Self-Calibration of a Rotating Camera With a Translational Offset

Qiang Ji, *Member, IEEE,* and Songtao Dai

***Abstract*—Camera self calibration, based on a purely rotational movement of the camera, receives the most attention among different camera self-calibration methods due to its algorithmic simplicity. The existing purely rotational methods, however, assume camera rotates around its optical center, therefore yielding no translation offset. This assumption is not realistic, since in practice, the precise location of the optical center is often unknown, and the rotation is often performed about an unknown but fixed point near the optical center. The conventional methods tend to ignore the offset, and therefore, could lead to significant errors with the estimated camera parameters. In this paper, we introduce a new rotation-based camera self-calibration method, which explicitly accounts for the unknown translation offset. To this end, the problem is mathematically formulated and solved for differently taking the translation into consideration. To obtain the camera parameters with unknown camera rotations, our algorithm requires the camera to rotate around an unknown but fixed axis twice, by the same yet unknown angle. This is not an unreasonable assumption for precalibrating a camera on an active head. Experiments with both synthetic and real data show that the systematic errors caused by ignoring the translational offset will be effectively eliminated by our approach.**

***Index Terms*—Camera calibration, homographic matrix, pure camera rotation, self-calibration.**

I. Introduction

CAMERA calibration is a fundamental issue in computer vision and photogrammetry, especially for three-dimensional (3-D) computer vision and robotics applications. Camera calibration is defined as the determination of internal sensor parameters such as focal length, pixel skew, and principal point [1]. Camera calibration permits prediction of where in the image a world point will appear [2] and allows extracting 3-D metric information and recovery of the Euclidean structure of the scene from 2-D images [3], [4]. Although there are techniques inferring 3-D information about scene from uncalibrated cameras, effective camera calibration procedures open up the possibility of using a wide range of existing algorithms for 3-D reconstruction and recognition, all relying on the knowledge of the camera parameters [5].

There are two basic methods of calibrating cameras in computer vision. The traditional one is to use an object with *a priori* known Euclidean dimensions, the calibration pattern, in order to find the relation between world coordinates and image coordinates from which the internal parameters of the camera can be extracted. Another approach developed quite recently is called camera self calibration, which requires that the camera undergo a movement [6]. These methods require only 2-D point matches in several views obtained during either known [7]–[9] or unknown [6], [10], [11] camera motions, without *a priori* 3-D knowledge of the calibration scene [12]. Camera self calibration is important, especially in circumstances where the execution of the underlying visual tasks does not permit the use of reference objects [13].

There are three basic approaches in performing self calibration of cameras: 1) general camera motion approach with either limited or arbitrary number of views [10], [6], [14]–[22]; 2) pure translation approach [9]; and 3) pure rotation approach [7], [8], [23], [24]. According to the first approach, the camera undergoes general motion (rotation and translation) to obtain images of a scene from different viewpoints. Calibration is then accomplished using the matched 2-D points from different images. First introduced by Maybank and Faugeras [10], [6], their calibration method makes use of the epipolar structure of image pairs. Given three pairs of views, the set of six resultant quadratic equations can be solved by using homotopy continuation [10], [25] to obtain camera parameters.

However, the method initially presented by Faugeras *et al.* requires extreme accuracy of computation [23], [26] and is computationally expensive and sensitive to noise [12]. Furthermore, in order to use the epipolar structure, this method requires that the two views be taken from different viewpoints and images must first undergo a complex rectification process [27]. The applicability of these methods is thus complicated by the problem of finding matched points in images taken from different viewpoints, because of occlusion, aspect changes and lighting changes that inevitably occur when the camera moves. In many cases, all the images of a scene are taken from the same point in space, and the epipolar structure does not exist, so Maybank and Faugeras' method does not apply [23]. Finally, their method only uses two images, therefore susceptible to input perturbations. The use of more images by more recent techniques [17], [28], [21] leads to significant performance improvement.

In order to overcome the problems inherited in the Maybank and Faugeras method, several other self-calibration methods have been developed thereafter. Many of them require camera to undergo certain particular types of motion and utilize the special motions for camera calibration. This knowledge simplifies the calibration task enormously and yields a numerically more stable

Manuscript received June 6, 2002; revised November 29, 2002. This paper was recommended for publication by Associate Editor F. Chaumette and Editor S. Hutchinson upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001.

Q. Ji is with the Department of Electrical, Computer, and System Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: qji@ecse.rpi.edu).

S. Dai was with the Department of Computer Science, University of Nevada, Reno, NV 89557 USA.

Digital Object Identifier 10.1109/TRA.2003.820921

1042-296X/04$20.00 © 2004 IEEE

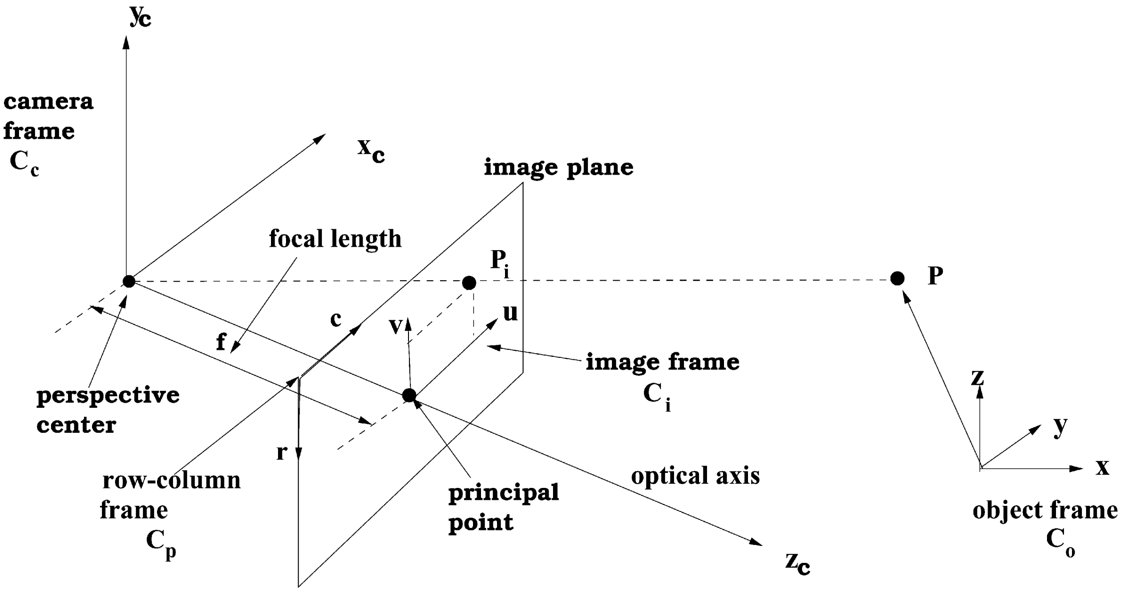

Fig. 1. Camera and perspective projection geometry.

and robust solution. Two most common approaches of camera self calibration are purely translational [9] and purely rotational approaches [7], [8], [23] with fixed camera parameters and the latest pure rotational approaches [29]–[31] that allow varying camera parameters. Among various approaches developed so far, approaches based on pure camera rotation have been proved to be very feasible, simple in operation, and received the most attention. According to this approach, multiple images of a scene are obtained by making the camera undergo only rotational movement while remain fixed in the same viewpoint. Camera calibration is then performed using the images.

Unfortunately, most such work made certain *a priori* assumptions about the rotational movement. A basic assumption is that the rotational movement is *pure* or close to *pure* in the sense that no relative translational movement is introduced between two camera coordinate frames. To ensure this, camera must rotate around the optical center.[1] In many real situations, however, the optical center is not exactly known and rotations are usually carried out around some fixed spatial point located on a support facility. As a result, the camera optical center undergoes a relative translational movement after rotation and the rotation is not therefore *pure*. In fact, the translational movement could be relatively significant in comparison to the distance of object points to the camera for certain applications. The existing purely rotational methods tend to ignore any translational movement, and thus, significant errors could be introduced to the estimated camera parameters, when camera is close to the calibration scene [32], [33]. Hayman *et al.* analytically studied the errors introduced when the assumption of pure rotation about the camera's optic center is violated. To the first order linear approximation, they derived expressions that relate the errors of the estimated camera parameters to the translational offset. Through study with synthetic data, they showed that pure rotational methods are only sufficient for applications where the translational offset is small relative to the distance of the calibration target to the camera. Their conclusions are primarily arrived at for cases when the rotation arm is of fixed length, as is indeed the case with many cameras mounted on tripods or robotic heads. However, for applications where high accuracy with the camera parameters is required, the errors introduced by the translational offset cannot be ignored. Recent work by Li [34] intended to overcome this problem by solving the translational offsets. But his method requires the use of a special calibration pattern.

In this paper, we introduce a new rotation-based camera self calibration method that uses the same images as the conventional rotational methods yet explicitly accounts for the unknown but fixed relative translation between rotation center and the optical center. To this end, our algorithm requires the camera to rotate around an unknown but fixed axis twice, by the same yet unknown angle. Images resulted from such rotations are then used to compute the camera parameters. Our approach consists of two steps. We first solve for the infinite homography matrix with camera undergoing equal yet but unknown translations and rotations. This represents a novel part of this paper. The camera internal parameters are then solved for from the infinite homography matrix using existing techniques.

