5th Symposium on Sensing via Image Information (SII'99), June 1999, Yokokahma, Japan.

Calibration of a Moving Camera Using a Planar Pattern
平面パタンを用いる移動カメラのキャリブレーション

Chikara Matsunaga* and Kenichi Kanatani†
松永 力 *　　金谷 健一 †
*Broadcast Division, FOR-A Co. Ltd, 2-3-3 Ohsaku, Sakura, Chiba 285-0802 Japan
†Department of Computer Science, Gunma University, Kiryu, Gunma 376-8515 Japan
*(株) 朋栄 放送システム開発部　†群馬大学工学部情報工学科

Abstract　複合現実感実現の基本となるカメラキャリブレーションのために、物体の背後に参照板を設置し、カメラ位置やズームを変えながら物体を自由に撮影する過程でその画像を解析しながら実行する方法 (「同時校正」) を提案する。これにより柔軟な応用が可能となり、簡単な処理で実現できる。観測される特徴点が事前には未知であり数も限定されているので高精度の計算が必要であり、また精度が状況ごとにまたフレームごとに異なる。本論文では誤差がないときに厳密解となる解析解を導出するとともに、誤差のあるとき精度の理論限界を達成する最適解を計算する。またその解の信頼性評価を行う手順を述べ、その有効性をシミュレーションおよび実画像実験によって確認する。本方法はＣＧと実写人物画像とを合成する「仮想スタジオ」にも有効である。

1. Introduction

Visually presenting 3-D shapes of real objects is one of the main goals of many Internet applications such as network cataloging and virtual museums, and generating virtual images by embedding graphics objects in real scenes or real objects in graphics scenes, known as *mixed reality*, is one of the central themes of image and media applications today. In order to reconstruct the 3-D shapes of real objects or scenes for such applications, we need to know the 3-D position of the camera that we use and its internal parameters. Thus, camera calibration is a first step in all vision and media applications.

The standard method is *pre-calibration*: we estimate the internal parameters by taking images of objects or patterns of known 3-D shape in a controlled environment in advance. The best known procedure for this is the method of Tsai [13]. If the internal parameters are known, we can compute the 3-D camera position and the 3-D shape of the object up to scale by a technique called *structure from motion* [2, 14]. If we know the 3-D shapes of the objects or the 3-D structure of the scene that we are observing, we can compute the 3-D camera position from a single image. This problem has been studied by many researchers as *self-localization* for mobile robot applications [1, 5, 11, 12].

Recently, techniques for computing both the camera parameters and the 3-D positions of the camera by observing multiple images of the same scene about which we have no prior knowledge have intensively been studied [8, 14]. Such a technique, known as *self-calibration*, may be very useful in unknown environments such as outdoors, but for stable reconstruction it requires a long sequence of images of the same scene taken from different and unconstrained camera positions and feature matching among them. It cannot be applied if the camera motion is constrained or if the scene changes as the camera moves, and the amount of computation required is too large for real-time applications.

In this paper, we focus on *virtual studio* applications: we take images of moving objects such as persons and superimpose them in a graphics-generated background in real time by computing the 3-D positions and the internal parameters of a moving camera. Since the camera parameters, such as zooming, may change from frame to frame, the calibration must be done for each frame. Hence, we cannot pre-calibrate or self-calibrate the camera. This difficulty can be avoided by placing an easily distinguishable planar pattern with a known geometry in the scene (Fig. 1): we detect the image of the pattern in each frame, compute the camera position and the internal parameters from it, and removing the image of the pattern by image segmentation. We call this *simultaneous calibration*.

Unlike pre-calibration, for which manual interventions can be employed, simultaneous calibration must be completely automated. To this end, we have already designed a special marker pattern in such a way that it can be detected with theoretically maximum robustness [6, 7]. Since, unlike in pre-calibration, we have no control over the position of the pattern relative to the current camera position or the number of marker points that are visible in the current frame, the accuracy of calibration is different from frame to frame. Hence, not only do

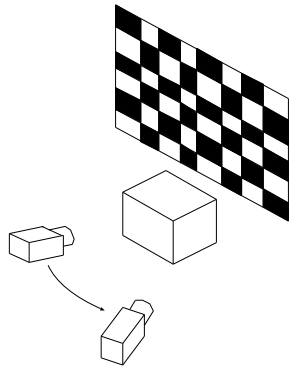


Figure 1: Simultaneous calibration of a moving camera.

we need an accurate computational procedure but also a scheme for evaluating the reliability of the computed solution.

