The following manuscript was published in

A camera self-calibration method suitable for
variant camera constraints

Chin-Hung Teng†, Yung-Sheng Chen‡, Wen-Hsing Hsu†,*

†Department of Electrical Engineering, National Tsing Hua University, Taiwan.
‡Department of Electrical Engineering, Yuan Ze University, Taiwan.

d897908@oz.nthu.edu.tw, eeyschen@ee.yzu.edu.tw, whhsu@ee.nthu.edu.tw.

This paper presents a self-calibration algorithm that seeks the camera intrinsic parameters to minimize the sum of squared distances between the measured and reprojected image points. By exploiting the constraint provided by fundamental matrix, the function to be minimized can be directly reduced to a function of the camera intrinsic parameters, thus variant camera constraints such as fixed or varying focal length can be easily imposed by controlling the parameters of the resulting function. We employed the simplex method to minimize the resulting function and tested the proposed algorithm on some simulated and real data. The experimental results demonstrate that our algorithm performs well for variant camera constraints as well as for two-view and multiple-view cases. © 2005 Optical Society of America

OCIS codes: 150.6910, 150.5670, 040.1490.
1. Introduction

Camera self-calibration is a long-lasting research topic in computer vision. In the past decades, many approaches to camera self-calibration have been proposed in the literature. For instance, Faugeras et al.\textsuperscript{1,2} first put forward a method by utilizing the epipolar geometry and the so-called Kruppa equations to calibrate the camera. Hartley\textsuperscript{3} proposed a self-calibration algorithm by first establishing the projective reconstruction of the scene and then computing the calibration matrix and refining the results using bundle adjustment. Following these methods, more approaches to camera self-calibration were also developed. Triggs\textsuperscript{4} utilized the absolute quadric to calibrate the camera and obtained a metric 3D structure. Luong and Faugeras\textsuperscript{5} showed that only correspondences between three images and the fundamental matrices computed from these correspondences are sufficient to recover the internal and external parameters of the camera. Pollefeys\textsuperscript{6} employed a stratified method to achieve camera self-calibration and metric reconstruction. He developed a modulus constraint\textsuperscript{7} to recover the scene structure to affine stratum and then upgraded it to the metric stratum. More approaches were also reported in the literature and a complete analysis for camera self-calibration can be found in Ref. 8 and Ref. 9.

In early periods, most self-calibration algorithms assumed fixed camera intrinsic parameters during the capture of images. This assumption will restrict the application of calibration algorithm since zooming and focusing are prohibited by
this assumption. Recently, some researchers\textsuperscript{8,10–13} have discussed the possibility of relaxing camera constraints to allow varying and unknown internal camera parameters. For instance, Bougnoux\textsuperscript{11} proposed a self-calibration method that allows varying camera focal length. He derived a closed-form formula from Kruppa equation so that the focal length can be directly computed from the fundamental matrix, epipoles and principal points. He then utilized the obtained focal length as an initialization to further refine the results and recover 3D scene structure. This technique is of course more general than the traditional methods owing to the more general camera constraints. However, if image sequences were obtained under nearly constant camera intrinsic parameters, the traditional methods may yield a better performance owing to the additional constraint of fixed camera internal parameters. Therefore, an algorithm which can deal with various types of camera constraints according to practical applications is more preferred. Pollefeys \textit{et al.}\textsuperscript{12,13} have proposed an algorithm which can calibrate the camera for variant kinds of camera constraint. In their approach, a projective structure is first reconstructed and then the scene structure is upgraded to metric stratum by utilizing the constraints derived from the absolute quadric. Since the image of the absolute quadric is directly related to the camera internal parameters, variant camera constraints can be easily imposed in their formulation to achieve flexible self-calibration. In this article, we will present a self-calibration algorithm that is suitable for two-view and multiple-view cases as well as for variant camera constraints. Unlike the approach proposed by
Pollefeys et al. who utilized the absolute quadric to calibrate the camera, we look for the camera intrinsic parameters to directly minimize the reprojection error, i.e., the sum of squared distances between the measured and reprojected image points. The reprojection error is originally a function of camera internal parameters, external parameters and 3D scene points. However, by exploiting the fundamental matrix we can estimate the most preferable camera external parameters for any given camera internal parameters. The 3D scene points can also be calculated by utilizing the estimated external parameters and the given internal parameters. Thus, the function to be minimized, i.e., the reprojection error, can be reduced to a function of only the camera internal parameters. To find the correct camera internal parameters, we employ the simplex method to minimize the resulting function because it does not require evaluating the function derivatives which is somewhat difficult for the proposed method. Since variant camera constraints can be easily imposed by modifying the parameters of the resulting function, the proposed algorithm is applicable for fixed or varying intrinsic parameters. In the following sections, the detailed procedures of proposed method will be described.
2. Proposed Method