II. Mathematical Projection Models

Let P be a 3-D point and $(x \ y \ z)^t$ be the coordinates of P relative to the object coordinate frame C_o. Let (c, r) be the pixel coordinates of P relative to the row-column frame C_p. C_p is located at the upper left corner of the image with c axis pointing from left to right and r axis pointing downward. Fig. 1 depicts the pin-hole camera model and the associated coordinate frames.

Based on the theory of full perspective projection, the projection that relates (c, r) on the image plane to the corresponding 3-D point (x, y, z) in the object frame can be described by

$$\lambda \begin{pmatrix} c \\ r \\ 1 \end{pmatrix} = K[R \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \qquad (1)$$

[1] Rotation around a point is defined as rotation around any axis passing through the point.

where λ is a scalar, K is a 3×3 matrix containing the intrinsic camera parameters, and R and T are respectively the rotation matrix and translation vector that specify the relative pose of the object frame to the camera frame. K is parameterized as

$$K = \begin{pmatrix} fs_x & 0 & u_0 \\ 0 & fs_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

where f is camera focal length, s_x and s_y are the scale factors (pixels/mm). u_0 and v_0 are the coordinates of the principal point in pixels relative to C_p. Here we assume image skew and lens distortions are negligible. K is called intrinsic camera calibration matrix. Details on pinhole camera model equations may be found in [35]. The goal of camera calibration is to determine elements of K.

III. Theory of the Proposed Approach

A. Problem Statement

According to our approach, to calibrate a camera, the camera may rotate around a fixed point other than the initial camera optical center. The rotations yield several image views of a given scene. The purpose of this research is to develop a method that computes the matrix K from these images, given the fact that rotation is not around the optical center. In the sections to follow, we first discuss the theory for the more ideal pure rotation case, followed by a discussion of the theory for the case where rotation is around a fixed point that deviates from the camera optical center.

B. Theory for Pure Rotation

Here, we will first reformulate the purely rotational approach originally put forward by Hartley [23]. For a general camera motion, from one configuration, I, to another configuration, II, if an object point $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ in a scene appears in both image I and II, then the two image points (c_1, r_1) and (c_2, r_2) of the same object point (x, y, z) are thus determined by the equations

$$\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = K[R_1 \quad T_1] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2}$$

$$\lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = K[R_2 \quad T_2] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \tag{3}$$

In a purely rotational approach, the fixed point of rotation coincides with the camera optical center and remains the same before and after rotation. For convenience and without loss of generality, we may just choose the object-frame coordinate system such that its origin lies at the optical center of the camera. With this choice, the translations in (2) and (3) turn out to be zero, which will greatly simplify the calibration task as indicated below. As a result, (2) and (3) are reduced to

$$\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = KR_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{4}$$

$$\lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = KR_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{5}$$

By eliminating the coordinates of the object point (x, y, z) in this pair of equations, we obtain the following image-image transformation (mapping) as

$$\lambda_1 R_1^{-1} K^{-1} \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = \lambda_2 R_2^{-1} K^{-1} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}. \tag{6}$$

Introducing a relative rotation

$$R_1 = RR_2$$

(6) can be simplified as

$$\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = \frac{\lambda_2}{\lambda_1} KRK^{-1} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} \tag{7}$$

which serves as the key working equation for the purely rotational approach to camera self-calibration. For convenience, let us define

$$H = KRK^{-1} \quad \text{and} \quad \lambda = \frac{\lambda_1}{\lambda_2}.$$

With these two definitions, (7) can be rewritten as

$$\lambda \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}. \tag{8}$$

Here the parameter λ depends on the location of the object point. Matrix H is usually called the *infinite homography* in computer vision literature [18]. As a result, this equation indicates that the image-image mapping of each object point yields two independent linear equations in terms of the elements of H matrix. Given a minimum of 4 matched 2-D points, H can be solved up to a scale factor. The scale factor can subsequently be resolved by using the fact that $\det(H) = 1$. Hartley [23] introduces a linear and an nonlinear method to solve for H.

C. Solution to K

Given H, we can then solve for the intrinsic matrix K. The solutions to K depend on if the rotation matrix R is known or unknown. We discuss solution for each case.

1) R Is Known: If R is known, then K can be derived from a single H analytically. Here we introduce a linear solution similar to that of Stein's [24]. From $H = KRK^{-1}$, we have

$$HK = KR. \tag{9}$$

Equation (9) provides nine linear equations, involving the four unknown intrinsic parameters. Element of K can, therefore, be solved analytically via a system of linear equations. Given more images produced with additional rotations, we can produce more equations like (9) and stack them together to form a larger system of linear equations, based on which we can solve for elements of K. The solution is more accurate due to the redundancies provided by additional images.

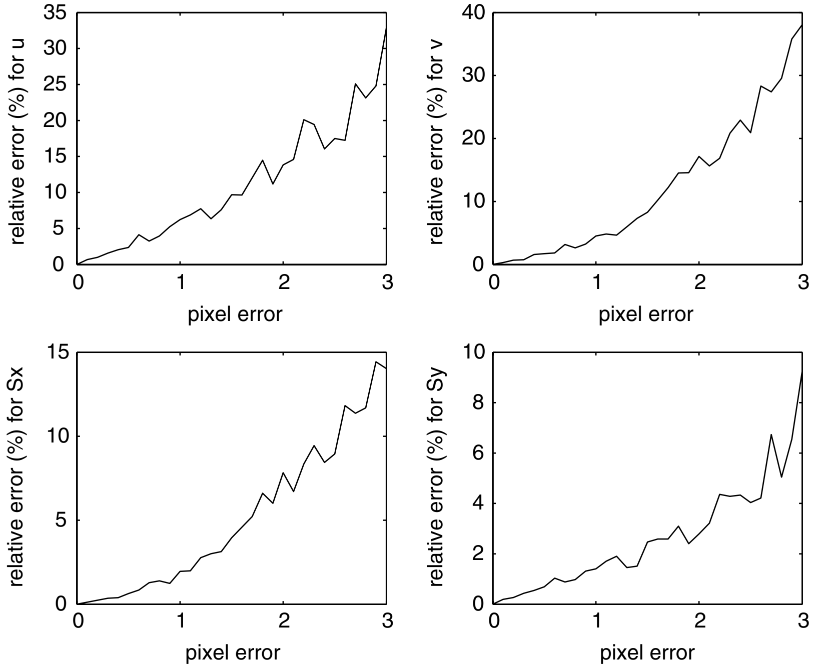

Fig. 2. Relative errors of the estimated camera parameters by the linear approach for pure rotation as a function of the image errors. 50 points are used.

2) R Is Unknown: Linear Solution: In practice, the rotation matrix R is often unknown. The above solution is not applicable. Here we introduced a linear solution similar to that of Hartley [23] to solve for the intrinsic matrix K. Let $C = KK^t$, then from $H = KRK^{-1}$ and using the orthonormality constraint of R, we have $HCH^t = C$. The solution to C, up to a scale factor, is formulated as a linear least-squares problem using a minimum of two H matrices. Solution to elements of C corresponds to the eigen vector of the smallest eigen value of the design matrix. The scale factor is solved by imposing the constraint that the last element of C (i.e., C(3, 3)) is 1.

Once C is solved for, Hartley proposed to use Cholesky Decomposition to linearly determine the camera calibration matrix with the constraint that the solution is an upper triangle matrix, assuming C is positive definite. We found this approach is difficult to impose the zero skew constraint. To impose this constraint, we explicitly encode it in the K matrix and obtain KK^t as

$$KK^t = \begin{pmatrix} f^2 s_x^2 + u^2 & uv & u \\ uv & f^2 s_y^2 + v^2 & v \\ u & v & 1 \end{pmatrix}.$$

Given C we can, therefore, obtain elements of K from

$$\begin{aligned} u &= C(3,1) \\ v &= C(3,2) \\ fs_x &= \sqrt{C(1,1) - u^2} \\ fs_y &= \sqrt{C(2,2) - v^2}. \end{aligned}$$

Our study, however, shows that the linear solution, though elegant, is sensitive to pixel errors as shown in Fig. 2. The figure shows all four parameters deviate considerably from their correct values when the error for each image point exceeds two pixels. This is especially the case for the parameters u and v, which are twice as sensitive to pixel errors as the two scale factors.

Moreover, the decomposition is numerically very unstable for a poorly conditioned C. One way of improving the sensitivity of the solution is to impose the zero-skew constraint while computing the matrix C as demonstrated in the work by De Agapito [31] and Zisserman [36]. Furthermore, the sensitivity and numerical instabilities problems can be further corrected using the bundle-adjustment method [37].

3) R Is Unknown: Nonlinear Solution: To further improve the solution to K, we introduce a one-step nonlinear approach, that computes K directly from H without computing C. The one-step nonlinear approach is found to produce more robust and accurate estimate.

