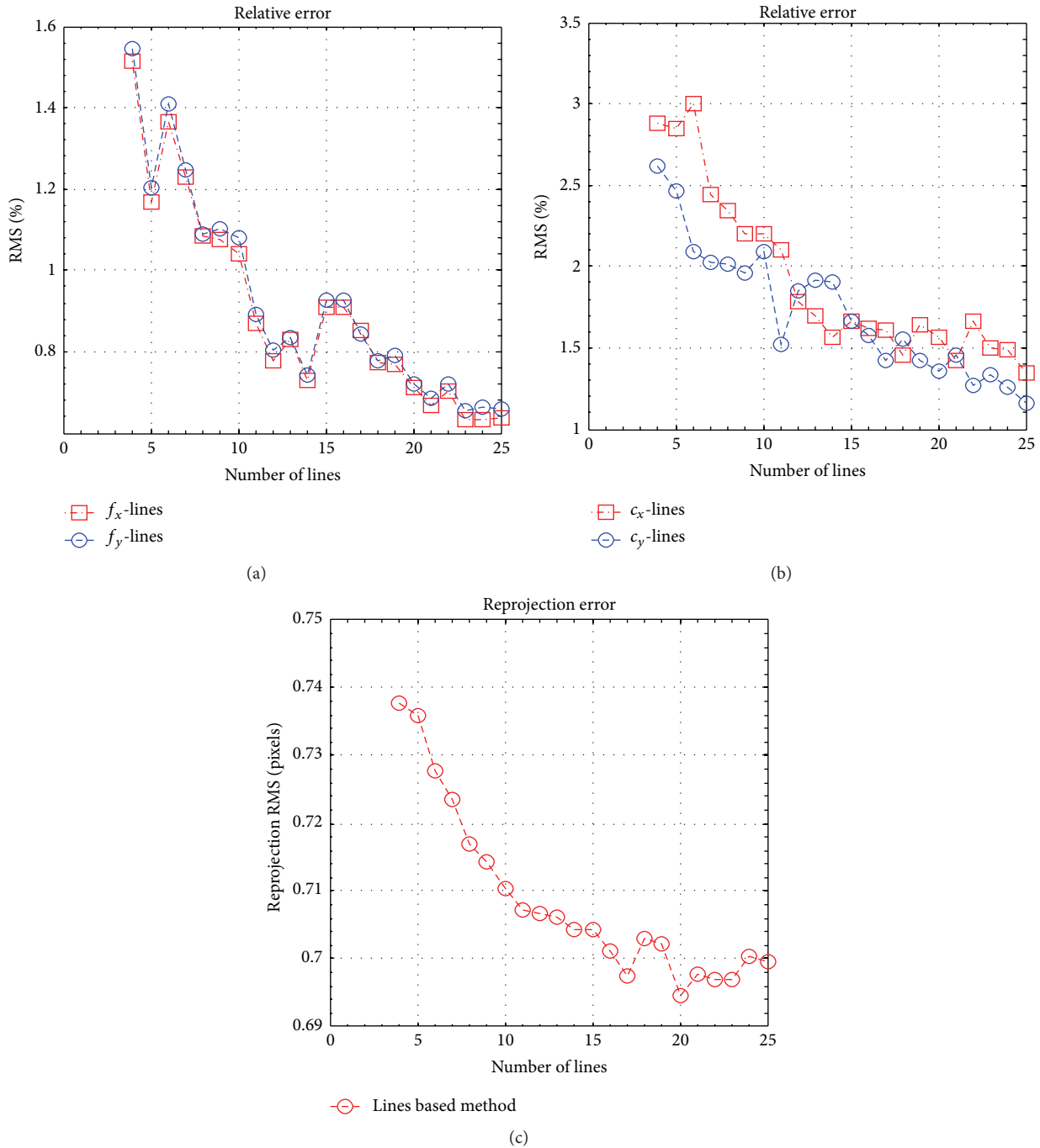


FIGURE 7: Errors versus the number of the lines.

parameters is very small about 5 pixels with respect to the corners based method. The last column of Table 1 shows the reprojection RMS of the three methods. When all of the 23 lines are utilized, the proposed method provides the almost same reprojection error as the corners based method. The 4-line based method returns the slightly larger reprojection error, since only the minimum of model lines are utilized.

