CAMERA CALIBRATION BY COMPUTATIONAL PROJECTIVE GEOMETRY

Kenichi Kanatani and Yasuhiro Onodera

Department of Computer Science, Gunma University
Kiryu, Gunma 376, Japan

ABSTRACT

A scheme of camera calibration is proposed. The *focal length* and the orientations of the scene coordinate axes are computed by detecting the *vanishing points* of two sets of lines orthogonally set in the scene. The absolute distance of the scene coordinate system is determined by locating a point whose scene coordinates are known. All points and lines are represented by unit vectors called *N-vectors*, and computation is based on projective geometry coupled with such computational considerations as efficiency of computation and suppression of noise in the data.

1. Introduction

Whenever we try to implement any computer vision technique by using a real camera, we immediately realize the importance of accurately calibrating the camera.[1–4,6–8)] In computer vision studies, the camera imaging is modeled as *perspective projection* from the origin O (called the *viewpoint*) of the camera-based XYZ-coordinate system onto an image plane parallel to the XY-plane in distance f, which is often referred to as the *focal length*, from the viewpoint O (Fig. 1).

Our scheme uses a specially designed "calibration board", which plays the role of the scene coordinate system. The focal length f is computed by detecting the *vanishing points* of the lines on the calibration board. The absolute distance of the scene coordinate origin from the camera is determined by locating a point whose position is known on the calibration board.

In our scheme, all points and lines are represented by unit vectors, which we call *N-vectors*. The treatment is based on a mathematical formalism, called *computational projective geometry*, which combines projective geometry with such computational considerations as efficiency of computation and suppression of noise in the data.

2. The Camera and the Scene Coordinate Systems

Let XYZ-coordinate system be the camera coordinate system with origin O (the viewpoint), and let $\bar{X}\bar{Y}\bar{Z}$-coordinate system be the scene coordinate system with origin $\bar{O}$ (Fig. 2).

Let $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ be the unit vectors lying along the $\bar{X}$-, $\bar{Y}$-, and $\bar{Z}$-axes, respectively. Let $\boldsymbol{m}_o$ be the unit vector starting from the viewpoint O and pointing toward the scene coordinate origin $\bar{O}$ (we call this vector $\boldsymbol{m}_o$ the *N-vector* of the scene coordinate origin $\bar{O}$). If the absolute distance $r_o = O\bar{O}$ of the scene coordinate origin $\bar{O}$ is known, the position and orientation of the scene $\bar{X}\bar{Y}\bar{Z}$-coordinate system are completely specified relative to the camera XYZ-coordinate system.

The *pose parameters* $\{\boldsymbol{R}, \boldsymbol{h}\}$ of the camera are defined such that the camera XYZ-coordinate system is obtained by first rotating the scene $\bar{X}\bar{Y}\bar{Z}$-coordinate system around its origin $\bar{O}$ by $\boldsymbol{R}$ and then translating it by $\boldsymbol{h}$, where the components of $\boldsymbol{R}$ and $\boldsymbol{h}$ are defined *with respect to the $\bar{X}\bar{Y}\bar{Z}$-coordinate system.* Then, we have the following result (we omit the proof):

Part of this work was supported by the Ministry of Education, Science, and Culture under Grant in Aid for Scientific Research (No. 02805043).

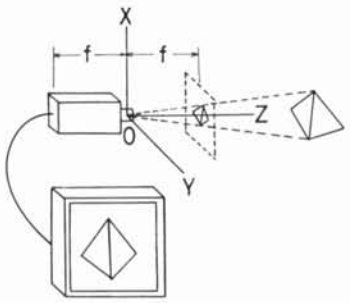

Fig. 1 Camera imaging geometry.

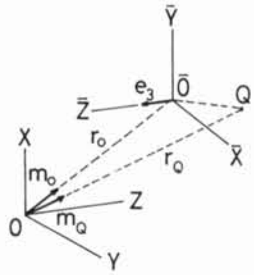

Fig. 2 The camera coordinate system and the scene coordinate system.