Given H, the orthonormality of R can be utilized to form an equation that does not involves R. Since $H = KRK^{-1}$, we have $R^{-t} = K^T H^{-t} K^{-t}$. Thus, the application of the orthonormality constraint $(R = R^{-t})$ of R yields

$$(KK^t)H^{-t} = H(KK^t) \tag{10}$$

where

$$KK^t = \begin{pmatrix} s_x^2 f^2 + u_0^2 & u_0 v_0 & u_0 \\ u_0 v_0 & s_y^2 f^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{pmatrix}.$$

Equation (10) is similar to the Kruppa equations for the rotational camera. It links the image transformation to the camera internal parameters.

Now, we present a nonlinear method, similar to the one proposed by by De Agapito [29], to solve for K from H using (10), assuming R is unknown. Like the linear solution, our solution also needs two different H matrices to uniquely determine K. From (10), we define matrix A^i as

$$A^i = (KK^t)H_i^{-t} - H_i(KK^t) \quad \text{where i} = 1, 2 \tag{11}$$

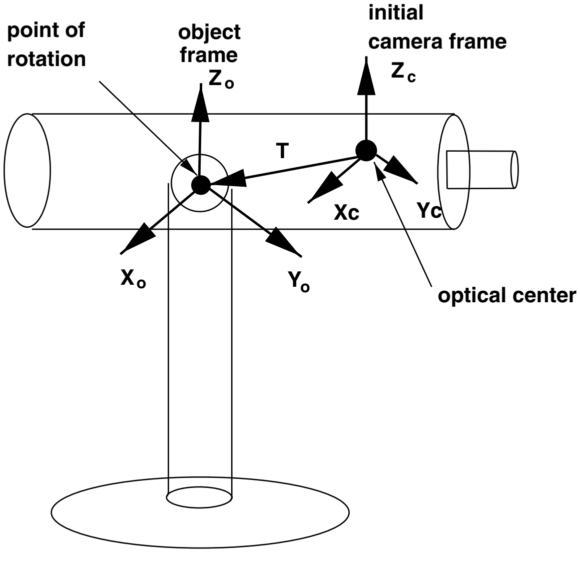

Fig. 3. Object frame and the initial camera frame, along with the optical center and the fixed point of rotation.

then we define the criterion function F to minimize as

$$F(K) = \sum_{i=1}^{2} A^{i\,2}_{00} + A^{i\,2}_{01} + A^{i\,2}_{02} + A^{i\,2}_{10} + A^{i\,2}_{11} + A^{i\,2}_{12} + A^{i\,2}_{20} + A^{i\,2}_{21} + A^{i\,2}_{22} \quad (12)$$

where A^i_{jk} is the (j, k) entry of the matrix A^i. Since H has been calculated in the previous steps, F is a function of four intrinsic parameters $(s_x f, s_y f, u_0, v_0)$. The four intrinsic parameters which make function F minimum are the solutions to (12).

Matlab function *lsqcurvefit()* is used to implement the nonlinear minimization problem. The initial guess for the nonlinear solution is obtained from linear solution introduced in Sections III-B and III-C that solves for K from H. Given a sequence of images, the estimate for H can be further refined using the conventional bundle-adjustment algorithm [37].

D. Camera Rotation With Fixed Translation

In many real-world environments, it is very difficult to perform pure camera rotations due to lack of knowledge about the location of camera optical center. Camera rotations are usually carried out around some fixed spatial point located on the camera support facility. This, therefore, yields a relative translation between the two camera frames before and after rotation. In this section, we mathematically reformulate the problem taking the relative translation into consideration. Our formulation is more general in that the pure rotation can be treated as a special case.

Without loss of generality, let us assume that the object frame is located at the fixed point of rotation and is aligned with the initial camera frame as shown in Fig. 3.

It is observed that, although both panning and tilting are allowed during the rotation of the camera, the point of rotation is fixed. This leads to an important fact: the location of the fixed point of rotation relative to the camera frame (either before or after) remains unchanged, i.e., $T_1 = T_2 = T$. This fact turns out to be a key point for our approach. As a result, (2) and (3) remain true, i.e.,

$$\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = K[R_1 \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (13)$$

$$\lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = K[R_2 \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (14)$$

In the case of pure rotation, we can choose T to be zero, then this set of equations is exactly the same as that of pure ideal rotation case. In the case of rotation with fixed translation, T can not be zero, since we will have different optical centers after each rotation. This greatly complicates the image-image mapping and the set of linear equations in terms of the entries of the H matrix such as those for pure rotation are no longer available.

Fortunately, in spite of its complication, (13) and (14) do have a potentially useful feature: fixed translation T. In order to make use of this valuable feature, we can take into account three different camera system configurations, I, II, and III, which correspond to three different images of a given object point and are determined by

$$\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} = K[R_1 \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (15)$$

$$\lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = K[R_2 \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (16)$$

$$\lambda_3 \begin{pmatrix} c_3 \\ r_3 \\ 1 \end{pmatrix} = K[R_3 \quad T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (17)$$

From (15) and (16), after eliminating the translation T, we obtain

$$\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} - \lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = K(R_1 - R_2) \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \quad (18)$$

Similarly, from (16) and (17), we obtain

$$\lambda_3 \begin{pmatrix} c_3 \\ r_3 \\ 1 \end{pmatrix} - \lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = K(R_3 - R_2) \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \quad (19)$$

Equations (18) and (19) can be rewritten as after placing K to the left side of the equations

$$K^{-1} \left(\lambda_1 \begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix} - \lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} \right) = (R_1 - R_2) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (20)$$

$$K^{-1} \left(\lambda_3 \begin{pmatrix} c_3 \\ r_3 \\ 1 \end{pmatrix} - \lambda_2 \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} \right) = (R_3 - R_2) \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \quad (21)$$

By introducing two relative rotations ${}_1^2R$ and ${}_2^3R$

$$R_2 = {}_1^2R R_1$$
$$R_3 = {}_2^3R R_2.$$

Thus, (20) and (21) take the form

$$K^{-1}\left(\lambda_1\begin{pmatrix}c_1\\r_1\\1\end{pmatrix} - \lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}\right) = \left({}_1^2R^t - I\right)R_2\begin{pmatrix}x\\y\\z\end{pmatrix} \tag{22}$$

$$K^{-1}\left(\lambda_3\begin{pmatrix}c_3\\r_3\\1\end{pmatrix} - \lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}\right) = \left({}_2^3R - I\right)R_2\begin{pmatrix}x\\y\\z\end{pmatrix}. \tag{23}$$

In many applications such as precalibrating a camera on an active head, the rotation of a camera can be accurately controlled. For the purpose of camera calibration, it is thus reasonable and convenient to choose the three orientations of the camera such that the following conditions are satisfied:

$${}_2^3R = {}_1^2R = R^t. \tag{24}$$

Physically, this simplification means the relative orientation between frames 1–2 is the same as the the relative orientation between frame 3 and frame 2. In other words, given frame 1 and frame 2, frame 3 can be obtained by rotating the camera again by the same rotation as the one between frame 1 to frame 2. With this simplification, (22) and (23) reduce to

$$K^{-1}\left(\lambda_1\begin{pmatrix}c_1\\r_1\\1\end{pmatrix} - \lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}\right) = -R(R^t - I)R_2\begin{pmatrix}x\\y\\z\end{pmatrix} \tag{25}$$

$$K^{-1}\left(\lambda_3\begin{pmatrix}c_3\\r_3\\1\end{pmatrix} - \lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}\right) = (R^t - I)R_2\begin{pmatrix}x\\y\\z\end{pmatrix}. \tag{26}$$

Equating (25) and (26) by removing $(R^t - I)R_2\begin{pmatrix}x\\y\\z\end{pmatrix}$ yields

$$\lambda_1\begin{pmatrix}c_1\\r_1\\1\end{pmatrix} - \lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix} = KRK^{-1}\left[\lambda_2\begin{pmatrix}c_2\\r_2\\1\end{pmatrix} - \lambda_3\begin{pmatrix}c_3\\r_3\\1\end{pmatrix}\right]. \tag{27}$$

Therefore, we obtain a key working equation, which is similar to (7) in purely rotational approach. The difference lies in that (7) is derived from two image frames and contains two unknown scalars (λ_1, λ_2), while (27) is derived from three image frames and thus contains three unknown scalars $\lambda_1, \lambda_2, \lambda_3$. In the remaining part of this section, we will present our solution to this working equation.