2.A. Energy Function Formulation

Consider the perspective projection of a 3D scene point \( M \) onto an image plane forming the image point \( m \). The relation between \( M \) and \( m \) can be described by the equation: \( m \sim PM \), where \( P \) is the projection matrix of the camera and the symbol \( \sim \) indicates that this equation is equated up to a scale owing to the adoption of homogeneous coordinates in representing \( M \) and \( m \). With regard to a physical camera, the projection matrix \( P \) can be decomposed as \( K[R|−Rt] \), where the rotation matrix \( R \) and translation vector \( t \) are the extrinsic or external parameters of the camera. The matrix \( K \) is the camera calibration matrix taking the following structure

\[
K = \begin{bmatrix}
\tau f & \alpha & u_0 \\
0 & f & v_0 \\
0 & 0 & 1
\end{bmatrix},
\]

where \( f \) is the focal length, \( \tau \) stands for camera aspect ratio, \( \alpha \) is the skew factor and \((u_0, v_0)\) denotes the image principal point. These parameters are termed as the intrinsic or internal parameters of the camera. Camera self-calibration states that given a set of correspondences between several views, find the intrinsic and extrinsic parameters of the undergoing cameras. Intuitively, camera calibration can be achieved by finding the projection matrices and the 3D scene points so that the reprojected points are as close as possible to the measured corresponding points, i.e., camera
calibration can be achieved by minimizing the following energy function

$$\sum_{i,j} \left[ \left( m_{j,x}^{(i)} - \frac{P_1^{(i)^T} M_j}{P_3^{(i)^T} M_j} \right)^2 + \left( m_{j,y}^{(i)} - \frac{P_2^{(i)^T} M_j}{P_3^{(i)^T} M_j} \right)^2 \right],$$  \hspace{1cm} (2)$$

where the indices $i$ and $j$ indicate $i$th view and $j$th point, $(m_{j,x}^{(i)}, m_{j,y}^{(i)})$ is the measured image coordinate of $m_j^{(i)}$, and $P_k^{(i)}$ denotes the $k$th row of $P^{(i)}$. This formulation is the well-known bundle adjustment. It has the advantages of being tolerant of missing data while providing a true ML estimate. Typically, using bundle adjustment often requires a good initialization so that it can converge to a reasonable solution. Therefore, bundle adjustment is generally be used as the final step of any reconstruction algorithm. However, bundle adjustment often becomes an extremely large minimization problem because of the large number of parameters it involved. Thus, bundle adjustment will sometimes become quite costly. In addition, bundle adjustment can only calibrate the camera up to projective structure, i.e., the 3D scene structure can only be recovered up to an arbitrary projective transformation. To obtain metric calibration, some constraints must be imposed. A typical constraint is to consider the physical structure of the projection matrix, $P = K[R| - Rt]$. Thus, a formulation similar to bundle adjustment while ensuring metric calibration can then be defined as follows:

$$\text{minimize } E(\Theta) = \sum_{i,j} \left[ \left( m_{j,x}^{(i)} - \frac{P_1^{(i)^T} M_j}{P_3^{(i)^T} M_j} \right)^2 + \left( m_{j,y}^{(i)} - \frac{P_2^{(i)^T} M_j}{P_3^{(i)^T} M_j} \right)^2 \right]$$