In order to further investigate the stability of the proposed method, we vary the number of lines from 4 to 23. The results are shown in Figure 12. c_x and c_y recovered by the proposed

method are around the values estimated by the corners based method only with a small deviation. The reprojection errors for the projected method decrease significantly from 4 to 17. When the number is above 17, the reprojection error is very close to that of the corners based method.

6.2.3. Application to Image-Based Modelling. In this subsection, we applied the proposed method on two image sequences. In the first image sequence, the card with the size of 54.0 mm \times 85.6 mm is utilized as the model object.

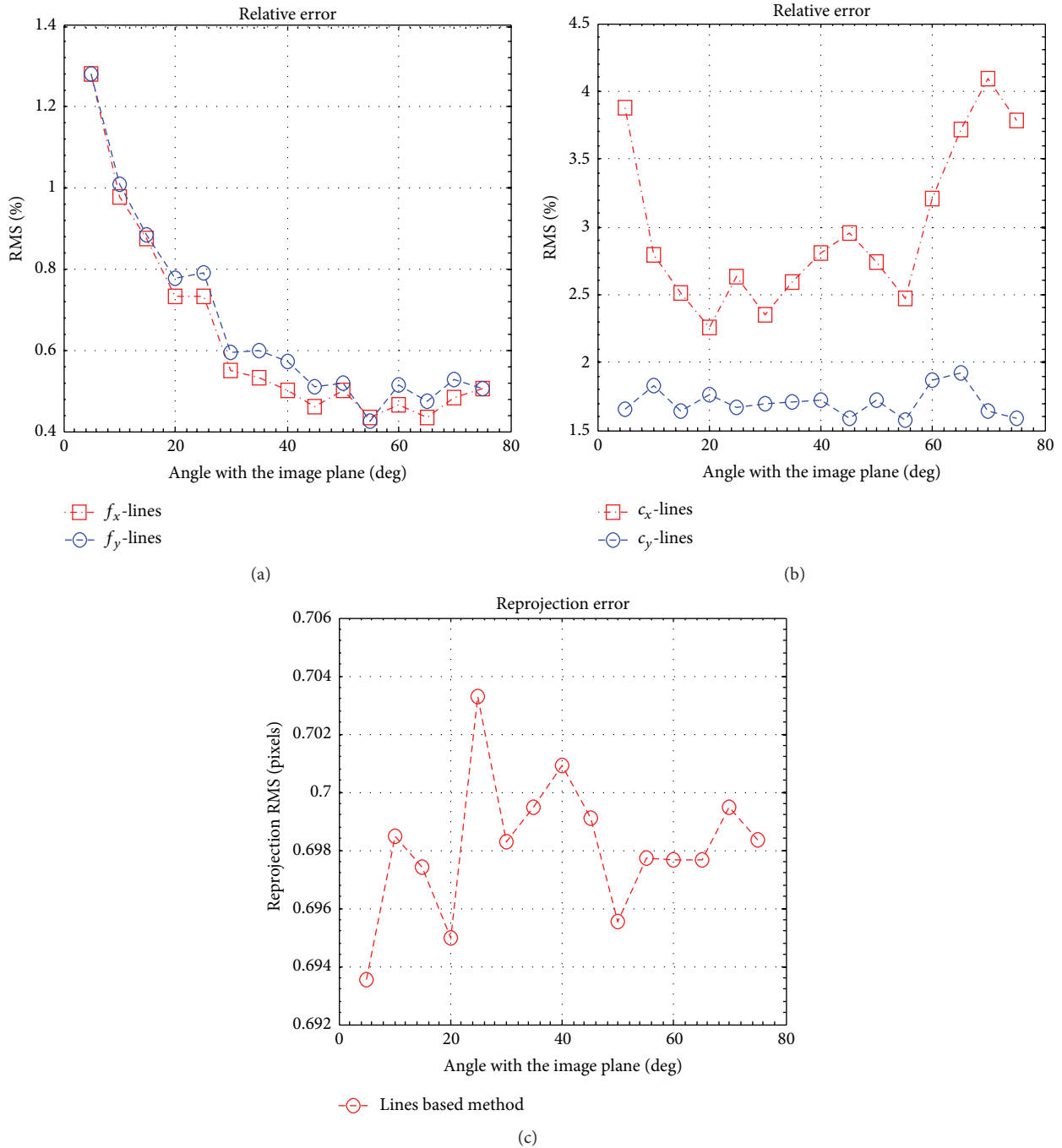


FIGURE 8: Errors versus the orientations of the plane.

The A4 paper with the size of 210 mm \times 297 mm is chosen as the model object for the second image sequence. In the experiment, a series of images are sampled from the videos to calibrate the camera intrinsic parameters and then the camera pose is optimized for each image frame. After that, the structure from motion developed by the methods [22–24] was run on the image sequences to build the complete models of the toys including Luffy and Hulk. In Figure 13, Figures (A), (B), (C), and (D) are some sampled images from the image sequences. The recovered camera poses by