Equation (27) shows that for each matched object point, we have two additional independent parameters, i.e., (λ_1/λ_2) and (λ_3/λ_2), and three equations. By eliminating these two additional parameters, we can obtain one equation for each matched point. Let

$$H = KRK^{-1}$$
$$\lambda_{12} = \frac{\lambda_1}{\lambda_2}$$
$$\lambda_{32} = \frac{\lambda_3}{\lambda_2}.$$

Then (27) can be written as

$$\lambda_{12}\begin{pmatrix}c_1\\r_1\\1\end{pmatrix} - \begin{pmatrix}c_2\\r_2\\1\end{pmatrix} = H\left[\begin{pmatrix}c_2\\r_2\\1\end{pmatrix} - \lambda_{32}\begin{pmatrix}c_3\\r_3\\1\end{pmatrix}\right]. \tag{28}$$

With the use of the notation for $H = \begin{pmatrix}H_1^t\\H_2^t\\H_3^t\end{pmatrix}$, where H_i^t represents the ith row of matrix H, (28) thus leads to the following three equations:

$$\lambda_{12}c_1 - c_2 = H_1^t\begin{pmatrix}c_2 - \lambda_{32}c_3\\r_2 - \lambda_{32}r_3\\1 - \lambda_{32}\end{pmatrix} \tag{29}$$

$$\lambda_{12}r_1 - r_2 = H_2^t\begin{pmatrix}c_2 - \lambda_{32}c_3\\r_2 - \lambda_{32}r_3\\1 - \lambda_{32}\end{pmatrix} \tag{30}$$

$$\lambda_{12} - 1 = H_3^t\begin{pmatrix}c_2 - \lambda_{32}c_3\\r_2 - \lambda_{32}r_3\\1 - \lambda_{32}\end{pmatrix}. \tag{31}$$

From (29) and (31), after eliminating the parameter λ_{12}, we have

$$c_1 - c_2 = (H_1^t - c_1H_3^t)\begin{pmatrix}c_2 - \lambda_{32}c_3\\r_2 - \lambda_{32}r_3\\1 - \lambda_{32}\end{pmatrix}. \tag{32}$$

Similarly, from (30) and (31) , we have

$$r_1 - r_2 = \left(H_2^t - r_1H_3^t\right)\begin{pmatrix}c_2 - \lambda_{32}c_3\\r_2 - \lambda_{32}r_3\\1 - \lambda_{32}\end{pmatrix}. \tag{33}$$

In (32) and (33), only one object-point dependent scalar parameter λ_{32} is left, which can be expressed, from (32), as

$$\lambda_{32} = \frac{c_2 - c_1 + (H_1^t - c_1H_3^t)\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}}{(H_1^t - c_1H_3^t)\begin{pmatrix}c_3\\r_3\\1\end{pmatrix}}. \tag{34}$$

Similarly, from (33), we can express λ_{32} as

$$\lambda_{32} = \frac{r_2 - r_1 + (H_2^t - r_1H_3^t)\begin{pmatrix}c_2\\r_2\\1\end{pmatrix}}{(H_2^t - r_1H_3^t)\begin{pmatrix}c_3\\r_3\\1\end{pmatrix}}. \tag{35}$$

Combining (34) and (35) and eliminating λ_{32} yield an nonlinear equation in terms of the H matrix elements as

$$\frac{c_2 - c_1 + (H_1^t - c_1 H_3^t)\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}{(H_1^t - c_1 H_3^t)\begin{pmatrix} c_3 \\ r_3 \\ 1 \end{pmatrix}} = \frac{r_2 - r_1 + (H_2^t - r_1 H_3^t)\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}{(H_2^t - r_1 H_3^t)\begin{pmatrix} c_3 \\ r_3 \\ 1 \end{pmatrix}}. \qquad (36)$$

Therefore, for each object point tracked, we obtain one nonlinear equation in terms of the H matrix elements only. Given sufficient number of points, H can be solved for numerically. Specifically, Let U and V represent the left and right sides of (36), then for each matched point in three views, we have ideally $U = V$.

If N matched points are chosen, we have $U_i = V_i$ for $i = 1, 2 \ldots, N$. The algebraic error criterion function f can therefore be established as

$$f(H) = \sum_{i=1}^{N} (U_i - V_i)^2. \qquad (37)$$

Given a minimum of eight (i.e., $N \geq 8$) points, H can be found up to a scale factor by minimizing (37). This minimization is implemented using the MATLAB function *lsqcurvefit()*. The initial estimate for the nonlinear solution was provided using the linear technique discussed in Sections III-B and III-C , assuming translation offset is zero. Our study using synthetic data showed that the H matrix can be relatively accurately estimated given the initial estimate about ±15% off its true value. The scale factor for H was subsequently solved for using the fact that the determinant of H is 1. Given H, the solutions introduced in Section III-C can be used to solve for K.

E. Degeneracy in Camera Self Calibration

Studies by Sturm [20] have identified certain types of camera motion, often referred to as *critical motion*, lead to degeneracy in camera self-calibration solution. Degeneracy means the existence of multiple solutions. A complete characterization of critical motion sequences for constant intrinsic camera parameters may be found in [20]. In particular, they have identified some degenerate motion sequences that often occur in practice such as rotation round the parallel (or the same) axis or planar motion, etc. They proved that for a camera with fixed intrinsic camera parameters, any camera motion resulted from rotation around the same axis with arbitrary translation is critical, therefore leading to multiple solutions. This seems to indicate that the motion sequence used for our technique is critical as well since it involves rotation about the same axis with fixed translation. Further study by Kahl *et al.* [38] and Zisserman *et al.* [36], however, shows that rotation about the fixed axis may not always be critical, depending on the constraints imposed on the camera parameters. For example, Kahl proved that if skew is assumed to be zero, then for constant camera parameters, only rotation about axes $(0, *, *)$ and $(*, 0, *)$ is critical, where '*' denotes an arbitrary real number. With assumption of only zero skew, our rotation is not critical if we can avoid rotating about these critical axes. In fact, this can be accomplished by first rotating about the pan axis (two equal rotations, three images) followed by rotating about the tilt axis (again, two equal rotations, three images). This provides two infinite homographies, no longer about a single axis, to be input into the second stage of the algorithm. This is also supported by our experiments since we can uniquely determine the four intrinsic camera parameters by rotating camera around an axis other than the critical axes mentioned above.

IV. Experiments

In this section, we present results of experiments that were aimed at characterizing the performance of our self calibration algorithm. The calibration algorithm was first tested on synthetic data to determine its performance under varying amount of noise, different numbers of points, and different exterior camera orientations. The algorithm is then applied to a set of real images to recover the intrinsic camera parameters. Finally, we present results from a comparison of our approach with an existing method that ignores the translation offset.

A. Tests With Synthetic Data

In the experiments with synthetic data, the 3-D data (3-D points) are generated randomly within certain specified ranges. 2-D data (image coordinates) are generated by projecting the 3-D data onto the image plane using fixed intrinsic camera parameters. To study the performance of our method under different camera parameters, three sets of ideal camera parameters are used, $(fs_x, fs_y, u_0, v_0) = (1306, 1306, 160, 120), (1306, 1206, 160, 120)$, and $(263, 263, 157, 127)$, respectively. Given the first image, the second image is generated by rotating the camera around an axis by θ degrees while the third image is generated by rotating the camera again around the same axis by θ degrees. The default translational offset is set at 20% for most experiments. The 2-D image data are subsequently corrupted by independently and identically distributed (i.i.d.) Gaussian noise of mean zero and standard deviation σ. The noise level is quantified using σ. σ varies from 0–4.0 pixels, although a realistic noise level is only about 1.5 pixels [23]. From the generated 2-D/3-D data, we estimate the intrinsic camera parameters (fs_x, fs_y, u_0, v_0) using our algorithm. The performance is judged by the absolute relative errors between the estimated parameters and their true values under various conditions as discussed below.

1) Influence of the Number of Matched Object Points and Image Noise: Here, we summarize the performance of our self calibration approach under varying number of image points and varying amount of image noise. In each case, the size of the synthetic image is 320 × 240 and the rotations are around two general axes (0.2, 0.5, 0.59) and (0.8, 0.5, 0.33). In the first experiment, we fixed the image noise at 1.5 pixel standard deviation

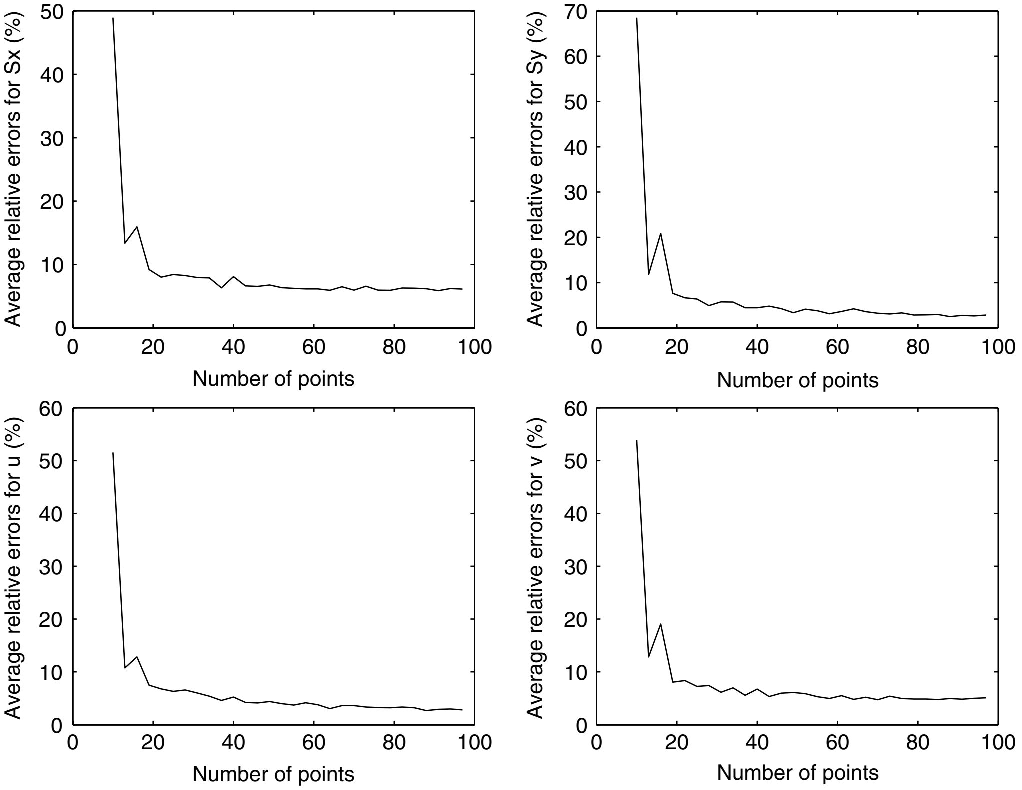

Fig. 4. Impact of number of points on the accuracy of the estimated camera parameters.