Proposition 1. Let

$$\boldsymbol{e}_1 = \begin{pmatrix} e_{1(1)} \\ e_{1(2)} \\ e_{1(3)} \end{pmatrix}, \quad \boldsymbol{e}_2 = \begin{pmatrix} e_{2(1)} \\ e_{2(2)} \\ e_{2(3)} \end{pmatrix}, \quad \boldsymbol{e}_3 = \begin{pmatrix} e_{3(1)} \\ e_{3(2)} \\ e_{3(3)} \end{pmatrix}, \tag{2.1}$$

$$\boldsymbol{m}_o = \begin{pmatrix} m_{o(1)} \\ m_{o(2)} \\ m_{o(3)} \end{pmatrix} \tag{2.2}$$

be the components of vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$, and $\boldsymbol{m}_o$ with respect to the camera XYZ-coordinate system. The pose parameters $\{\boldsymbol{R}, \boldsymbol{h}\}$ are given by

$$\boldsymbol{R} = \begin{pmatrix} e_{1(1)} & e_{1(2)} & e_{1(3)} \\ e_{2(1)} & e_{2(2)} & e_{2(3)} \\ e_{3(1)} & e_{3(2)} & e_{3(3)} \end{pmatrix}, \tag{2.3}$$

$$\boldsymbol{h}=-r_o\begin{pmatrix}R_{11}m_{o(1)}+R_{12}m_{o(2)}+R_{13}m_{o(3)}\\R_{21}m_{o(1)}+R_{22}m_{o(2)}+R_{23}m_{o(3)}\\R_{31}m_{o(1)}+R_{32}m_{o(2)}+R_{33}m_{o(3)}\end{pmatrix}. \tag{2.4}$$

3. Determination of the Motion Parameters

Suppose the camera is moved in the scene. The position and orientation of the $X'Y'Z'$-coordinate system after the motion relative to the XYZ-coordinate system before the motion are specified so that the $X'Y'Z'$-coordinate system is obtained by first rotating the XYZ-coordinate system around its origin O by $\bar{R}$ and then translating it $\bar{h}$, where the components of $\bar{R}$ and $\bar{h}$ are defined *with respect to the XYZ-coordinate system.* We call $\{\bar{R}, \bar{h}\}$ the *motion parameters* (Fig. 3). Then, we obtain the following result (we omit the proof).

Proposition 2. If the pose parameters are $\{\boldsymbol{R}, \boldsymbol{h}\}$ and $\{\boldsymbol{R}', \boldsymbol{h}'\}$ before and after the motion, respectively, the motion parameters $\{\bar{R}, \bar{h}\}$ are given by

$$\bar{R}=\boldsymbol{R}^T\boldsymbol{R}', \quad \bar{h}=\boldsymbol{R}^T(\boldsymbol{h}'-\boldsymbol{h}). \tag{3.1}$$

4. Determination of the Absolute Depth

Assuming that vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$, and $\boldsymbol{m}_o$ have already been obtained, consider the absolute distance r_o of the scene coordinate origin $\bar{O}$ from the viewpoint O. Suppose we observe a point Q on the $\bar{X}\bar{Y}$-plane, and suppose the distance $\bar{O}Q$ from the scene coordinate origin $\bar{O}$ is known (Fig. 2). Let $\boldsymbol{m}_Q$ be the unit vector starting from the viewpoint O and pointing toward Q (we call this vector the *N-vector* of point Q). In the following, $(\boldsymbol{a}, \boldsymbol{b})$ designates the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\|\boldsymbol{a}\|$ designates the norm of vector $\boldsymbol{a}$.

Proposition 3. The absolute distance r_o of the scene coordinate origin $\bar{O}$ from the viewpoint O is given by

$$r_o=\frac{|(\boldsymbol{m}_Q, \boldsymbol{e}_3)|\bar{O}Q}{\|(\boldsymbol{m}_o, \boldsymbol{e}_3)\boldsymbol{m}_Q-(\boldsymbol{m}_Q, \boldsymbol{e}_3)\boldsymbol{m}_o\|}. \tag{4.1}$$

Proof. If we put $r_Q = OQ$, we have

$$\vec{\bar{O}Q}=\vec{OQ}-\vec{O\bar{O}}=r_Q\boldsymbol{m}_Q-r_o\boldsymbol{m}_o. \tag{4.2}$$