subject to $P^{(i)} = K^{(i)} \left[ R^{(i)} | - R^{(i)} t^{(i)} \right], \forall i,$  \hspace{1cm} (3)
where \( \Theta \) denotes the parameter set of this energy function. Generally, \( \Theta \) should include all the calibration matrices \( K^{(i)} \), all the external parameters \( R^{(i)} \) and \( t^{(i)} \), and all the 3D scene points \( M_j \). It seems that this formulation is more complex than bundle adjustment owing to the additional constraints \( P^{(i)} = K^{(i)}[R^{(i)}| - R^{(i)}t^{(i)}], \forall i \). However, it will be shown later that the energy function given by Eq. (3) can be greatly simplified by the fundamental matrix. Specifically, the fundamental matrix provides a constraint to relate the calibration matrix \( K \) to the rotation matrix \( R \) and translation vector \( t \). Therefore, given an arbitrary set of \( K \), we can find a set of \( R \) and \( t \) that is most suitable to the given set of \( K \) from the given fundamental matrices. Subsequently, we can form the projection matrix \( P^{(i)} \) by \( K^{(i)}[R^{(i)}| - R^{(i)}t^{(i)}] \) and determine the 3D scene points \( M_j \) from the obtained \( P^{(i)} \) and the measured corresponding points. In this sense, we can evaluate the energy function of Eq. (3) at various points of \( \Theta \) and the constraints in Eq. (3) are automatically satisfied. Since \( R^{(i)}, t^{(i)} \) and \( M_j \) can be computed from \( K^{(i)} \) via the fundamental matrices, the parameter set \( \Theta \) is automatically reduced to only the set of \( K^{(i)} \), i.e., \( \Theta = \{K^{(i)}, \forall i\} \). We must emphasize that in the following sections \( \Theta = \{K^{(i)}, \forall i\} \) is just the parameter set of the given energy function, not the true calibration matrices of the cameras. For incorrect camera calibration matrices, we can still evaluate the given energy function from the estimated fundamental matrices and measured corresponding points. However, only the true calibration matrices can minimize this energy function. The method for evaluating and minimizing this energy function will be described in the
following sections.

2.B. Evaluating the Energy Function

To evaluate the given energy function for various $\Theta$, a world coordinate system must be selected first as a basis for determining the camera pose of each view. In the proposed method, one view (denoted as the first view in the following discussion) is selected as the reference view and the world coordinate system is aligned with the camera coordinate system of the reference view. Then, by the method proposed by Hartley,\textsuperscript{14} we can determine the camera pose of each view that is most suitable to the given $\Theta$ and fundamental matrix. In depth, the fundamental matrix $F^{(i)}$ between $i$th view and reference view can be expressed as $K^{(i)}^{-T}R^{(i)}S^{(i)}K^{(1)}^{-1}$, where $S^{(i)}$ is the skew symmetric matrix created from $t^{(i)}$. The essential matrix for $i$th view is defined as $E^{(i)} = R^{(i)}S^{(i)}$ and, therefore, given $\Theta = \{K^{(i)}, \forall i\}$ the essential matrix can then be calculated from $F^{(i)}$. Hartley demonstrated that $R^{(i)}$ and $t^{(i)}$ can be estimated up to a scale via the singular value decomposition of $E^{(i)}$. That is, if $E^{(i)} = UDV^T$, then

$$S^{(i)} = VZV^T$$ (4)

$$R^{(i)} = UGV^T \text{ or } R^{(i)} = UG^TV^T,$$ (5)

where

$$G = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$ (6)
By accounting for the two possible signs of $t^{(i)}$ and the two possible choices of $R^{(i)}$, there are four possible configurations for the camera. However, only one is survived by testing with a single 3D point to determine whether it is located in front of the two views. This does not imply that our method need to know in advance any 3D point to determine the correct camera configuration. The testing 3D point can be calculated from the recovered projection matrices and the image point correspondence. Notice that the previous method can only recover the camera position $t^{(i)}$ up to an arbitrary scale. Thus, to clarify this point, we denote the estimated translation vector as $\tilde{t}^{(i)}$ to distinguish it from the true $t^{(i)}$ ($t^{(i)} = s^{(i)}\tilde{t}^{(i)}$ for some scale $s^{(i)}$).

In addition to $R^{(i)}$, $\tilde{t}^{(i)}$ and the given $\Theta = \{K^{(i)}, \forall i\}$, we also require the 3D scene points $M_j$ to evaluate the energy function given by Eq. (3). Generally, simple linear triangulation can reconstruct the 3D scene point as long as the projection matrices are determined. Unfortunately, from previous method we cannot fully determine the projection matrix for each view owing to the unknown scale of $\tilde{t}^{(i)}$. However, the scale of $\tilde{t}^{(i)}$ and the 3D scene points can be obtained simultaneously by solving a large linear system. Specifically, the projection matrix of $i$th view can be expressed as $P^{(i)} = K^{(i)}[R^{(i)}| - s^{(i)}R^{(i)}\tilde{t}^{(i)}]$, where $s^{(i)}$ denotes the scale of $\tilde{t}^{(i)}$. Then, by the equation $m_j^{(i)} \sim P^{(i)}M_j$, each point $m_j^{(i)}$ provides two linear constraints on the unknown parameters $(X_j, Y_j, Z_j, s^{(i)})$, where $[X_j, Y_j, Z_j]^T$ is the inhomogeneous interpretation of $M_j$. By collecting all the unknown parameters into a vector $x$ and stacking all the constraints into a matrix $A$, a large linear system $Ax = 0$ is produced. If there are
n views and N scene points which are all seen by the n views, then the matrix A will be of size $2nN \times (3N + n)$. Since the world coordinate system is aligned with the camera coordinate system of the reference view, the scale of the reference view is irrelevant. Moreover, without any information about actual distance in the real world, we can only calibrate the camera and recover the 3D scene structure up to metric stratum, thus the scale of second view can be arbitrarily set. Therefore, we can selected $s^{(1)} = 0$ and $s^{(2)} = 1$. Then by elementary algebra computation, the linear system $Ax = 0$ turns to a new system $\tilde{A}\tilde{x} = b$, where the matrix $\tilde{A}$ is of size $2nN \times (3N + n - 2)$. The 3D scene points can then be obtained by solving this new linear system in least-squares sense.