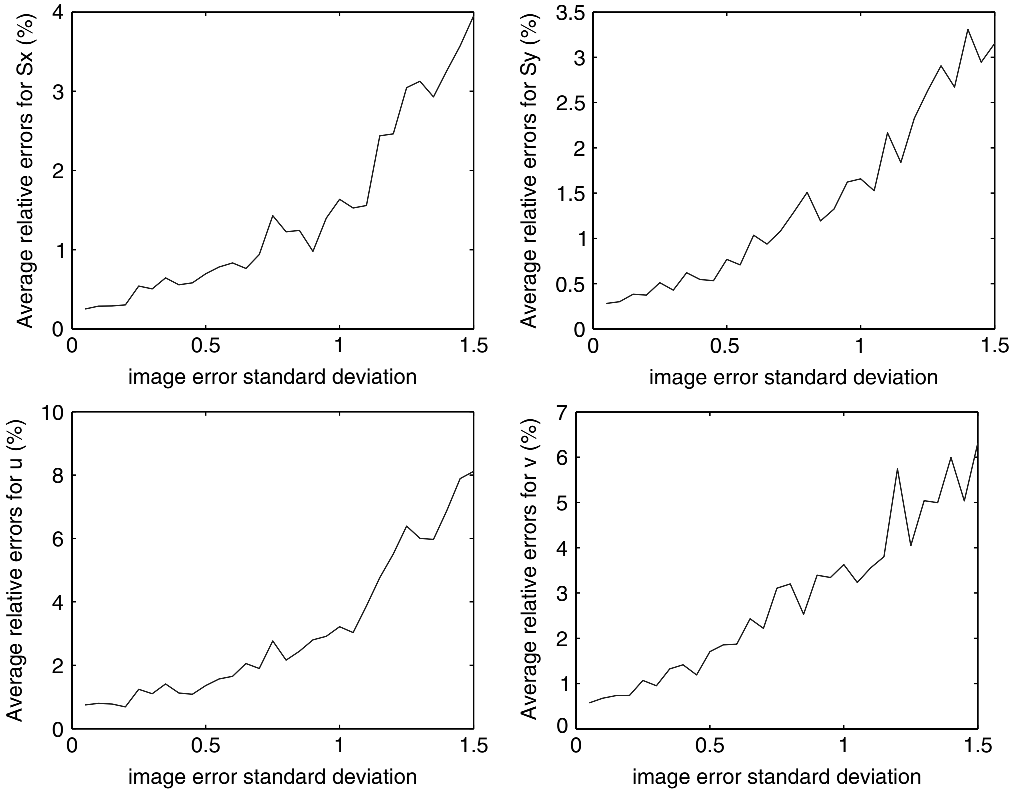

Fig. 5. Impact of image noise level on the accuracy of the estimated camera parameters.

and gradually vary the number of points. The results are summarized in Fig. 4, which plots the relative errors of the estimated camera parameters as a function of the number of points participating in the computation. In the second experiment, we fix the number of points at 75 points and vary the image noise level. The results are summarized in Fig. 5, which plots the relative errors of the estimated camera parameters as a function of the image noise level. For each case, the initial guess of the intrinsic

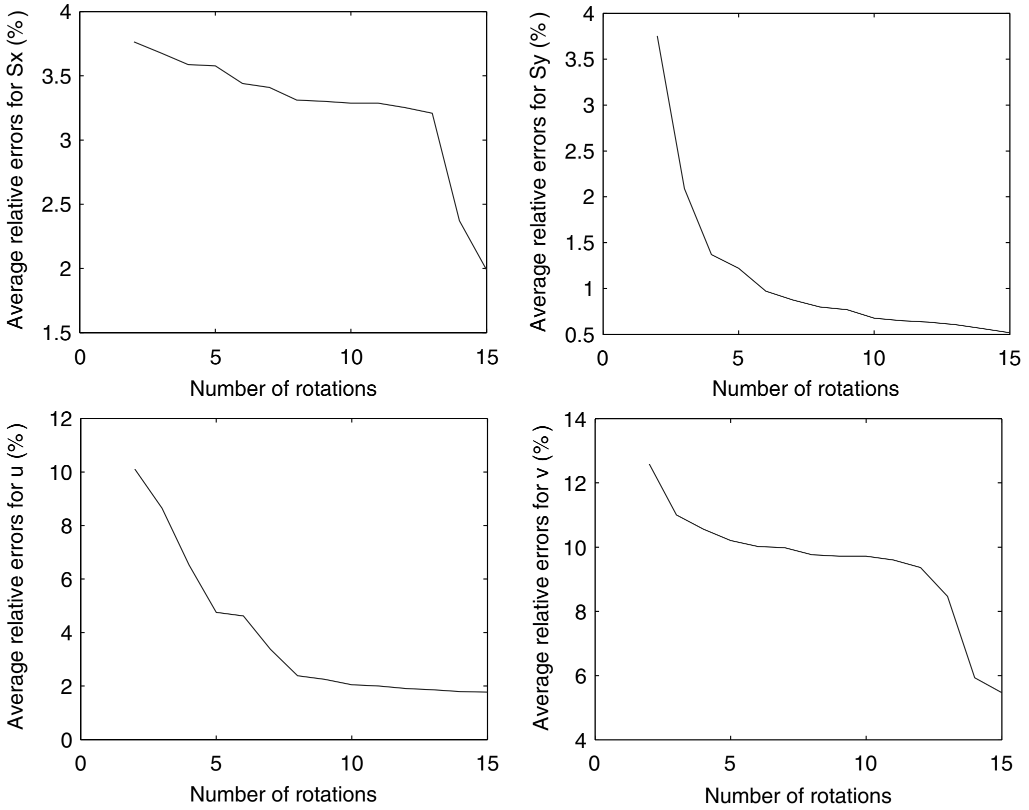

Fig. 6. Relative estimation errors versus the number of images used.

camera parameters are 20% deviated from their true values and each point in the figure represents the average of 100 trials under the same noise level.

It is clear from the two figures that increasing the number of points can significantly improve the tolerance of our algorithm to noise. The algorithm is rather unstable using only 10 points. However, with 20 points or more, the algorithm can tolerate rather high noise level $(\sigma = 1.5)$, which is close to the realistic noise level encountered with real image data. When the number of points increases to 100, our algorithm shows a marked improvement. It can generate very accurate results for all four parameters even under noise level up to two-pixel standard deviation.

2) Accuracy Versus the Number of Images: Here, we study the accuracy of our method as a function of the number of images (rotations) used. Theoretically, we should input as many images from a long sequence as possible to any self-calibration problem, and find the solution using all data simultaneously. This could improve the robustness and accuracy of the method. But in practice, it is difficult to acquire many images, partially due to occlusion. To avoid this problem, we use three images (two rotations) to compute a homography. We then acquire another three images under different rotations to compute another homography. This repeats until we collect a set of homography matrices, which are then stacked together and used simultaneously to solve for the intrinsic camera parameters. For the experiments we did, we acquired up to 15 homography matrices and studied the accuracy of our method as a function of the number of homography matrices used. The results are summarized in Fig. 6, where the number of 3-D points are used was 100 and the noise level of each pixel was at 0.8. Each point in the figure represents the average of 100 trails. We can conclude that as the number of images used increases, the estimation errors decrease. For some camera parameters (e.g., u and Sy), this decrease, however, begins to taper off when the number of rotations reaches certain level (e.g., 10).

3) Robustness Versus Initial Guesses: For our nonlinear solution, an initial guess of the intrinsic camera parameters is required for further calculations. In order to test the robustness of our calibration algorithm, in case of the number of matched object points 100, we chose two different initial estimates which are about 10% and 20% deviated from the true intrinsic values. These results are plotted in Fig. 7, which plots the relative errors of the estimated camera parameters under varying amount of image noise, with two different initial guesses. It is clear that the algorithm, in spite of different initial guesses, follows a very close solution even under very large noise levels. The robustness of our self-calibration algorithm is thus clearly demonstrated.