Since point Q is on the $\bar{X}\bar{Y}$-plane, vector $\vec{\bar{O}Q}$ is orthogonal to $\boldsymbol{e}_3$. Hence,

$$(\vec{\bar{O}Q}, \boldsymbol{e}_3)=r_Q(\boldsymbol{m}_Q, \boldsymbol{e}_3)-r_o(\boldsymbol{m}_o, \boldsymbol{e}_3)=0. \tag{4.3}$$

Eliminating r_Q from eqns (4.2) and (4.3), we have

$$\vec{\bar{O}Q}=r_o\Big(\frac{(\boldsymbol{m}_o, \boldsymbol{e}_3)}{(\boldsymbol{m}_Q, \boldsymbol{e}_3)}\boldsymbol{m}_Q-\boldsymbol{m}_o\Big), \tag{4.4}$$

from which we obtain eqn (4.1).

5. N-vectors of Points and Lines

Given a point P in the scene, we call the unit vector $\boldsymbol{m}$ starting from the viewpoint O and pointing toward P the *N-vector* of point P. Evidently, if (a, b) are the image coordinates of point P (Fig. 4), its N-vector $\boldsymbol{m}$ is given by[5)]

$$\boldsymbol{m}=\mathrm{N}\Big[\begin{pmatrix}a\\b\\f\end{pmatrix}\Big], \tag{5.1}$$

where $\mathrm{N}[\boldsymbol{a}] = \boldsymbol{a}/\|\boldsymbol{a}\|$ designates the normalization of vector $\boldsymbol{a}$ into a unit vector.

Given a line l in the scene, we define its *N-vector* as the unit vector $\boldsymbol{n}$ normal to the plane passing via the viewpoint O and line l (Fig. 4). It is easily confirmed

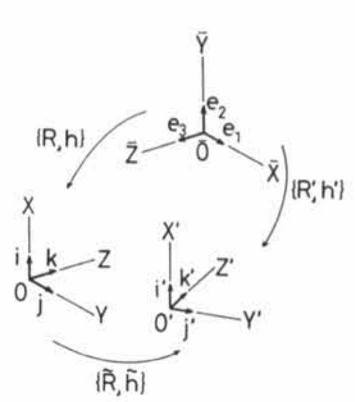

Fig. 3 Motion parameters of the two cameras.

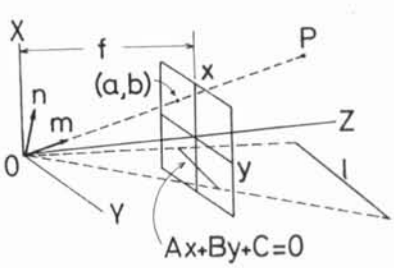

Fig. 4 The N-vectors of a point and a line.

that if the projection image of line l is $Ax + By + C = 0$, its N-vector $\boldsymbol{n}$ is given by[5)]

$$\boldsymbol{n}=\pm\mathrm{N}\Big[\begin{pmatrix}A\\B\\C/f\end{pmatrix}\Big], \tag{5.2}$$

where the sign is arbitrarily chosen.

Proposition 4. The N-vector $\boldsymbol{m}$ of the intersection P of two lines l_1 and l_2 on the image plane is given by

$$\boldsymbol{m}=\pm\mathrm{N}[\boldsymbol{n}_1\times\boldsymbol{n}_2], \tag{5.3}$$

where $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are the N-vectors of the lines l_1 and l_2, respectively, and the sign is chosen so that the Z component becomes positive.

Proof. It is easy to see from Fig. 4 that a point whose N-vector is $\boldsymbol{m}$ is on a line whose N-vector is $\boldsymbol{n}$ on the image plane if and only if $(\boldsymbol{m}, \boldsymbol{n}) = 0$. Hence, if point P is on both lines l_1 and l_2, N-vector $\boldsymbol{m}$ must be orthogonal to both $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$.

Lines which meet at a common intersection on the image plane are said to be *concurrent.* If lines are detected by image processing, error is inevitable, and lines which are supposed to be concurrent may not be concurrent.