One example of the energy function $E(\Theta)$ for a simple case is illustrated in Fig. 1(a). We generated a number of 3D points and took two views for these points. The camera internal parameters were assumed to be known except the focal length which was fixed for the two views. Thus, the parameter set $\Theta$ contains only the unknown but fixed focal length, i.e., $\Theta = f$. The graph of $E(f)$ shown in Fig. 1(a) was obtained by varying $f$ from 400 to 2000 and evaluating $E(f)$ using previous method. In this experiment, the true focal length was set to 1000 and Fig. 1(a) clearly indicates that the minimum of $E(f)$ is located at the desired point. The unique minimum in this range of $f$ allows us to apply a simple minimization algorithm to locate this minimum. This minimization algorithm is introduced in the following.
2.C. Minimizing the Energy Function

By previous approach, we can evaluate the energy function of Eq. (3) at various points of $\Theta$. This provides us a method to minimize this function by evaluating its values at several different points in a systematic way. One approach that can achieve this goal is the *simplex method*.\(^{15}\) The main advantage of this method is that it does not need to evaluate the derivatives of the energy function. Generally, the simplex method can not promise to be more efficient than other minimization algorithms such as Levenberg-Marquardt (LM) algorithm. However, other minimization algorithms often require estimating the function derivatives which is somewhat difficult owing to the additional constraints $P^{(i)} = K^{(i)}[R^{(i)} - R^{(i)}t^{(i)}]$ for all $i$ in Eq. (3). Typically, numerical approximation for derivatives is possible but this will introduce some estimation errors. On the other hand, we found that when the camera configuration is close to critical motion,\(^{16}\) the graph of $E(\Theta)$ will be a long valley and the convergence speed of LM algorithm will be quite slow and even cannot converge to the correct solution. One example that illustrates the inefficiency of LM algorithm near critical motion case is shown in Fig. 1(b). This experiment is similar to that of Fig. 1(a) except that the focal lengths of the two views are different. In this case, the parameter set $\Theta = \{f_1, f_2\}$, the focal lengths of the two views. The true focal lengths of the two views were set to 1000 and 1100 as marked in Fig. 1(b). The level sets of $E(\Theta)$ were generated by evaluating $E(\Theta)$ with $f_1$ and $f_2$ ranged from 500 to 2500. Figure 1(b) shows the
loci of the two minimization algorithms, simplex and LM, under the same number of function evaluations. This figure clearly reveals that LM algorithm is stepped slowly at the long valley and produces a solution that far away from the true one. However, the simplex method can always converge to the true solution for this near degenerated case.

The simplex method is similar to rolling a polyhedron in a high-dimensional space. There are four types of changing the shape of polyhedron: reflection, reflection and expansion, contraction, and multiple contraction. Given the initial vertices of the polyhedron, it will roll down and shrink to the minimum of the function. The initial vertices of polyhedron will certainly influence the final solution of camera self-calibration. However, as illustrated in Fig. 1 there is usually only one minimum in the possible range of $\Theta$, thus the initial points are not so critical as long as they are set to this possible range. This is reasonable since in most cases we can obtain the rough range of $\Theta$ from experience. The detailed procedures and an implementation of the simplex method can be found in Ref. 15.