4) Robustness in Terms of the Camera Translation Offset: One major feature of our algorithm is that the translation offset (distance between point rotation and the initial camera optical center) is fixed but unknown. Here, we studied two quite different camera translation offset ratios[2] (small translation ratio 0.2 and large translation ratio 0.5) in the case of a number of points equal to 100. We achieved similar results for all four parameters. The result is shown in

[2] The ratio of the translational offset to the distance of the calibration scene to the camera.

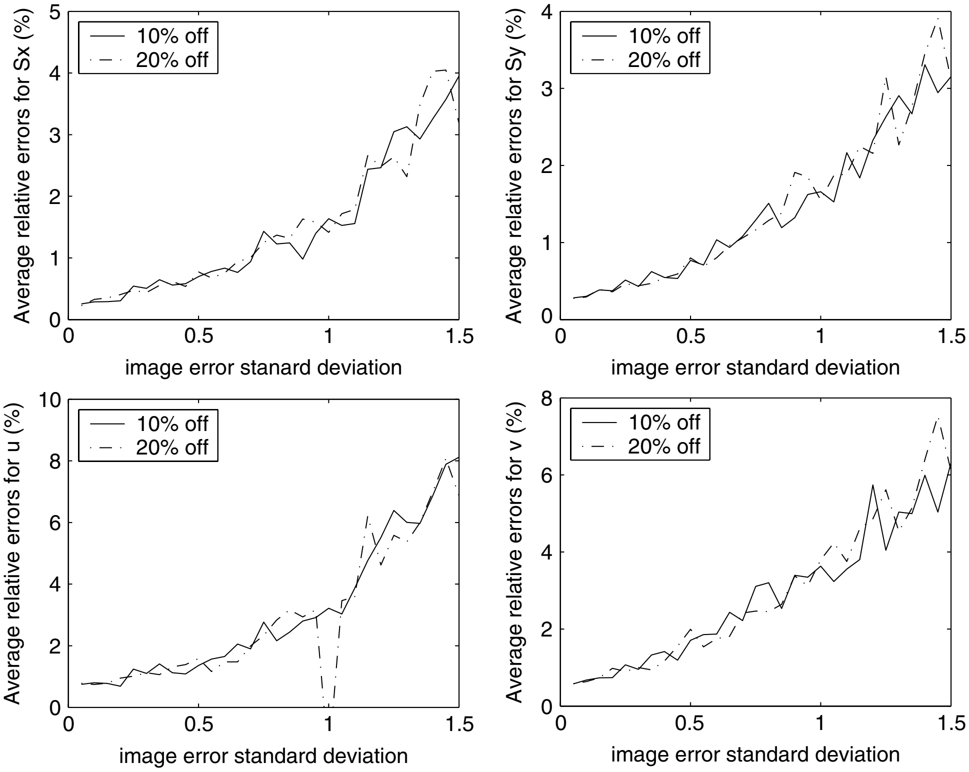

Fig. 7. Impact of initial guess on the accuracy of the estimated camera parameters under different image noise levels and with two different initial guesses.

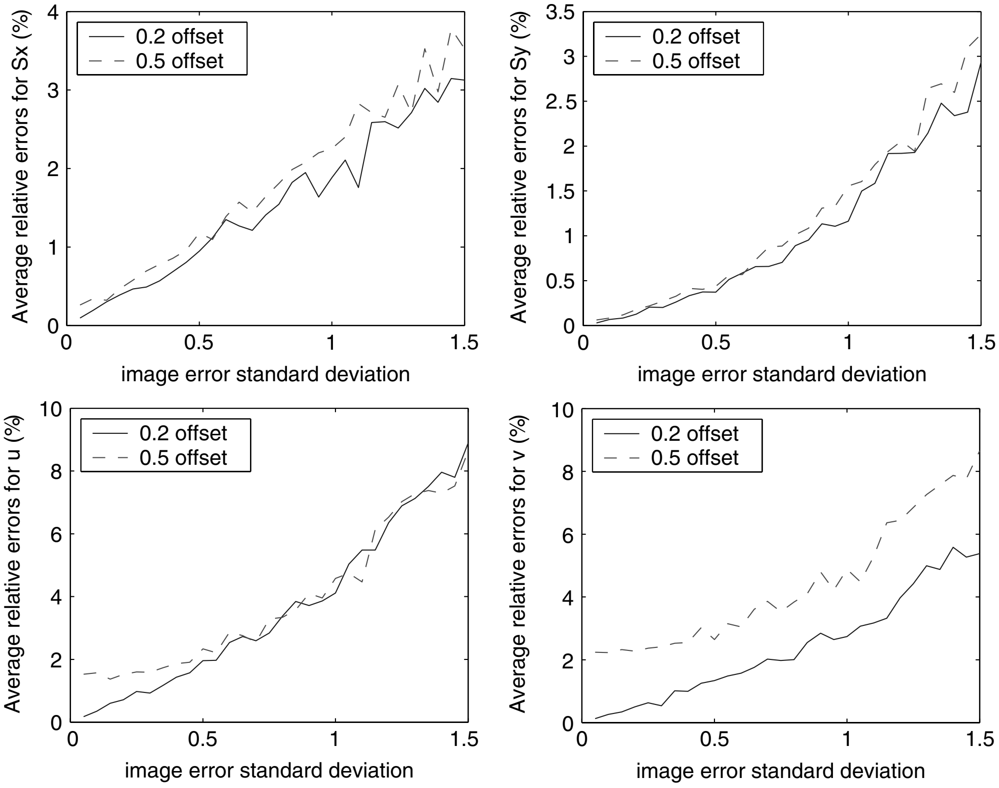

Fig. 8. Impact of different camera translation offsets on the accuracy of the estimated camera parameters.

Fig. 8. It is clear that the two estimations follow each other closely. But the one with larger offset ratio tends to produce larger estimation errors. This implies that for a real image, the point of rotation should not be too far from the camera optical center, which is often the case in practice. The larger estimation errors of our method under large translation offset and noise may partially attribute to the poor initial estimates provided by the linear methods.

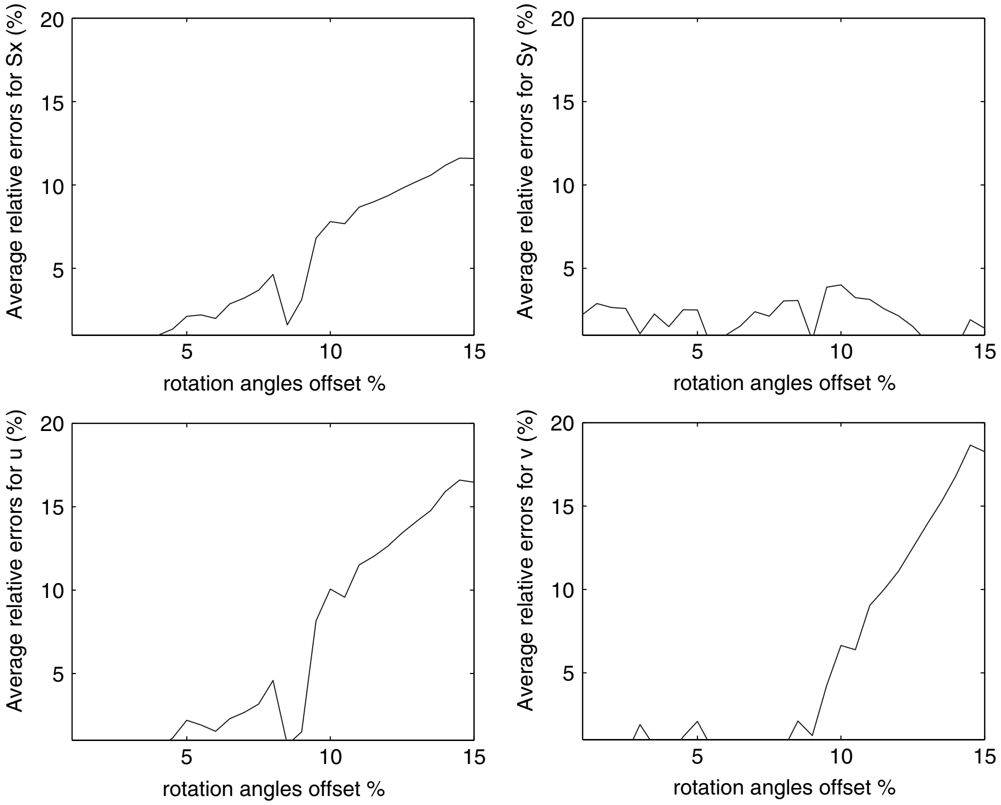

Fig. 9. The impact of inequality of two rotations on the accuracy of the estimated camera parameters.

5) Camera Estimation Accuracy Versus Inequality in the Rotations: In practice, it is difficult to ensure the two consecutive rotations are exactly equal. It is expected that there are minor discrepancies between the two rotations. Here, we study the sensitivity of our approach to a minor inequality of the two rotations. The inequality is expressed in terms of the relative difference between the two rotation angles. The results are plotted in Fig. 9. For this experiment, the number points used was 100 and the noise level was fixed at 0.8 pixel standard deviation.