In this paper, we describe such a technique by introducing a statistical model of image noise and applying the theory of statistical optimization of Kanatani [3]. We demonstrate its effectiveness by simulation and real-image experiments.

2. Basic Principle

We fix an XYZ world coordinate system and place a planar pattern in parallel to the XY plane at a known distance d. We take the focal length f as the only unknown camera parameter, assuming that other parameters have already been pre-calibrated so that the imaging geometry can be modeled as a perspective projection.

Suppose we place a hypothetical camera with a known focal length f_0 in such a way that the center of projection is at the origin, the Z-axis coincides with the optical axis, and the world X- and Y-axes are parallel to the image x- and y-axes. Then, we rotate the camera by an unknown rotation matrix $\boldsymbol{R}$, translate it by an unknown vector $\boldsymbol{t}$, and change the focal length into an unknown value f. We call $\{\boldsymbol{t}, \boldsymbol{R}\}$ the *motion parameters*. Suppose N points on the planar pattern with known coordinates (X_α, Y_α, d), are observed at (x_α, y_α) in the image, $\alpha = 1, ..., N$. If we define the 3-D vectors

$$\bar{\boldsymbol{x}}_\alpha = \begin{pmatrix} X_\alpha/d \\ Y_\alpha/d \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}_\alpha = \begin{pmatrix} x_\alpha/f_0 \\ y_\alpha/f_0 \\ 1 \end{pmatrix}, \tag{1}$$

we have the following relationship [3]:

$$\boldsymbol{x}_\alpha = Z[\boldsymbol{H}\bar{\boldsymbol{x}}_\alpha]. \tag{2}$$

Here, $Z[\,\cdot\,]$ denotes normalization to make the third component 1, and $\boldsymbol{H}$ is the matrix in the following form [3]:

$$\boldsymbol{H} = \begin{pmatrix} 1 & & \\ & 1 & \\ & & f_0/f \end{pmatrix} \boldsymbol{R}^\top \left(\boldsymbol{I} - \frac{\boldsymbol{t}\boldsymbol{k}^\top}{d} \right). \tag{3}$$

Throughout this paper, $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ denote $(1,0,0)^\top$, $(0,1,0)^\top$, and $(0,0,1)^\top$, respectively. Eq. (2) defines an image transformation called *homography* [3]. A statistically optimal algorithm for computing it from feature correspondences was presented by Kanatani [4] and its C++ code is publicly available via the Web[1] .

Since the unknowns $\{\boldsymbol{t}, \boldsymbol{R}\}$ and f have 7 degrees of freedom, we can determine them in theory if four or more points $\{(x_\alpha, y_\alpha)\}$ are observed.

3. Optimal Computation

Let $V[\boldsymbol{x}_\alpha]$ be the covariance matrix of the data vector $\boldsymbol{x}_\alpha$. We assume that it is known only up to scale and write

$$V[\boldsymbol{x}_\alpha] = \epsilon^2 V_0[\boldsymbol{x}_\alpha]. \tag{4}$$

We call the unknown magnitude ϵ the *noise level* and the matrix $V_0[\boldsymbol{x}_\alpha]$ the *normalized covariance matrix*. Since the third component of $\boldsymbol{x}$ is 1, $V_0[\boldsymbol{x}_\alpha]$ is a singular matrix of rank 2 with zeros in the third low and the third column. If the noise has no particular dependence on positions and orientations, we use the default value $\mathrm{diag}(1,1,0)$ (the diagonal matrix with diagonal elements 1, 1, and 0 in that order). An optimal estimate of $\boldsymbol{H}$ is obtained by minimizing the sum of the squared *Mahalanobis distances*

$$J = \frac{1}{N}\sum_{\alpha=1}^{N} (\boldsymbol{x}_\alpha - Z[\boldsymbol{H}\bar{\boldsymbol{x}}_\alpha], V_0[\boldsymbol{x}_\alpha]^-(\boldsymbol{x}_\alpha - Z[\boldsymbol{H}\bar{\boldsymbol{x}}_\alpha])), \tag{5}$$

where $(\,\cdot\,)^-$ denotes the (Moore-Penrose) generalized inverse. We view J as a function of non-dimensional variables

$$\phi = \frac{f}{f_0}, \qquad \boldsymbol{\tau} = \frac{\boldsymbol{t}}{d}. \tag{6}$$