Since the common intersection may not be found within the image frame (it is located *at infinity* if the lines happen to be parallel on the image plane), the following procedure is most reasonable for estimating the common intersection of not necessarily concurrent lines (Fig. 5).[5)]

Procedure 1. Let $\boldsymbol{n}_1$, ..., $\boldsymbol{n}_N$ be the N-vectors of not necessarily concurrent lines. The N-vector $\boldsymbol{m}$ of their common intersection is estimated by the unit eigenvector of the *moment matrix*

$$M = \sum_{\alpha=1}^{N} W_\alpha n_\alpha n_\alpha^T \qquad (5.4)$$

for the minimum eigenvalue, where W_α is an appropriate weight for the α-th line. The sign is chosen so that the Z component becomes positive.

Derivation. If all the lines exactly pass through a point whose N-vector is $\boldsymbol{m}$, we have $(\boldsymbol{m}, \boldsymbol{n}_\alpha) = 0$, $\alpha = 1, ..., N$ (Fig. 4). Hence, it is reasonable to determine $\boldsymbol{m}$ by minimizing $\sum_{\alpha=1}^{N} W_\alpha (\boldsymbol{m}, \boldsymbol{n}_\alpha)^2$. In terms of the moment matrix $\boldsymbol{N}$ of eqn (5.4), this is rewritten as $(\boldsymbol{m}, \boldsymbol{N}\boldsymbol{m})$, which is minimized under the constraint $\|\boldsymbol{m}\| = 1$, as is well known, by the unit eigenvector of $\boldsymbol{N}$ for the minimum eigenvalue.

In order to apply Proposition 4 and Procedure 1, *the true value of the focal length f is not necessary*: An arbitrary value of the focal length f can be used for computing N-vectors. If the value of the focal length f is altered, it is easy to confirm that all N-vectors are altered as follows (Fig. 6):

Proposition 5. If $\boldsymbol{m} = (m_1, m_2, m_3)^T$ is the N-vector of a point and $\boldsymbol{n} = (n_1, n_2, n_3)^T$ is the N-vector of a line with respect to focal length f, their N-vectors $\boldsymbol{m}'$, $\boldsymbol{n}'$ with respect to another focal length f' are respectively given by

$$\boldsymbol{m}' = N\left[\begin{pmatrix} m_1 \\ m_2 \\ (f'/f)m_3 \end{pmatrix}\right], \quad \boldsymbol{n}' = N\left[\begin{pmatrix} n_1 \\ n_2 \\ (f/f')n_3 \end{pmatrix}\right]. \qquad (5.5)$$

6. Determination of the Focal Length

Lines parallel in the scene, when projected onto the image plane, meet at a common point called the *vanishing point*. The following is fundamental (Fig. 7):[5]

Proposition 6. The N-vector of the vanishing point of a line in the scene indicates its 3-D orientation.

From this, we obtain the following procedure to compute the focal length f.

Procedure 2.

1. Take an image of two mutually orthogonal sets of parallel lines in the scene.
2. Assuming a tentative value $\hat{f}$ of the focal length, compute the N-vectors $\boldsymbol{m} = (m_1, m_2, m_3)^T$ and $\boldsymbol{m}' = (m_1', m_2', m_3')^T$ of the vanishing points of these lines by Procedure 1.
3. The true focal length f is given by

$$f = \hat{f}\sqrt{-\frac{m_1 m_1' + m_2 m_2'}{m_3 m_3'}}. \qquad (6.1)$$

Derivation. If $\boldsymbol{m} = (m_1, m_2, m_3)^T$ and $\boldsymbol{m}' = (m_1', m_2', m_3')^T$ are the N-vectors of the vanishing points with respect to focal length $\hat{f}$, their N-vectors with respect to the true focal length f are respectively given, according to Proposition 5, by

$$N\left[\begin{pmatrix} m_1 \\ m_2 \\ (f/\hat{f})m_3 \end{pmatrix}\right], \quad N\left[\begin{pmatrix} m_1' \\ m_2' \\ (f/\hat{f})m_3' \end{pmatrix}\right]. \qquad (6.2)$$

According to Proposition 6, they indicate the 3D orientations of the two sets of parallel lines orthogonal to each

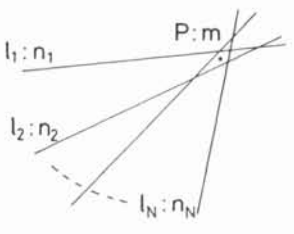

Fig. 5 Estimation of the common intersection.