From Fig. 9, the relative rotation angle difference ranges from 1% to 15%, and the maximum relative errors of the estimated intrinsic parameters remain below 20%. This demonstrates this our algorithm can tolerate modest inequality between two consecutive rotations. This is important for practical purposes.

B. Comparison With Conventional Pure Rotational Approach

In this section, we perform a quantitative comparison of our approach with a pure rotational self-calibration approach that does not account for the translation offset. Due to lack of ground-truth data with real images and the difficulty of obtaining different translation offsets, the performance comparison is undertaken using synthetic data. The pure rotational approach we chose to compare with is the linear approach introduced in Section III-B, with zero skew imposed.

To produce the image data, the camera is rotated around a general axis $((1/\sqrt{3}),(1/\sqrt{3}),(1/\sqrt{3}))$ passing through the point (0, 0, 0) for 5, 10, and 15 degrees, respectively, to produce the needed three images. The images are subsequently perturbed with a noise level of 0.5. The average distance (T_z) between camera and the 3-D points are fixed at 4 m. The offset (distance between the initial camera optical center and the point of rotation) varies to produce the data. Fig. 10 shows the errors of the estimated intrinsic camera parameters as a function of the offset ratios. It is clear that the proposed method produces rather consistent and accurate results (small errors) even under larger offset (ratio $=$ 1.0) for all four parameters. It basically eliminates the errors resulting from rotating the camera around a point other than the camera optical center. On the other hand, the conventional pure rotational approach becomes unstable even under a very small translation offset ratio (0.1). We see a significant increase in estimation error with the conventional approach as the offset ratio increases. Nevertheless, when the offset is very small, e.g., less than 0.05, the conventional approach can produce reasonable results. This echoes the conclusion from Hayman's study [33]. This explains why the pure rotational approach works reasonably well for some applications where the offset percentage is negligible. This study demonstrates that our approach is much better for applications where the offset percentage is unknown but is expected to be large. Our study, however, also reveals that our method becomes numerically unstable as the offset continues to increase. This may be either due to the highly nonlinearity nature of our method or to the poor initial estimates provided by the linear method under a large translation offset.

C. Tests With Real Images

The ultimate test of our algorithm is its performance on real images. Here, we present the results from real image data. A charge-coupled device (CCD) camera was used in our experiments, which was mounted on a support, and the rotations were

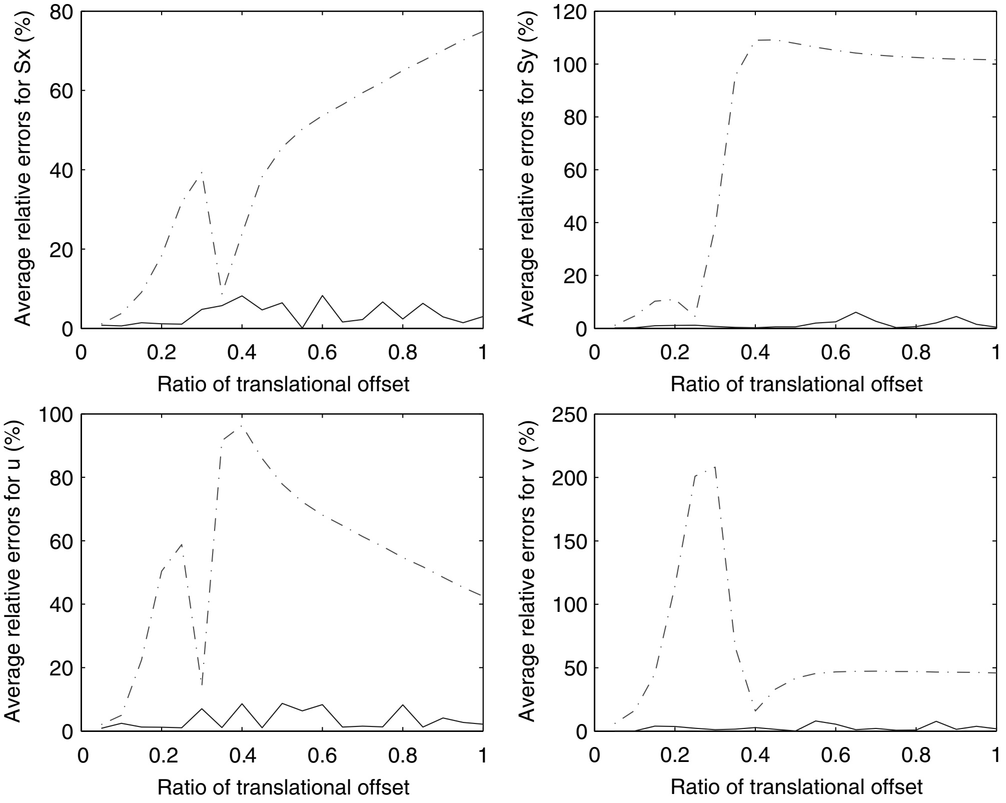

Fig. 10. Performance comparison of the proposed method and a conventional pure rotation method in estimating the four intrinsic camera parameters as a function of translational offset ratios. The solid lines represent our method, while the dotted lines the linear pure rotational methods.

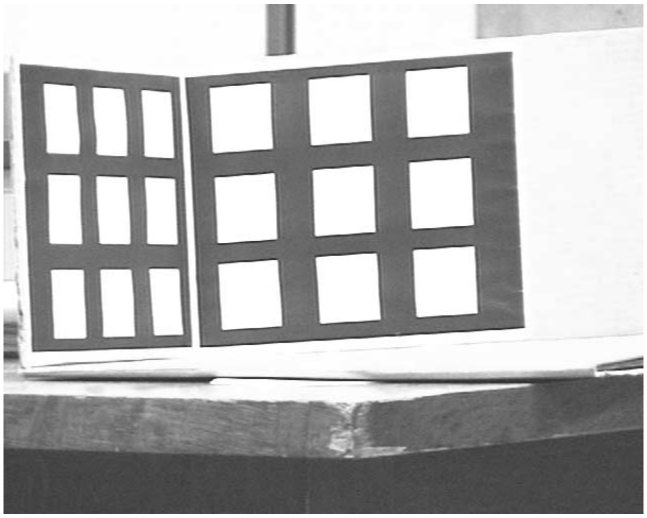

Fig. 11. First image of the calibration pattern.

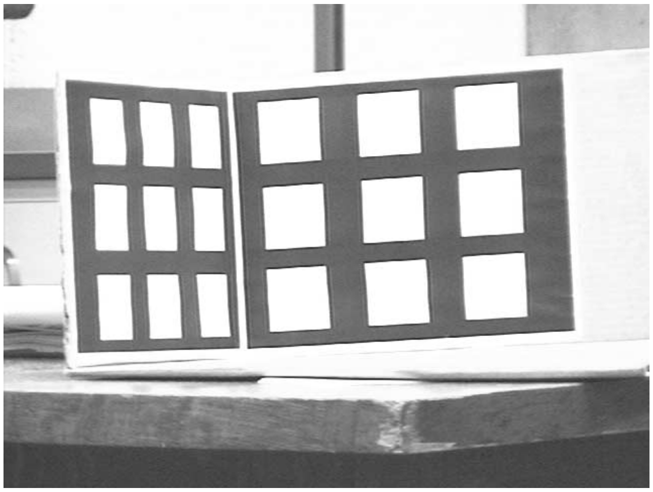

Fig. 12. Second image of the calibration pattern after the first camera rotation.

performed by panning and tilting the camera around the point of support. Images were taken of a calibration metric, which is located about 1.2 m away from the camera. Sample images of the calibration pattern are shown in Figs. 11, 12, and 13, respectively. Points matching are done manually. The image size is 320 × 240. The calibration results were summarized in Table I and compared with the calibration results obtained using a conventional camera calibration procedure [39]. The results from the pure rotation approach that ignores the offset is also included in the table. It it clear that the calibration results from our approach and the traditional approach are in quite satisfactory agreement, while the results from the pure rotation approach deviate significantly from the results of the traditional approach. The errors with the pure rotational approach are due to ignoring the translation offset resulting from camera rotation.

V. Summary and Discussion

In this paper, we introduce a more practical rotation approach to camera self calibration. Our approach consists of two steps. We first solve for the infinite homography matrix with the camera undergoing equal but unknown translations and rotations. This represents a novel part of this paper. The camera

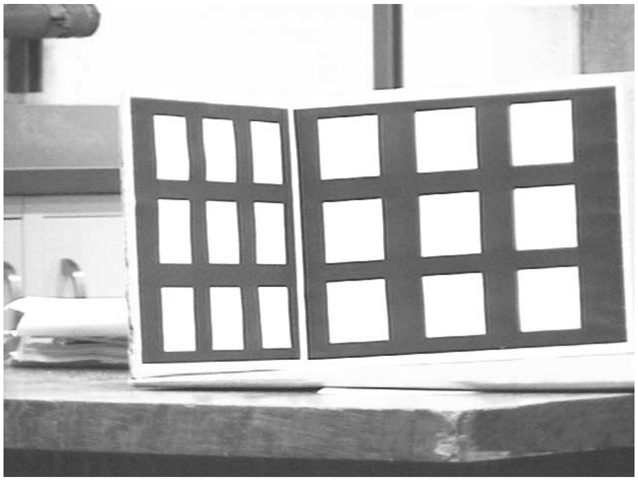

Fig. 13. Third image of the calibration pattern after rotating the camera twice.