the proposed method are shown in Figure (E). The left one of Figure (E) shows the camera poses for the whole image sequence, while the right one corresponds to the sampled views for the following reconstruction. By recovering the whole motion trajectory of the camera, we can easily choose a subset of the frames which are suitable and adequate for modeling. Two rendered views of the reconstructed objects are shown in Figure (F). From Figure 13, we can see that the complete model of the objects has been reconstructed by moving the camera around the objects. For the size of the

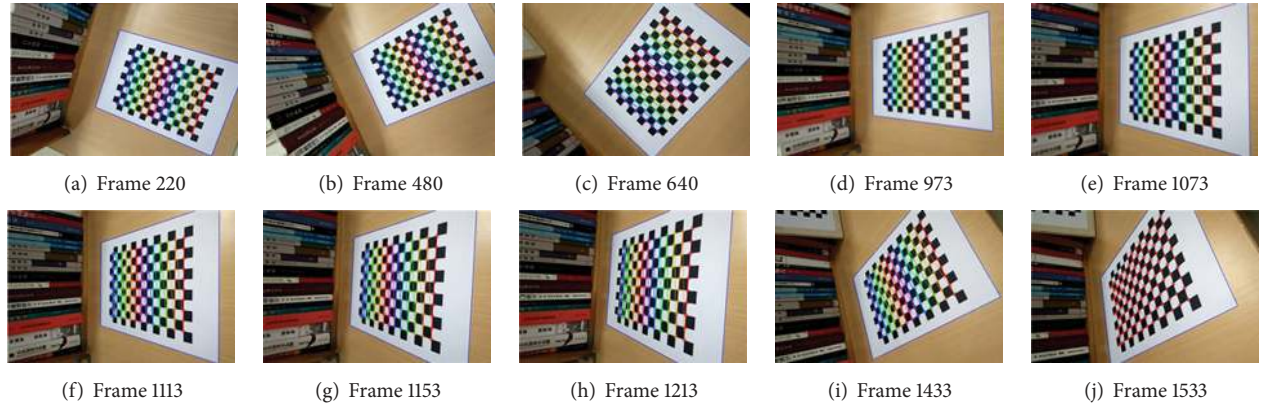


FIGURE 9: Homography tracking for the chessboard plane. Projected model lines using the homography of the proposed method are drawn in blue. The interior corners extracted in the images are drawn as color circles with the color lines between them denoting the order. The first row shows that the corners can be extracted successfully when the angle between the model plane and the image plane is small. However, when the angle is increasing, the corner detection will be less precise and even fail, as can be seen in the second row. For the whole sequence, the proposed method can provide good match with the image edges.