The first order perturbations of ϕ and $\boldsymbol{\tau}$ are written, respectively, as $\phi \to \phi + \Delta\phi$ and $\boldsymbol{\tau} \to \boldsymbol{\tau} + \Delta\boldsymbol{\tau}$. The first order perturbation of $\boldsymbol{R}$ is written as $\boldsymbol{R} \to \boldsymbol{R} + \Delta\boldsymbol{\Omega} \times \boldsymbol{R}$, where $\Delta\boldsymbol{\Omega}$ is a 3-D vector and $\Delta\boldsymbol{\Omega} \times \boldsymbol{R}$ is a matrix whose columns are the vector products of $\Delta\boldsymbol{\Omega}$ and each columns of $\boldsymbol{R}$. We define the gradient ∇J and the Hessian $\nabla^2 J$ with respect to $\{\phi, \boldsymbol{\tau}, \boldsymbol{R}\}$ in such a way that the Taylor expansion of J has the form

$$\begin{aligned} &J(\phi + \Delta\phi, \boldsymbol{\tau} + \Delta\boldsymbol{\tau}, \boldsymbol{R} + \Delta\boldsymbol{\Omega} \times \boldsymbol{R}) \\ &= J(\phi, \boldsymbol{\tau}, \boldsymbol{R}) + (\nabla J, \begin{pmatrix} \Delta\phi \\ \Delta\boldsymbol{\tau} \\ \Delta\boldsymbol{\Omega} \end{pmatrix}) \\ &\quad + \frac{1}{2}(\begin{pmatrix} \Delta\phi \\ \Delta\boldsymbol{\tau} \\ \Delta\boldsymbol{\Omega} \end{pmatrix}, \nabla^2 J \begin{pmatrix} \Delta\phi \\ \Delta\boldsymbol{\tau} \\ \Delta\boldsymbol{\Omega} \end{pmatrix}) + \cdots, \end{aligned} \tag{7}$$

[1] http://www.ail.cs.gunma-u.ac.jp/~kanatani/e

where $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The solution that minimizes J is obtained by the following Newton iterations:

1. Give an initial guess of ϕ, $\boldsymbol{\tau}$, and $\boldsymbol{R}$.
2. Compute the gradient ∇J and the Hessian $\nabla^2 J$.
3. Compute $\Delta\phi$, $\Delta\boldsymbol{\tau}$, and $\Delta\boldsymbol{\Omega}$ by solving the linear equation

$$\left(\nabla^2 J\right) \begin{pmatrix} \Delta\phi \\ \Delta\boldsymbol{\tau} \\ \Delta\boldsymbol{\Omega} \end{pmatrix} = -\nabla J. \tag{8}$$

4. If $|\Delta\phi| < \epsilon_\phi$, $\|\Delta\boldsymbol{\tau}\| < \epsilon_\tau$, and $\|\Delta\boldsymbol{\Omega}\| < \epsilon_{\mathbf{R}}$, return ϕ, $\boldsymbol{\tau}$, and $\boldsymbol{R}$ and stop. Otherwise, update ϕ, $\boldsymbol{\tau}$, and $\boldsymbol{R}$ in the form

$$\phi \leftarrow \phi + \Delta\phi, \quad \boldsymbol{\tau} \leftarrow \boldsymbol{\tau} + \Delta\boldsymbol{\tau}, \quad \boldsymbol{R} \leftarrow \mathcal{R}(\Delta\boldsymbol{\Omega})\boldsymbol{R}, \tag{9}$$

and go back to Step 3.

Here, $\mathcal{R}(\Delta\boldsymbol{\Omega})$ denotes the rotation matrix of angle $\|\Delta\boldsymbol{\Omega}\|$ around $\Delta\boldsymbol{\Omega}$, and ϵ_ϕ, ϵ_τ, are $\epsilon_{\mathbf{R}}$ are thresholds for convergence.