2.D. Self-Calibration under Different Camera Constraints

In general, to calibrate the camera some constraints on the camera intrinsic parameters must be imposed.$^{8,12}$ One typical constraint for camera self-calibration is unknown but fixed camera intrinsic parameters. The setting of camera constraint should depend on the practical applications. Different applications have different camera
constraints. Since the parameter set $\Theta$ of our energy function directly represents the camera intrinsic parameters, different camera constraints can be easily imposed by modifying the parameter set of our energy function. For instance, for the case of unknown but fixed camera intrinsic parameters, we can modify the parameter set $\Theta$ from $\{K(i), \forall i\}$ to $\{f, \alpha, \tau, u_0, v_0\}$ and the constraint is then automatically satisfied. The estimated camera parameters will then obey the given constraint. There are several special cases that have been proved to be performed well by experiments using proposed approach. These special cases are listed in the following:

1. Known aspect ratio, skew factor and principal point, unknown but fixed focal length. In this case, $\Theta = f$. Figure 1(a) is an example of this case.

2. Known aspect ratio and skew factor, unknown but fixed principal point and focal length. In this case, $\Theta = \{f, u_0, v_0\}$.

3. Known aspect ratio, skew factor and principal point, unknown and varying focal length. In this case, $\Theta = \{f_i, \forall i\}$, where $f_i$ represents the focal length of $i$th view. Figure 1(b) is an example of this case.

4. Unknown but fixed camera intrinsic parameters. As mentioned above, $\Theta = \{f, \alpha, \tau, u_0, v_0\}$.

Previous cases are typical constraints for camera self-calibration. The proposed approach is also applicable to other special cases. For instance, the focal length of some
views may be varying while the others are fixed. This situation is occurred when
the camera has refocusing or zooming for some views and only have different camera
poses for other views. In this case, $\Theta$ may be written as $\{f, f_i, i \in I\}$, where the index
set $I$ contains those views with varying focal length and $f$ represents the fixed focal
length of other views. In the following, the experimental results for different camera
constraints are shown.

3. Experimental Results

3.A. Simulated Data

To verify the feasibility of proposed method, some experiments were conducted.
We generated 200 points randomly distributed on the surface of a sphere centered
at $(0, 0, 150)$ with radius 20. Six views for the 200 points were generated according
to the camera model described in Section 2.A. The external parameters of the
six views are listed in Table 1. The internal parameters of the camera will be
different according to different camera constraints. To analyze the performance of
our approach, the Gaussian white noise with standard deviation ranged from 0.0
to 1.0 were added in the corresponding points of the six views. The experimental
results are summarized in Table 2. We first utilized the corresponding points in the
first two views to calibrate the camera. For the two-view case, it is only possible
to recover two camera parameters, typically the focal length of the two views. Therefore, only two experiments, the fixed focal length and varying focal length, were
proceeded for the two-view case. For six-view case, variant camera constraints can be imposed, with known or unknown, fixed or varying camera parameters. From the results in Table 2, some points can be concluded. First, our approach can be applied in two-view and multiple-view cases. Second, our approach is applicable to variant camera constraints. Third, our approach can produce satisfactory results under the three noisy cases. Most relative errors are less than 5%. Since one important application of camera calibration is to recover the 3D points, we also illustrate in Fig. 2 the reconstructed 3D points of the five cases listed in Table 2. For this figure, the standard deviation of imposed Gaussian white noise is 0.5. Since we can only recover the 3D points up to metric stratum, i.e., to any similarity transformation, the absolute scale of the reconstructed 3D points is irrelevant. Only the relative positions and the overall shape of the reconstructed 3D points are important. Therefore, we omit the axis label in Fig. 2 to avoid confusion. From this figure, one can see that the sphere were correctly reconstructed, although some reconstructed points have slight movement.

To illustrate the performance of proposed method, we also compare our results with other techniques in this paper. For the two-view and varying focal length case, it has been known that the focal length can be directly computed from the fundamental matrix, epipoles and principal points. In this article, we employ the closed-form solution proposed by Bougnoux to compare with our approach. To improve estimation accuracy, each experiment was carried out 100 times and the mean and
standard deviation were calculated. The comparison results under different noise levels are listed in Table 3. This table reveals that the proposed method has nearly the same performance as Bougnoux’s closed-form solution. However, by considering the flexibility, our algorithm is more flexible than Bougnoux’s closed-form solution since our approach can be easily extended to fixed focal length and more than two views cases. For the six-view case, we compare our approach with a method proposed by Pollefeys et al. Pollefeys et al.\textsuperscript{12} developed a linear approach to estimate the varying focal length when the principal point and skew factor are known in advance.