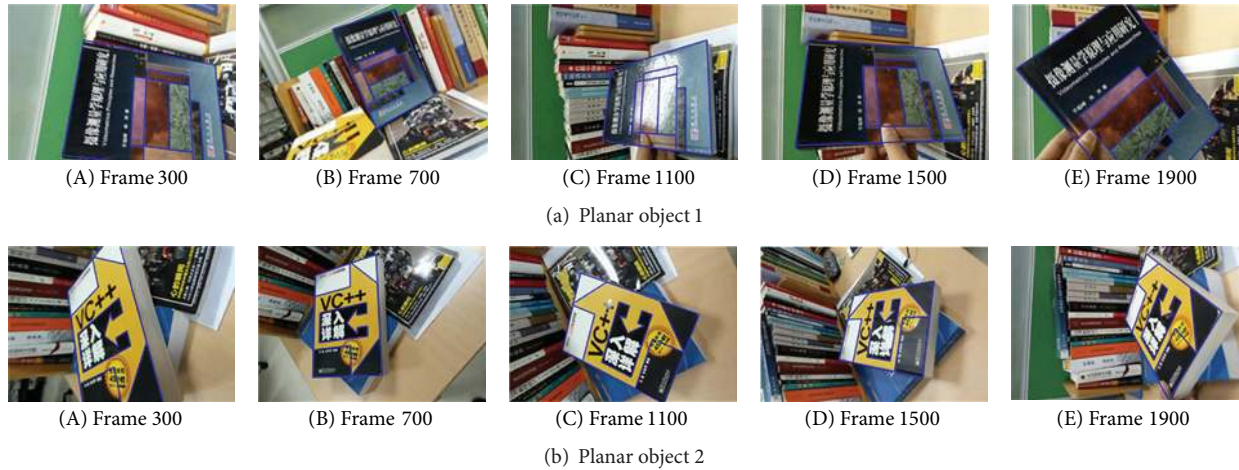


FIGURE 10: Homography tracking for the book covers in clutter environment.

TABLE 1: Calibration results with real data of 4 images.

Methods/pixels	f_x	f_y	c_x	c_y	Reprojection error
Corners based method	1147.23	1146.68	475.39	258.04	0.41
Lines based method	1150.76	1151.49	474.61	262.60	0.42
4-line based method	1141.14	1139.99	480.25	253.07	0.67

handy items is known, the objects can be reconstructed with the metric information.

6.2.4. Discussion. In practice, the corners detection often suffers from a failure, when the angle between the model and image plane is large or when some of the corners are invisible or corrupted by the image noise and blur. However,

the edge detection is more stable in such case. Moreover, in the simulated experiments, since the line segment is fitted by the corners lying on it, the proposed method provides almost the same performance with the corners based method. In our homography tracking framework, much more image edge points corresponding to the sample model points are utilized, and therefore the line segment can be fitted with higher accuracy. In addition, the proposed method is more flexible and suitable for the general user of the smartphone who wants to take the vision task, since it only uses the common and handy planar object rather than the prepared planar pattern.

7. Conclusions

In this paper, we have investigated the possibility of camera calibration using common and handy planar objects undertaking general motion for the smartphone vision tasks. A linear algorithm supported by the edge model based



FIGURE 11: Four images of model plane for camera calibration.