4. Reliability Evaluation

The squared noise level ϵ^2 can be estimated from the residual $\hat{J}$ in the following form [3]:

$$\hat{\epsilon}^2 = \frac{\hat{J}}{2 - 7/N}. \tag{10}$$

Let $\nabla^2 \hat{J}$ be the resulting Hessian. The covariance matrix of $\{\hat{\phi},\ \hat{\boldsymbol{\tau}},\ \hat{\boldsymbol{R}}\}$ is estimated in the follwoing form [3]:

$$V[\hat{\phi}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{R}}] = \frac{2\hat{\epsilon}^2}{N}\left(\nabla^2 \hat{J}\right)^{-1}. \tag{11}$$

If the true values of $\{\phi,\ \boldsymbol{\tau},\ \boldsymbol{R}\}$ are substituted, this gives the *Cramer-Rao lower bound* on $V[\hat{\phi}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{R}}]$ [3].

The (1,1) element $V[\hat{\phi}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{R}}]$ gives the variance $V[\hat{\phi}]$ of ϕ. If the error distibution is approximated to be Gaussian, the 99.7% confidence interval of f is given by

$$f_0(\hat{\phi} - 3\sqrt{V[\hat{\phi}]}) < f < f_0(\hat{\phi} + 3\sqrt{V[\hat{\phi}]}). \tag{12}$$

The submatrix of $V[\hat{\phi}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{R}}]$ defined by its second to fourth rows and columns gives the covariance matrix $V[\hat{\boldsymbol{\tau}}]$ of $\boldsymbol{\tau}$. Let $\bar{\boldsymbol{R}}$ be the true rotation. Let $\Delta\Omega$ and $\boldsymbol{l}$ be the angle and axis of the relative rotation $\hat{\boldsymbol{R}}\bar{\boldsymbol{R}}^\top$. If we define vector

$$\Delta\boldsymbol{\Omega} = \Delta\Omega\boldsymbol{l}, \tag{13}$$

the submatrix of $V[\hat{\phi}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{R}}]$ defined by its fifth to seventh rows and columns gives the covariance matrix $V[\hat{\boldsymbol{R}}]$ of $\Delta\boldsymbol{\Omega}$.

5. Analytical Solution

The initial guess of ϕ, $\boldsymbol{\tau}$, and $\boldsymbol{R}$ can be obtained by the following analytical procedure, which produces an exact solution in the absence of noise. First, we compute the homography matrix $\boldsymbol{H}$ by least squares as follows:

$$J_{LS} = \frac{1}{N}\sum_{\alpha=1}^{N} \|\boldsymbol{x}_\alpha \times \boldsymbol{H}\bar{\boldsymbol{x}}_\alpha\|^2 \to \min. \tag{14}$$

Since J_{LS} is a quadratic form in $\boldsymbol{H}$, the solution is immediately obtained up to scale by the standard eigenvalue analysis [4]. Hence, there exits a constant k such that

$$\boldsymbol{H} = k\boldsymbol{A}^{-1}\boldsymbol{R}^\top(\boldsymbol{I} - \boldsymbol{\tau}\boldsymbol{k}^\top), \tag{15}$$

where we define $\boldsymbol{A} = \mathrm{diag}(1, 1, \phi)$. Multiplying each side by $\boldsymbol{A}/k$ from left and then multiplying each side by its transpose, we obtain

$$\frac{1}{k^2}\boldsymbol{H}^\top\boldsymbol{A}^2\boldsymbol{H} = (\boldsymbol{I} - \boldsymbol{k}\boldsymbol{\tau}^\top - \boldsymbol{\tau}\boldsymbol{k}^\top + \|\boldsymbol{\tau}\|^2\boldsymbol{k}\boldsymbol{k}^\top). \tag{16}$$

Letting

$$\xi = \frac{1}{k^2}, \qquad \eta = \frac{\phi^2 - 1}{k^2}, \tag{17}$$

and equating (1,1), (2,2), and (1,2) elements on both sides of eq. (16), we obtain the following linear equation in ξ and η:

$$\begin{pmatrix} \|\boldsymbol{H}\boldsymbol{i}\|^2 & H_{31}^2 \\ \|\boldsymbol{H}\boldsymbol{j}\|^2 & H_{32}^2 \\ (\boldsymbol{H}\boldsymbol{i}, \boldsymbol{H}\boldsymbol{j}) & H_{31}H_{32} \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}. \tag{18}$$