The comparison results are shown in Table 4. In this experiment, the standard deviation of imposed Gaussian white noise was 0.5 and the results were obtained after 100 executions. From this table, one can see that our approach has smaller standard deviation than the linear approach of Pollefeys et al., which implies that our method is more stable than the method developed by Pollefeys et al. To examine the estimation accuracy, the relative error of the six views, which is defined as $|f - \hat{f}|/f$, where $f$ and $\hat{f}$ are the true and estimated focal length, respectively, is depicted in Fig. 3. From this figure we can see that our algorithm also has a more accurate estimation of focal length. Since the linear approach of Pollefeys et al. requires the projective reconstruction of the scene, it seems that degraded performance of this approach is arisen from the additional errors of the projective reconstruction. In our implementation the projective reconstruction was created from the first two views and then the projection matrices of other views are sequentially aligned.
with the obtained projective framework. The accuracy of projective reconstruction could be further improved if a more sophisticated method such as projective bundle adjustment was applied. However, projective bundle adjustment often involves a large number of parameters, thus is quite costly. Nevertheless, in our implementation both the Bougnoux’s closed-form solution and the linear approach of Pollefeys et al. are more efficient than the proposed method since their solutions can be directly computed while our algorithm is an iterative method.

Since in our method we evaluate the energy function by exploiting the fundamental matrix, the estimation accuracy of fundamental matrix will definitely influence the performance of our approach. In our experiments, the fundamental matrix was estimated by the normalized 8-point algorithm.\textsuperscript{9,17} There are many other methods reported in the literature that can produce more accurate estimation of fundamental matrix,\textsuperscript{18,19} thus we believe that our approach could be further improved if a sophisticated method for fundamental matrix estimation is applied.

3.B. Real Data

We also tested our approach on real image data. Five views of a car were captured and 312 corresponding points were manually selected. One view of the car is shown in Fig. 4(a). We assume zero skew factor and unit aspect ratio for the five views. The principal point is supposed to locate at the image center. Only the focal lengths
of the five views are unknown. Generally, the principal point may not exactly locate at image center, but this simple assumption can reduce the number of unknown parameters of the simplex method and a satisfactory result can still be yielded. The estimated focal lengths for the five views are 1463, 1468, 1595, 1634, and 1629. Since we do not know the exact focal lengths of the five views, we cannot judge the accuracy of estimated focal lengths. However, we can recover the 3D points from the estimated focal lengths. The results are illustrated in Figs. 4(b) and (c). We depict the results from different viewpoints to visualize the reconstructed 3D points. These figures show that the 3D shape of the car was roughly reconstructed, which implies that the estimated focal lengths are acceptable. This experiment demonstrates the feasibility of proposed method for real applications such as 3D scene points reconstruction and depth estimation.

3.C. Time Complexity

The time complexity of proposed method depends on the complexity for evaluating the given energy function. Recall that to evaluate this energy function, we must solve a linear system $\tilde{A}\tilde{x} = b$ to acquire the 3D scene points and the scale of each view. Since the matrix $\tilde{A}$ is of size $2nN \times (3N + n - 2)$, solving this linear system will be quite time-consuming if large views and points are in hand. One example can help us to realize the time-consuming in solving this linear system. For the six-view and 200 points case, the matrix $\tilde{A}$ is of size $2400 \times 604$ and it requires 8.77 s to evaluate
the energy function in our implementation. This result was obtained by running the program under Matlab environment on a PC with Pentium IV 2.4 GHz CPU and 768 MB RAM. Fortunately, a simplified approach can be applied to compute the 3D scene points while reducing the time complexity. The scale for each view can be individually computed for each point, thus solving the large linear system turns to solving $N$ much smaller linear systems with the size $2n \times (n + 1)$. That is, for each point $M_j$ a linear system with unknown parameters $(X_j, Y_j, Z_j, s^{(3)}, \ldots, s^{(n)})$ is formulated and solved. In this manner each point will produce a scale for each view. Theoretically, the calculated scales of each view will be the same for all points. However, owing to numerical errors and noise, they are different but quite close to the true one. We can select the final scale of each view by the average or median of these estimated scales. This is not exactly correct but a satisfactory result can still be generated and the computing time is greatly reduced. For example, it requires only 0.56 s to evaluate the energy function for previous case under the same environment and the performance is only slightly degraded. In fact, all the experimental results reported in this paper, including the simulated and real data, are all obtained using this simplified approach.