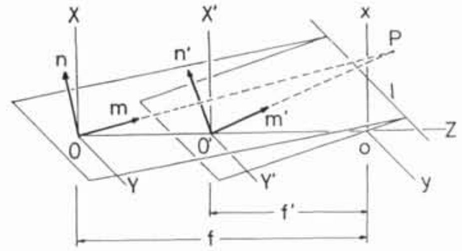

Fig. 6 The focal length f and N-vectors.

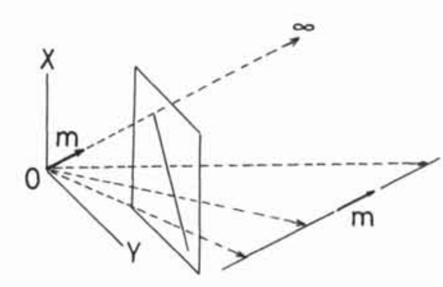

Fig. 7 The vanishing point of a line in the scene.

other. Hence, they must be orthogonal to each other, i.e.,

$$m_1 m_1' + m_2 m_2' + \left(\frac{f}{\hat{f}}\right)^2 m_3 m_3' = 0, \qquad (6.3)$$

from which follows eqn (6.1).

7. Procedure of Camera Calibration

Now, we describe a camera calibration procedure using a calibration board on which the square grid pattern of Fig. 8 is drawn.

The scene $\bar{X}\bar{Y}\bar{Z}$-coordinate system is defined by identifying point P_9 with the origin $\bar{O}$ and defining the $\bar{X}$- and $\bar{Y}$-axes by $\boldsymbol{e}_1 = \overrightarrow{P_9 P_4}$ and $\boldsymbol{e}_2 = \overrightarrow{P_9 P_2}$. The sides of the four squares are all regarded as having unit length. The $\bar{Z}$-axis is perpendicular to both the $\bar{X}$- and $\bar{Y}$-axes (hence $\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2$). Then, we obtain the following procedure.

Procedure 3.

1. Detect line segments in the calibration pattern image (say, by the Hough transform or by manual specification though an interactive interface).
2. Fit lines to the detected line segments by the least-squares method (Fig. 9), and compute the N-vectors of the fitted lines with respect a temporary focal length $\hat{f}$.
3. Compute the N-vectors of points P_1, ..., P_9 with

respect the temporary focal length $\hat{f}$ by Proposition 4.

4. Compute the N-vector $\boldsymbol{m}$ of the vanishing point of lines $P_1P_2P_3$, $P_8P_9P_4$, $P_7P_6P_5$ and the N-vector $\boldsymbol{m}'$ of the vanishing point of lines $P_7P_8P_1$, $P_6P_9P_2$, $P_5P_4P_3$ with respect to the temporary focal length $\hat{f}$ by Procedure 1.
5. Compute the true focal length f by Procedure 2.
6. Convert the N-vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$ of the vanishing points into the N-vectors with respect to the true f by Proposition 5, choosing their signs so that

$$(\boldsymbol{m}, \boldsymbol{m}_4-\boldsymbol{m}_8)>0, \quad (\boldsymbol{m}', \boldsymbol{m}_2-\boldsymbol{m}_6)>0, \tag{7.1}$$

where $\boldsymbol{m}_2$, $\boldsymbol{m}_4$, $\boldsymbol{m}_6$, $\boldsymbol{m}_8$ are the N-vectors (with respect to the true f) of points P_2, P_4, P_6, P_8, respectively.
7. Compute $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ by

$$\boldsymbol{e}_1=\frac{1}{\sqrt{2}}\left(N[\boldsymbol{m}+\boldsymbol{m}']+N[\boldsymbol{m}-\boldsymbol{m}']\right),$$

$$\boldsymbol{e}_2=\frac{1}{\sqrt{2}}\left(N[\boldsymbol{m}+\boldsymbol{m}']-N[\boldsymbol{m}-\boldsymbol{m}']\right). \tag{7.2}$$