Solving this by least squares, we obtain the nondimensional focal length ϕ in the following form:

$$\phi = \sqrt{1 + \frac{\eta}{\xi}}. \tag{19}$$

Equating (1,3) and (2,3) elements on both sides of eq. (16), we obtain τ_1 and τ_2 in the form

$$\begin{aligned} \tau_1 &= -\xi(\boldsymbol{H}\boldsymbol{i}, \boldsymbol{H}\boldsymbol{k}) - \eta H_{31}H_{33}, \\ \tau_2 &= -\xi(\boldsymbol{H}\boldsymbol{j}, \boldsymbol{H}\boldsymbol{k}) - \eta H_{32}H_{33}. \end{aligned} \tag{20}$$

Substituting these into the equation obtained by equating the (3,3) elements of eq. (16) on both sides, we obtain the following quadratic equation in τ_3:

$$\tau_3^2 - 2\tau_3 + (1 + \tau_1^2 + \tau_2^2 - \xi\|\boldsymbol{H}\boldsymbol{k}\|^2 - \eta H_{33}^2) = 0. \tag{21}$$

Since the camera is in front of the planar pattern, we choose from among the two solutions the one that satisfies $\tau_3 \leq 1$. The rotation $\boldsymbol{R}$ can be obtained from eq. (15). However, the resulting solution need not be an exact orthogonal matrix if there is noise

in the data. So, we apply the following least-squares minimization:

$$\left\| \boldsymbol{R} - \frac{1}{k}(\boldsymbol{I} - \boldsymbol{k}\boldsymbol{\tau}^\top)^{-1}\boldsymbol{H}^\top \boldsymbol{A} \right\|^2 \to \min. \qquad (22)$$

The solution is obtained analytically as follows. First, compute the following singular value decomposition:

$$\left(\boldsymbol{H}^\top + \frac{\boldsymbol{k}(\boldsymbol{H}\boldsymbol{\tau})^\top}{1-(\boldsymbol{k},\boldsymbol{\tau})}\right)\boldsymbol{A} = \boldsymbol{V}\begin{pmatrix}\sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3\end{pmatrix}\boldsymbol{U}^\top. \qquad (23)$$

Here, $\boldsymbol{V}$ and $\boldsymbol{U}$ are orthogonal matrix, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ (≥ 0) are the singular values. Then, the rotation matrix $\boldsymbol{R}$ that minimizes eq. (23) is given as follows [2]:

$$\boldsymbol{R} = \boldsymbol{V}\begin{pmatrix}1 & & \\ & 1 & \\ & & \det(\boldsymbol{V}\boldsymbol{U}^\top)\end{pmatrix}\boldsymbol{U}^\top. \qquad (24)$$

6. Simulation 1

Fig. 2 shows a simulated image of a grid pattern viewed from an angle. We added Gaussian random noise of mean 0 and standard deviation 1 (pixel) to the x and y coordinates of the vertices independently and computed the focal length and the motion parameters 1000 times, each time using different noise. Fig. 3(a) is the histogram of the computed focal length $\hat{f}$, and Fig. 3(b) is the corresponding histogram of the analytical solution. The vertical lines in the figures indicate the standard deviation (12) computed from the theoretical lower bound.

Fig. 4(a) is a 3-D plot of the distribution of the error vector $\Delta\boldsymbol{t} = \hat{\boldsymbol{t}} - \bar{\boldsymbol{t}}$ of translation, and Fig. 4(b) is the corresponding result for the analytical solution. The ellipses in the figures indicate the theoretical lower bound on the standard deviation in each orientation given by

$$(\Delta\boldsymbol{t}, V[\hat{\boldsymbol{\tau}}]^{-1}\Delta\boldsymbol{t}) = d^2, \qquad (25)$$

where the covariance matrix $V[\hat{\boldsymbol{\tau}}]$ is obtained from eq. (11).