TABLE I
CALIBRATION RESULTS OF OUR METHOD USING IMAGES OBTAINED BY ROTATING THE CAMERA AROUND THE POINT OF SUPPORT

	Conventional method	our method	pure rotation
fs_x	263	268	295
u_0	157	164	130
v_0	126	127	150

internal parameters are then solved for from the infinite homography matrix using existing techniques. Experiments based on synthetic data demonstrate the robustness and accuracy of our approach under different conditions such as different noise levels, different number of points, different initial estimates, and different translation offsets. Compared with the existing methods that ignore rotation offset, ours is especially effective when the offset is large. Our study also reveals that given a sufficient number of points, our method can tolerate (in terms of numerical stability and estimation accuracy) error up to three pixels, much higher than the 1.5 pixel errors in most images. Our experiments with real images show that our approach, in the presence of translation offset, produces estimates comparable with those obtained using the traditional camera calibration method. This provides further justification of our self-calibration algorithm. As pointed out throughout the paper, the camera estimates can be further refined using the bundle adjustment method, initialized using our methods. It can be used to refine the initial projective homographies, and also in the final Euclidean step.

While our approach is similar to that of Stein's [24], our technique is, however, more practical, since our method does not need the knowledge of the rotations, which are often difficult to acquire accurately. Yet, our method can utilize the rotations, if known, to improve the estimation accuracy. On the other hand, for applications where the rotation matrices can be accurately estimated from an apparatus such as the shaft encoder of an active head, the two techniques are comparable.

The major assumption that underlies our algorithm is that the two rotations must be about the same axis and with equal angles. We realize this assumption limits the practical utilities of our approach. Our study indicates that our algorithm can tolerate minor inequality between the two rotations (up to 15) and still produces reasonably accurate results. Moreover, this assumption can be realized for certain applications, such as photogrammetry and some robotics applications, with the aid of auxiliary equipment. In particular, for precalibrating a camera on an active head that can pan and tilt, it is not an unreasonable assumption to make.

REFERENCES

[1] R. I. Hartley and J. L. Mundy, "The relationship between photogrammetry and computer vision," presented at the *Int. Society for Optical Engineers Conf.*, 1993.

[2] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[3] Z. Y. Zhang. (1996) Self-Maintaining Camera Calibration in Time. [Online]. Available: http://www.sop.inria.fr/robotvis/personnel/vubs/calib/calib.html

[4] O. Faugeras, "Stratification of three-dimensional vision: Projective, affine and metric representations," *J. Opt. Soc. Amer.*, vol. 12, pp. 465–484, 1995.

[5] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[6] S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 123–151, 1992.

[7] A. Basu, "Active calibration: Alternative strategy and analysis," in *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, 1993, pp. 495–500.

[8] F. Du and M. Brady, "Self-calibration of the intrinsic parameters of cameras for active vision systems," in *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, 1993, pp. 477–482.

[9] L. Dron, "Dynamic camera self-calibration from controlled motion sequences," in *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, 1993, pp. 501–506.

[10] O. D. Faugeras, O. T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Proc. ECCV'92, Lecture Notes in Computer Science*, vol. 588, 1992, pp. 321–334.

[11] Q. T. Luong and O. D. Faugeras, "Self-Calibration of a Stereo Rig from Unknown Camera Motions and Point Correspondences," INRIA, Sophia Antipolis, France, Tech. Rep. RR-2014, 1993.

[12] Q. T. Luong, "An optimization framework for efficient self-calibration and motion determination," in *Proc. Int. Conf. Pattern Recognition*, 1994, pp. 248–252.

[13] S. D. Ma, "A self-calibration technique for active vision system," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 114–120, Feb. 1996.

[14] R. I. Hartley, "Euclidean reconstruction from uncalibrated views," in *Applications of Invariance in Computer Vision*, J. Mundy, A. Zisserman, and D. Forsyth, Eds. New York: Springer-Verlag, 1994, pp. 237–256. LNCS 825.

[15] M. Armstrong, A. Zisserman, and R. Hartley, "Self-calibration from image triplets," in *Proc. 4th Eur. Conf. Computer Vision*, vol. 1, 1996, pp. 3–16.

[16] C. Wiles and M. Brady, "Ground plane motion camera models," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. 238–247.

[17] A. Heyden and K. Astrom, "Euclidean reconstruction from constant intrinsic parameters," in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 1996, pp. 339–343.

[18] M. Pollefeys and L. Van Gool, "Stratified self-calibration with the modulus constraint," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 707–724, Aug. 1999.

[19] O. Faugeras, L. Quan, and P. Sturm, "Self-calibration of a 1-D projective camera and its application to the self-calibration of a 2-D projective camera," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1179–1185, Oct. 2000.

[20] P. Sturm, "Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction," in *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, 1997, pp. 1100–1105.

[21] B. Triggs, "Autocalibration and the absolute quadric," in *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, 1997, pp. 609–614.

[22] J.-E. Ha, J.-Y. Yang, K.-J. Yoon, and I.-S. Kweon, "Self-calibration using the linear projective reconstruction," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, Apr. 2000, pp. 885–890.

[23] R. Hartley, "Self-calibration of stationary cameras," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 5–23, 1996.

[24] G. P. Stein, “Accurate internal camera calibration using rotation, with analysis of sources of error,” in *Proc. Int. Conf. Computer Vision*, 1995, pp. 230–236.

[25] Q. T. Luong, “Matrice fondamentale et calibration visuelle sur l’environmenta,” Ph.D. dissertation, Univ. Paris-Sud, Centre D’Orsay, Paris, France, 1992.

[26] P. Sturm, “A case against Kruppa’s equations for camera self-calibration,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1199–1204, Oct. 2000.

[27] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” *Machine Vis. Applicat.*, vol. 12, pp. 16–22, 2000.

[28] M. Pollefeys, “Self-calibration and metric 3-D reconstruction from uncalibrated image sequences,” Ph.D. dissertation, Catholic Univ. Leuven, Leuven, Belgium, 1999.

[29] L. de Agapito, E. Hayman, and I. D. Reid, “Self calibration of a rotating camera with varying intrinsic parameters,” in *Proc. British Machine Vision Conf.*, 1998, pp. 105–114.

[30] Y. Seo and K. S. Hong, “Auto-calibration of a rotating and zooming camera,” presented at the *IAPR Workshop Machine Vision Application*, 1998.

[31] L. de Agapito, R. I. Hartley, and E. Hayman, “Linear self-calibration of a rotating and zooming camera,” *Proc. IEEE Conf. Computer Vision, Pattern Recognition*, vol. 1, p. 21, June 1999.

[32] L. Wang, S. B. Kang, H. Y. Shum, and G. Xu, “Error analysis of pure rotation-based self-calibration,” in *Proc. Int. Conf. Computer Vision (ICCV)*, 2001, pp. 464–471.

[33] E. Hayman and D. W. Murray, “The effects of translational misalignment in the self-calibration of rotating and zooming cameras,” Univ. Oxford, Oxford, U.K., Tech. Rep. OUEL 2250/02, 2002.

[34] M. Li, “Kinematic calibration of an active head-eye system,” *IEEE Trans. Robot. Automat.*, vol. 14, pp. 153–158, Feb. 1998.

[35] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.

[36] A. Zisserman, D. Liebowitz, and M. Armstrong, “Resolving ambiguities in auto-calibration,” *Philos. Trans. Roy. Soc. London*, ser. A, vol. 356, no. 1740, pp. 1193–1211, 1998.

[37] B. Triggs, A. Zisserman, and R. Szeliski, “Vision algorithms: Theory and practice,” presented at the *Workshop on Vision Algorithms, 7th IEEE Int. Conf. Computer Vision*, 1999.

[38] F. Kahl, B. Triggs, and K. Astrom, “Critical motions for auto-calibration when some intrinsic parameters can vary,” *J. Math. Imaging, Vision*, vol. 13, no. 2, pp. 131–146, 2000.

[39] Z. Y. Zhang. (1998) A Flexible New Technique for Camera Calibration, Tech. Rep. MSR-TR-98-71. [Online]. Available: http://research.microsoft.com/zhang

Qiang Ji (S’92–M’98) received the M.S. degree in electrical engineering from the University of Arizona, Tucson, in 1993 and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, in 1998.

He is currently an Assistant Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. Previously, he was an Assistant Professor with the Department of Computer Science, University of Nevada at Reno. His research interests include computer vision, probabilistic reasoning under uncertainty using Bayesian networks, pattern recognition, information fusion, and robotics. He has published more than 40 papers in referred journals and conferences in these areas. His research has been funded by local and federal government agencies including the National Science Foundation (NSF), National Institutes of Health (NIH), the Air Force Office of Scientific Research (AFOSR), the Office of Naval Research (ONR), and the Army Research Office (ARO), and by private companies including Boeing and Honda.

Songtao Dai received the M.S. degree in computer science, and the Ph.D. degree in physics, in 2000 from the University of Nevada at Reno.