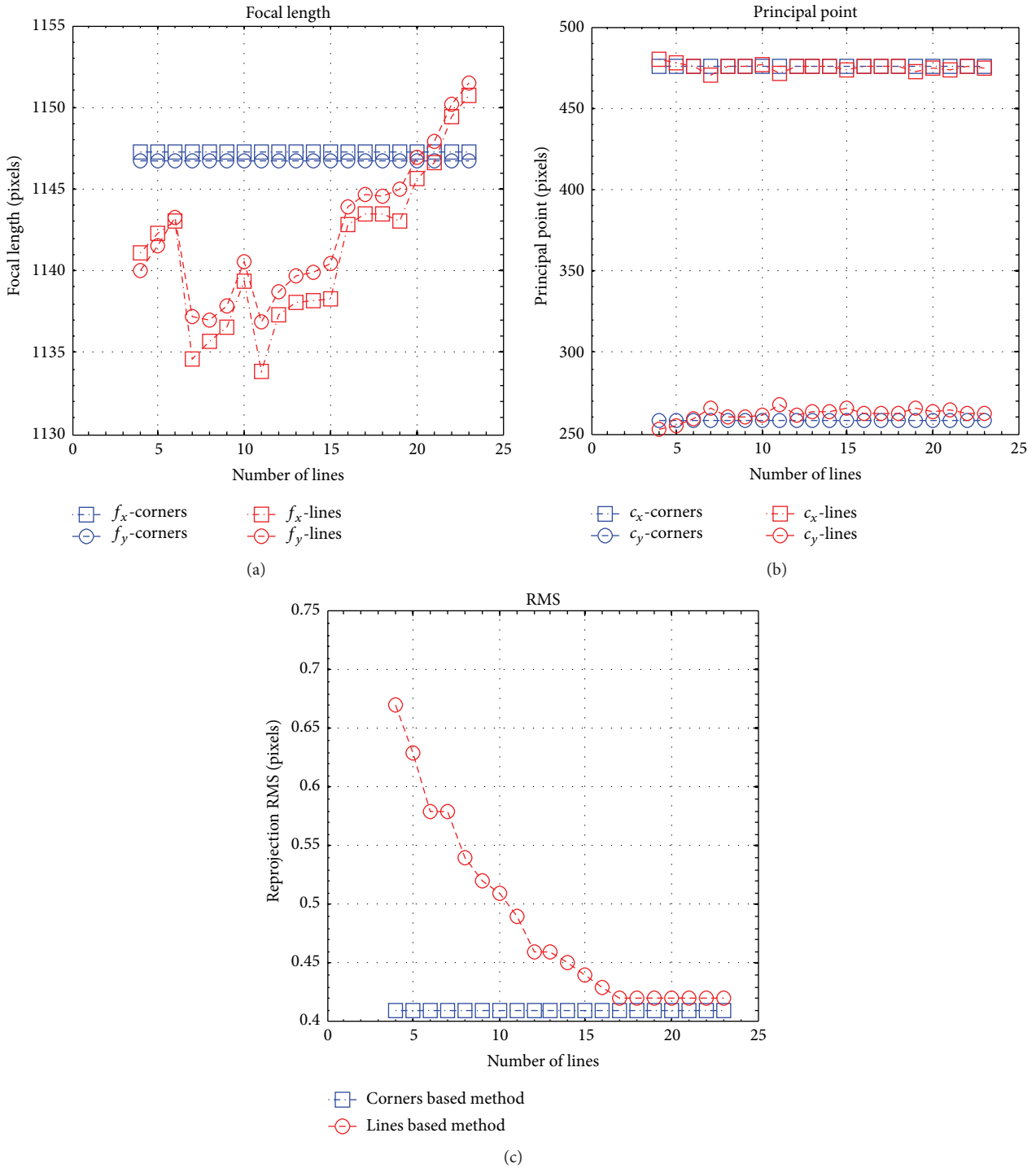


FIGURE 12: Results versus the number of the model lines.