Fig. 5(a) is a 3-D plot of the error vector $\Delta\boldsymbol{\Omega}$ of rotation defined by eq. (13), and Fig. 5(b) is the corresponding result for the analytical solution. The ellipses in the figures indicate the theoretical lower bound on the standard deviation in each orientation given by

$$(\Delta\boldsymbol{\Omega}, V[\hat{\boldsymbol{R}}]^{-1}\Delta\boldsymbol{\Omega}) = 1. \qquad (26)$$

We also computed

$$S_f = \sqrt{\frac{1}{n}\sum_{k=1}^{n}|\Delta f_k|^2}, \qquad \bar{S}_f = f_0\sqrt{V[\hat{\phi}]},$$

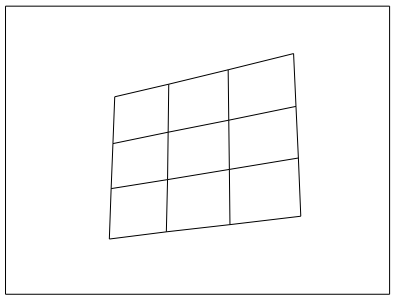
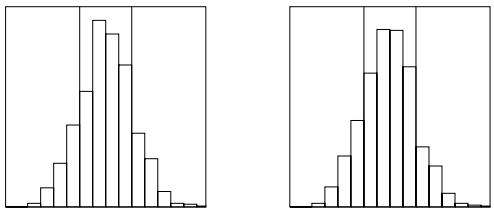
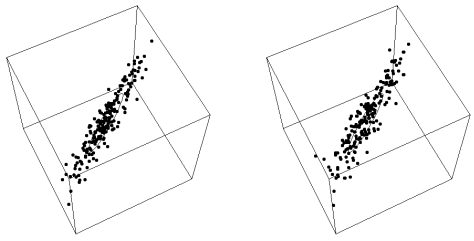


Figure 2: A simulated image of a grid pattern.



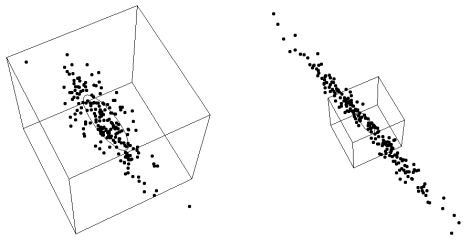
(a) (b)

Figure 3: Histogram of the computed focal length. (a) Optimal solution. (b) Analytical solution.



(a) (b)

Figure 4: Error distribution of the computed translation. (a) Optimal solution. (b) Analytical solution.



(a) (b)

Figure 5: Error distribution of the computed rotation. (a) Optimal solution. (b) Analytical solution.

Table 1: Standard deviations of the optimal and analytical solutions and their corresponding theoretical lower bounds.

	optimal	analytical	bound
focal length (pixles)	33.4	39.3	34.0
translation (cm)	32.9	38.5	32.6
rotation (deg)	0.413	1.292	0.414

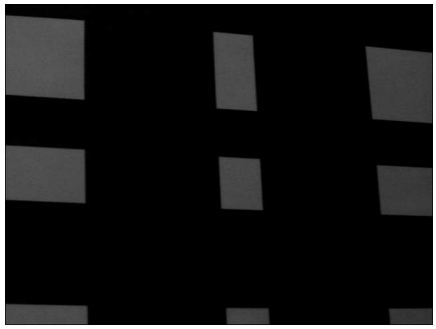


Figure 6: A real image of a grid pattern.

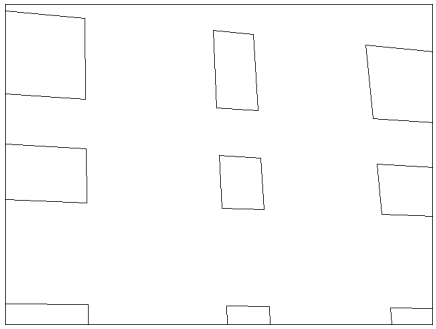


Figure 7: The grid pattern detected from Fig. 6.