4. Conclusions

Camera self-calibration plays an important role in many applications of computer vision. For different applications there are generally different camera constraints. A
good self-calibration method should faithfully exploit these constraints to acquire more accurate results. In this paper, we have presented a self-calibration algorithm that is suitable for variant camera constraints. The proposed method is derived from minimizing an energy function similar to bundle adjustment. However, the proposed method does not have the problems encountered in bundle adjustment. Moreover, this approach is directly derived from the viewpoint of optimization; it does not require estimating some projective geometry entities such as plane at infinity and absolute conic. Only some basic matrix computations are sufficient to implement the proposed approach. Since in our approach the function parameters are just the camera intrinsic parameters, variant camera constraints can be easily imposed by modifying the function parameters. This leads to flexible camera self-calibration. Only slight modifications are sufficient to transfer the proposed approach to fit the practical requirements of many applications. One possible application of proposed approach is the robot vision where several cameras may be set up. The cameras may be calibrated using specific calibration target to acquire some camera parameters such as aspect ratio, skew factor and principal point. The focal length may be varying or fixed for some images. We can set suitable camera constraints in our approach according to practical situations to obtain more robust results. We have conducted some experiments and the results demonstrate the flexibility and well performance of proposed algorithm on real applications.
References


Table 1. Camera external parameters of simulated data

<table>
<thead>
<tr>
<th>View</th>
<th>Translation vector</th>
<th>Rotation axis</th>
<th>Rotation angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>View 1</td>
<td>(0, 0, 0)</td>
<td>(0, 0, 0)</td>
<td>0°</td>
</tr>
<tr>
<td>View 2</td>
<td>(20, 20, 0)</td>
<td>(1, 1, 0)</td>
<td>−12°</td>
</tr>
<tr>
<td>View 3</td>
<td>(-20, -20, 0)</td>
<td>(1, 1, 0)</td>
<td>12°</td>
</tr>
<tr>
<td>View 4</td>
<td>(-10, 15, 0)</td>
<td>(0, 1, 0)</td>
<td>7.2°</td>
</tr>
<tr>
<td>View 5</td>
<td>(15, -10, 0)</td>
<td>(0, 1, 0)</td>
<td>−7.2°</td>
</tr>
<tr>
<td>View 6</td>
<td>(10, 10, 0)</td>
<td>(1, 0, 0)</td>
<td>7.2°</td>
</tr>
</tbody>
</table>
Table 2. Summary of proposed method under variant camera constraints

<table>
<thead>
<tr>
<th>Number of views</th>
<th>Camera constraint</th>
<th>True camera parameters</th>
<th>Estimated camera parameters</th>
<th>σₙ = 0.0ᵃ</th>
<th>σₙ = 0.5</th>
<th>σₙ = 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 views</td>
<td>Known τ, α, p,</td>
<td>f = 1000</td>
<td>f = 1000.00</td>
<td>f = 1000.00</td>
<td>f = 1020.75</td>
<td>f = 1042.72</td>
</tr>
<tr>
<td></td>
<td>unknown but</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>fixed fᵇ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Known τ, α, p,</td>
<td>f₁ = 1000</td>
<td>f₁ = 1000.00</td>
<td>f₁ = 1010.75</td>
<td>f₁ = 1020.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unknown and</td>
<td>f₂ = 1100</td>
<td>f₂ = 1100.00</td>
<td>f₂ = 1121.18</td>
<td>f₂ = 1142.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varying f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 views</td>
<td>Known τ, α,</td>
<td>f = 1000</td>
<td>f = 1000.00</td>
<td>f = 1011.00</td>
<td>f = 1018.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unknown but</td>
<td>p = (640, 480)</td>
<td>p = (640, 480)</td>
<td>p = (642, 481)</td>
<td>p = (641, 482)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fixed p and f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unknown but</td>
<td>f = 1000</td>
<td>f = 999.99</td>
<td>f = 1035.70</td>
<td>f = 1066.94</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fixed τ, α, p,</td>
<td>τ = 1.0</td>
<td>τ = 1.00</td>
<td>τ = 0.95</td>
<td>τ = 0.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and f</td>
<td>α = 0.0</td>
<td>α = −0.0061</td>
<td>α = −0.0027</td>
<td>α = −0.0137</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p = (640, 480)</td>
<td></td>
<td>p = (640, 480)</td>
<td>p = (664, 461)</td>
<td>p = (631, 477)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Known τ, α, p,</td>
<td>f₁ = 1000</td>
<td>f₁ = 1000.00</td>
<td>f₁ = 1013.74</td>
<td>f₁ = 1028.10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unknown and</td>
<td>f₂ = 1100</td>
<td>f₂ = 1100.00</td>
<td>f₂ = 1130.73</td>
<td>f₂ = 1163.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varying f</td>
<td>f₃ = 900</td>
<td>f₃ = 900.00</td>
<td>f₃ = 893.67</td>
<td>f₃ = 885.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>f₄ = 1050</td>
<td>f₄ = 1050.00</td>
<td>f₄ = 1055.31</td>
<td>f₄ = 1064.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>f₅ = 950</td>
<td>f₅ = 950.00</td>
<td>f₅ = 929.70</td>
<td>f₅ = 906.74</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>f₆ = 1200</td>
<td>f₆ = 1200.00</td>
<td>f₆ = 1230.93</td>
<td>f₆ = 1260.72</td>
<td></td>
</tr>
</tbody>
</table>