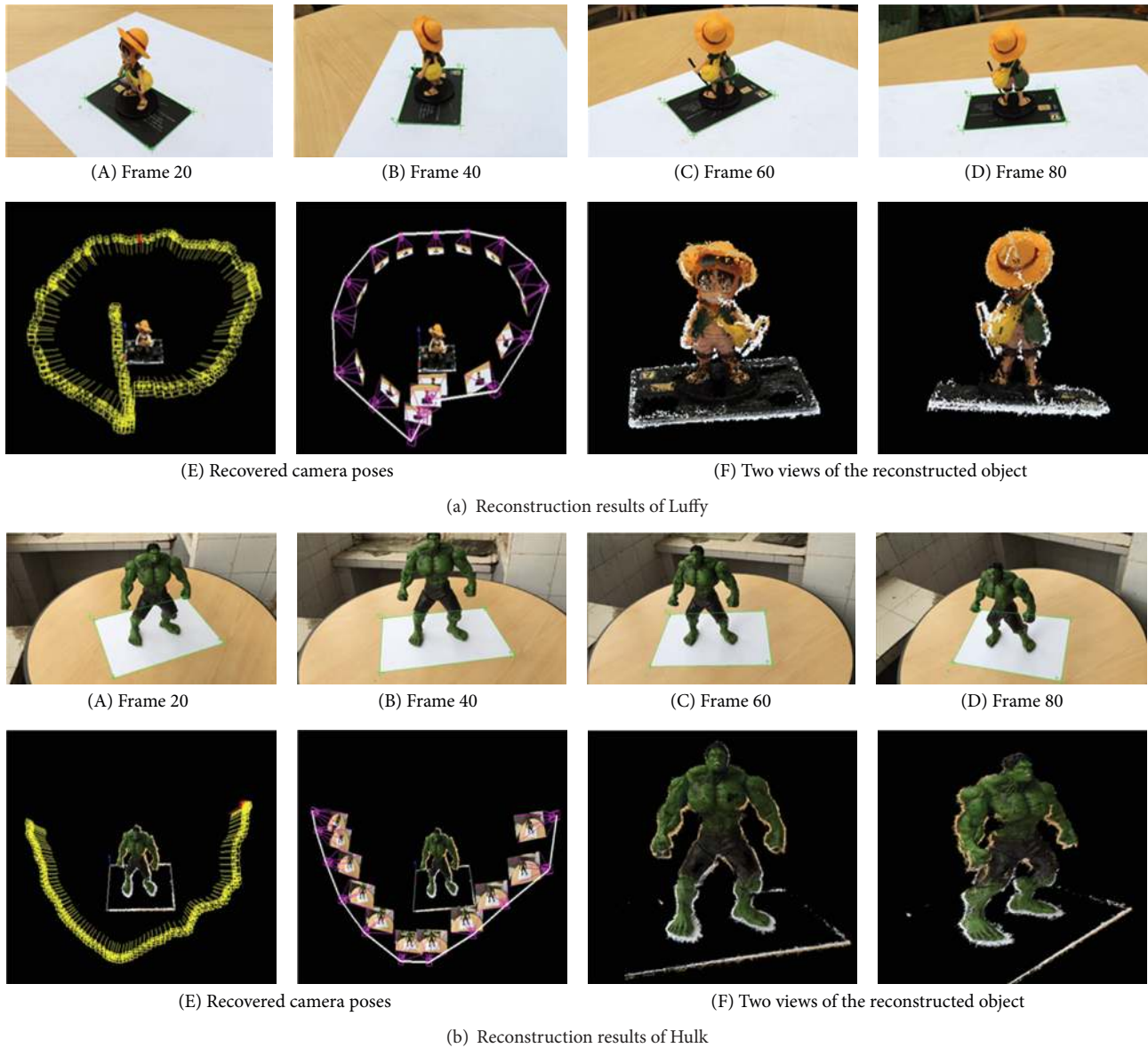


FIGURE 13: Results of image-based modelling. The green lines are the projections of the four edges of the calibration items by using the recovered camera parameters. The green crosses are the projections of the model points. We can see that the proposed method provides good matches of the projections of the model lines with the observed positions in the images.

homography tracking is proposed, followed by a nonlinear technique to refine the results. Both the computer simulated and real images have been utilized to validate the proposed algorithm. The experimental results exhibited that the proposed algorithm is valid and robust and provides more flexible performance than the widely used planar pattern based method.

In addition, for the general user who will do vision task, the prepared planar calibration may be not always in hand. However, the common items in our daily life almost have the standard size and planar structure. By exploiting the edge information, we proposed an easier and practical camera calibration method. Moreover, in the proposed method, the uncertainty of the image line segment was encoded in the

error model, which takes the finite nature of the observations into account. The problem of camera calibration using lines was formalized in the probabilistic framework and solved by the maximum likelihood approach.