$$S_{\mathbf{t}} = \sqrt{\frac{1}{n}\sum_{k=1}^{n} ||\Delta \boldsymbol{t}_k||^2}, \qquad \bar{S}_{\mathbf{t}} = d\sqrt{\mathrm{tr}V[\hat{\boldsymbol{\tau}}]},$$

$$S_{\mathbf{R}} = \sqrt{\frac{1}{n}\sum_{k=1}^{n} ||\Delta \boldsymbol{\Omega}_k||^2}, \qquad \bar{S}_{\mathbf{R}} = \sqrt{\mathrm{tr}V[\hat{\boldsymbol{R}}]}, \quad (27)$$

where the subscript k indicates values of the kth trial. In other words, S_f, $S_{\mathbf{t}}$, and $S_{\mathbf{R}}$ are the sample standard deviations of the focal length, the translation, and the rotation, respectively, and $\bar{S}_f$, $\bar{S}_{\mathbf{t}}$, and $\bar{S}_{\mathbf{R}}$ are their corresponding theoretical lower bounds. Table 1 lists values for 1000 trials

From these results, we can see that our optimization scheme improves the accuracy as compared with the initial analytical solution. We can also see that the accuracy almost attains the theoretical lower bound. This means that the reliability of the solution can be evaluated by computing the lower bounds.

7. Real Image Experiment 1

Fig. 6 is the input real image. We thresholded it into a binary image, traced the boundaries of the rectangular regions, segmented each of the resulting digital loops into four linear segments, and fitted lines to each by least squares. To the gray levels of the pixels near the fitted lines, we fitted new lines by least squares and determined the vertex positions by computing their intersections. Fig. 7 is the detected grid pattern. This is a small portion of a large grid pattern designed in such a way that the cross ratio of four neighboring lines is different everywhere in a statistically optimal manner [6, 7]. From a 3×3 grid in Fig. 7, we obtain $f = 1718$ (pixels).

Table 2: Estimated standard deviation.

focal length (pixels)	36.6
translation (cm)	3.44
rotation (deg)	0.253

Table 2 lists the standard deviations of the focal length, the translation, and the rotation evaluated by computing $f_0\sqrt{V[\hat{\phi}]}$, $d\sqrt{\mathrm{tr}V[\hat{\boldsymbol{\tau}}]}$, and $\sqrt{\mathrm{tr}V[\hat{\boldsymbol{R}}]}$, respectively. Thus, the reliability is automatically computed as a byproduct of the estimation procedure, and we need not know the true values of these parameters for this computation.

8. Simulation 2

We simulate a circular camera motion in a plane that makes angle 70° with the grid pattern shown in Fig. 2. The camera is rotated so that the center of the pattern is fixed at the center of the image frame. The focal length is assumed to be $f = 1200$ (pixels). Random Gaussian noise of mean 0 and standard deviation 1 (pixel) is added to each of the coordinate of the grid points independently at each frame. Fig. 8 plots the estimated focal length for the angle θ that the optical axis makes with the surface normal to the pattern. The dotted line indicates the true value, and the vertical line segments indicate the confidence interval defined by eq. (12).

Fig. 9 shows the computed positions of the camera. The reliability of each value is indicated by an ellipsoid defined by eq. (26) (enlarged three times). The dashed line indicates the true trajectory of the camera.

9. Real Image Experiment 2

Fig. 10 is a real image of a tennis court. Since the size of the court is stipulated by an international rule, we can compute the focal length and the camera position by using this knowledge. The focal length is estimated to be $f = 955$ (pixels). The camera is estimated to be at 627cm above the ground. Fig. 11 is the top view generated by mapping Fig. 10 via the computed homography. The estimated camera position is marked by a cross. Table 3 lists the standard deviations of the focal length, the translation, and the rotation.

The images of the poles and the persons in Fig. 11 can be regarded as their "shadows" on the ground created by a hypothetical light source at the camera position. Hence, we can compute their heights. The

right pole is estimated to be 113cm in height. The person near the camera is estimated to be 171cm tall. Our method can be applied to analyze video images of sports broadcasting; we can compute the focal length, the camera position, and 3-D structure of the scene at each frame [9, 10].

10. Concluding Remarks

This paper has described the procedure for simultaneous calibration of a freely moving and zooming camera frame by frame by placing a planar pattern in the scene. This scheme is more flexible than pre-calibration and more practical than self-calibration. We derived an analytical solution that is exact in the absence of noise and presented an iterative algorithm for statistical optimization based on a mathematical model of image noise. Our scheme automatically produces the reliability of the computed solution. We demonstrated the effectiveness of our scheme by simulation and real-image experiments.