ᵃ The symbol σₙ denotes the standard deviation of imposed Gaussian white noise.
ᵇ In this table, τ denotes the camera aspect ratio, α is the camera skew factor, p represents the image principal point, f stands for the focal length and fᵢ is the focal length of i-th view.
Table 3. Comparison of proposed method with Bougnoux’s closed-form solution

<table>
<thead>
<tr>
<th>Noise level</th>
<th>Focal length</th>
<th>Proposed method</th>
<th>Bougnoux’s method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>View 1</td>
<td>View 2</td>
<td>View 1</td>
</tr>
<tr>
<td>$\sigma_n = 0.0$</td>
<td>Mean</td>
<td>1000.00</td>
<td>1000.00</td>
</tr>
<tr>
<td></td>
<td>St. dev.</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$\sigma_n = 0.5$</td>
<td>Mean</td>
<td>999.08</td>
<td>1099.27</td>
</tr>
<tr>
<td></td>
<td>St. dev.</td>
<td>30.47</td>
<td>32.33</td>
</tr>
<tr>
<td>$\sigma_n = 1.0$</td>
<td>Mean</td>
<td>1000.48</td>
<td>1101.27</td>
</tr>
<tr>
<td></td>
<td>St. dev.</td>
<td>62.12</td>
<td>65.66</td>
</tr>
</tbody>
</table>

Table 4. Comparison of proposed method with the linear approach of Pollefeys et al.

<table>
<thead>
<tr>
<th>View</th>
<th>True focal length</th>
<th>Estimated focal length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed method</td>
</tr>
<tr>
<td>View 1</td>
<td>1000</td>
<td>996.44 ± 15.15</td>
</tr>
<tr>
<td>View 2</td>
<td>1100</td>
<td>1100.24 ± 19.14</td>
</tr>
<tr>
<td>View 3</td>
<td>900</td>
<td>899.65 ± 19.41</td>
</tr>
<tr>
<td>View 4</td>
<td>1050</td>
<td>1049.52 ± 24.15</td>
</tr>
<tr>
<td>View 5</td>
<td>950</td>
<td>953.44 ± 21.87</td>
</tr>
<tr>
<td>View 6</td>
<td>1200</td>
<td>1196.24 ± 22.75</td>
</tr>
</tbody>
</table>
List of Figure Captions

Fig. 1. (a) The graph of $E(\Theta)$ for two views and fixed focal length case. (b) The level sets of $E(\Theta)$ for two views and varying focal length case. The asterisk symbol represents the locus of LM algorithm while the circle symbol denotes the locus of one vertex of the simplex method. The two loci were obtained under the same number of function evaluations.

Fig. 2. Reconstruction results under Gaussian white noise with standard deviation equal to 0.5. (a) The simulated 3D points, (b)-(f) the reconstructed 3D points of the five cases in Table 2 from top to bottom, respectively.

Fig. 3. 3D reconstruction for some points on a car. (a) One view of the testing images, (b) front view and (c) top view of the reconstructed 3D point cloud.

Fig. 4. Relative errors of estimated focal length.
Fig. 1. (a) The graph of $E(\Theta)$ for two views and fixed focal length case. (b) The level sets of $E(\Theta)$ for two views and varying focal length case. The asterisk symbol represents the locus of LM algorithm while the circle symbol denotes the locus of one vertex of the simplex method. The two loci were obtained under the same number of function evaluations.
Fig. 2. Reconstruction results under Gaussian white noise with standard deviation equal to 0.5. (a) The simulated 3D points, (b)-(f) the reconstructed 3D points of the five cases in Table 2 from top to bottom, respectively.
Fig. 3. Relative errors of estimated focal length.
Fig. 4. 3D reconstruction for some points on a car. (a) One view of the testing images, (b) front view and (c) top view of the reconstructed 3D point cloud.