Conflict of Interests

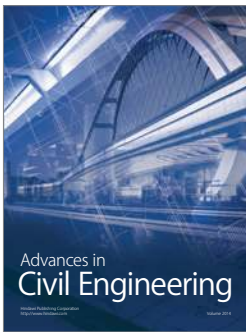
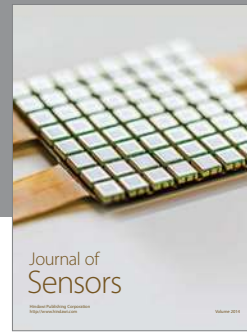
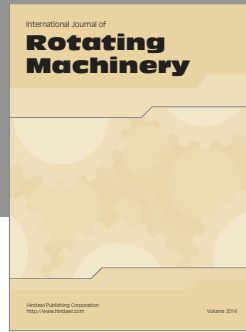
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research was supported by the National Basic Research Program of China (2013CB733100) and National Natural Science Foundation of China Grant (no. 11332012).

References

- [1] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.
- [2] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.
- [3] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [4] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [5] E. Peng and L. Li, "Camera calibration using one-dimensional information and its applications in both controlled and uncontrolled environments," *Pattern Recognition*, vol. 43, no. 3, pp. 1188–1198, 2010.
- [6] F. C. Wu, Z. Y. Hu, and H. J. Zhu, "Camera calibration with moving one-dimensional objects," *Pattern Recognition*, vol. 38, no. 5, pp. 755–765, 2005.
- [7] P. Hammarstedt, P. Sturm, and A. Heyden, "Degenerate cases and closed-form solutions for camera calibration with one-dimensional objects," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 317–324, Beijing, China, October 2005.
- [8] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, 2004.
- [9] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: theory and experiments," in *Computer Vision—ECCV'92: Second European Conference on Computer Vision Santa Margherita Ligure, Italy, May 19–22, 1992 Proceedings*, vol. 588 of *Lecture Notes in Computer Science*, pp. 321–334, Springer, Berlin, Germany, 1992.
- [10] Q.-T. Luong and O. D. Faugeras, "Self-calibration of a moving camera from point correspondences and fundamental matrices," *International Journal of Computer Vision*, vol. 22, no. 3, pp. 261–289, 1997.
- [11] R. I. Hartley, "Algorithm for self calibration from several views," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pp. 908–912, June 1994.
- [12] G. P. Stein, "Accurate internal camera calibration using rotation, with analysis of sources of error," in *Proceedings of the 5th International Conference on Computer Vision (ICCV '95)*, pp. 230–236, Cambridge, Mass, USA, June 1995.
- [13] R. I. Hartley, "Self-calibration from multiple views with a rotating camera," in *Proceedings of the European Conference on Computer Vision (ECCV '94)*, pp. 471–478, Stockholm, Sweden, May 1994.
- [14] C. Ricolfe-Viala and A.-J. Sánchez-Salmerón, "Robust metric calibration of non-linear camera lens distortion," *Pattern Recognition*, vol. 43, no. 4, pp. 1688–1699, 2010.
- [15] A. Petit, E. Marchand, and K. Kanani, "Combining complementary edge, keypoint and color features in model-based tracking for highly dynamic scenes," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 4115–4120, IEEE, Hong Kong, May 2014.
- [16] C. Choi and H. I. Christensen, "Real-time 3D model-based tracking using edge and keypoint features for robotic manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 4048–4055, Anchorage, Alaska, USA, May 2010.
- [17] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, 2006.
- [18] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, 2002.
- [19] R. Hanek, N. Navab, and M. Appel, "Yet another method for pose estimation: a probabilistic approach using points, lines, and cylinders," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, IEEE, Fort Collins, Colo, USA, June 1999.
- [20] R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *CVGIP: Image Understanding*, vol. 60, no. 3, pp. 313–342, 1994.
- [21] Intel, Open Source Computer Vision Library, 2014.
- [22] Z. Langming, Z. Xiaohu, and G. Banglei, "A flexible method for multi-view point clouds alignment of small-size object," *Measurement*, vol. 58, pp. 115–129, 2014.
- [23] C. Wu, "Towards linear-time incremental structure from motion," in *Proceedings of the International Conference on 3D Vision (3DV '13)*, pp. 127–134, Seattle, Wash, USA, July 2013.
- [24] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