This process forces vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$ to be orthogonal (Fig. 10) in case they are not (due to nose, etc.).
8. Compute

$$\boldsymbol{e}_3=\boldsymbol{e}_1\times\boldsymbol{e}_2, \tag{7.3}$$

and let the unit vector $\boldsymbol{m}_o$ be the N-vector (with respect to the true f) of point P_9.
9. Compute the absolute distance r_o by Proposition 3, in which we take the reference point Q to be each of P_1, ..., P_8 with the knowledge that

$$P_9P_1=P_9P_3=P_9P_5=P_9P_7=\sqrt{2},$$

$$P_9P_2=P_9P_4=P_9P_6=P_9P_8=1, \tag{7.4}$$

and average the results.
10. Compute the pose parameters $\{\boldsymbol{R}, \boldsymbol{h}\}$ by Proposition 1.

8. Concluding Remarks

We have presented a scheme of camera calibration from images. We have shown that a consistent treatment is possible if all points and lines are represented by unit vectors, which we called "N-vectors".

The present scheme is very easy to implement and applicable to self-sensing of the locations of mobile robots as well as calibration of the positions of the two cameras for stereo systems.

References

1. B. Caprile and V. Torre, Using vanishing points for camera calibration, *Int. J. Comput. Vision*, **4** (1990), 127-140.
2. T. Echigo, A camera calibration technique using sets of parallel lines, *Proc. IEEE Workshop Industrial Appl. Machine Intell. Vision*, April 1989, Tokyo, Japan, pp. 151-156.
3. O. D. Faugeras and G. Toscani, Camera calibration for 3D computer vision, *Proc. IEEE Int. Workshop Industrial Appl. Machine Vision Machine Intell.*, February 1987, Tokyo, Japan, pp. 240-247.

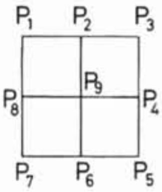

Fig. 8 The square grid pattern of the calibration board.

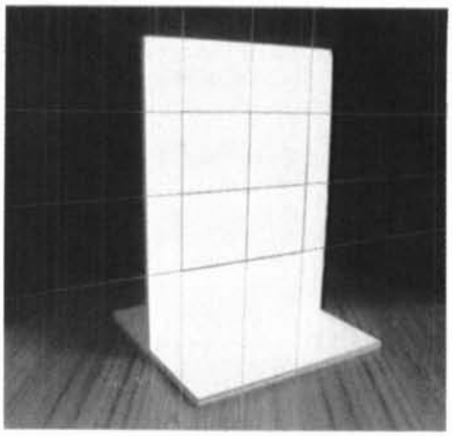

Fig. 9 Line fitting of the calibration board image.

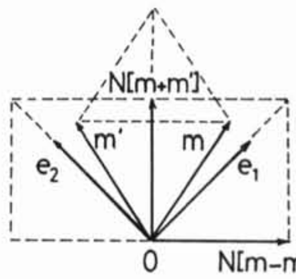

Fig. 10 Orthogonalization of two unit vectors.

4. W. I. Grosky and L. A. Tamburino, A unified approach to the linear camera calibration problem, *IEEE Trans. Pattern Anal. Machine Intell.*, **12** (1990), 663-671.
5. K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer, Berlin, 1990.
6. R. K. Lentz and R. Y. Tsai, Techniques for calibration of the scale factor and image center for high-accuracy 3-D machine vision metrology, *IEEE Trans. Pattern Anal. Machine Intell.*, **10** (1988), 713-720.
7. R. K. Lentz and R. Y. Tsai, Calibrating a Cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship, *IEEE Trans. Pattern Anal. Machine Intell.*, **11** (1989), 916-928.
8. R. Y. Tsai, An efficient and accurate camera calibration technique for 3D machine vision, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, June 1986, Miami Beach, FL, USA, pp. 364-376.