References

[1] M. Betke and L. Gurvits, Mobile robot localization using landmarks, *IEEE Trans. Robotics Automation*, **13**-2 (1997), 251–263.

[2] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, 1993.

[3] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, 1996.

[4] K. Kanatani, Optimal homography computation with a reliability measure, *Proc. IAPR Workshop on Machine Vision Applications*, November 1998, Makuhari, Japan, pp. 426–429.

[5] K. Kanatani and N. Ohta, Optimal robot self-localization and accuracy bounds, *IEICE Trans. Inf. & Syst.*, **E82-D**-2 (1999), 447–452.

[6] Y. Kanazawa, C. Matsunaga and K. Kanatani, Best marker pattern design for recognition by cross ratio (in Japanese), *IPSJ SIG Notes*, 99-CVIM-115-13, March 1999, pp. 97–104.

[7] C. Matsunaga, K. Niijima and K. Kanatani, Best marker pattern design for recognition by cross ratio: Experimental investigation (in Japanese), *IPSJ SIG Notes*, 99-CVIM-115-14, March 1999, pp. 105–110.

[8] M. Polleyfes, R. Koch and L. Van Gool, Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, *Proc. Int. Conf. Comput. Vision*, January 1998, Bombay, India, pp. 90–95.

[9] I. Reid and A. Zisserman, Goal-directed video metrology, *Proc. 4th Euro. Conf. Comput. Vision*, April 1996, Cambridge, U.K., Vol. II, pp. 647–658.

[10] Y. Seo and K. S. Hong, Auto-calibration of a rotating and zooming camera, *Proc. IAPR Workshop on Machine Vision Applications*, November 1998, Makuhari, Chiba, Japan, pp. 274–277.

[11] K. T. Sutherland and W. B. Thompson, Localizing in unconstrained environment: Dealing with the errors, *IEEE Trans. Robotics Automation*, **10**-6 (1994), 740–754.

[12] R. Talluri and J. K. Aggarwal, Mobile robot self-location using model-image feature correspondence, *IEEE Trans. Robotics Automation*, **12**-1 (1996), 63–77.

[13] R. Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision methodology using off-the-shelf TV cameras and lenses, *IEEE J. Robotics Automation*, **3**-4 (1987), 323–344.

[14] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, Kluwer Academic, Dordrecht, The Netherlands, 1996.

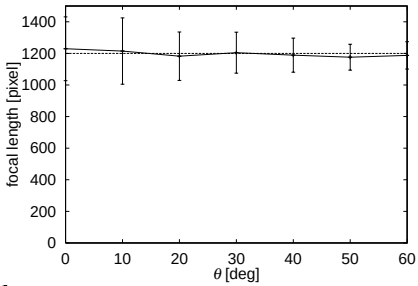


Figure 8: Estimated focal length and its reliability. The dashed line indicates the true value.

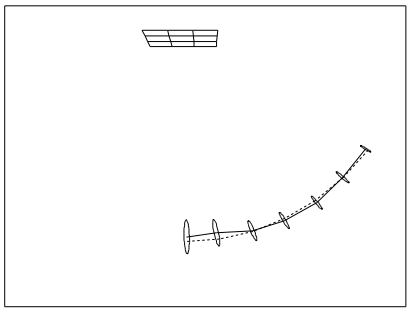


Figure 9: Estimated camera positions and their reliability. The dashed line indicates the true trajectory.

Figure 10: A real image of a tennis court.

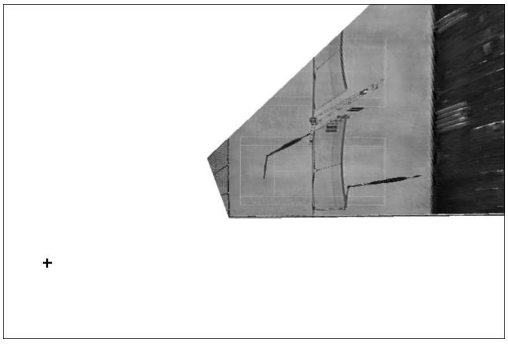


Figure 11: The computed camera position viewed from above.

Table 3: Estimated standard deviation.

focal length (pixel)	6.99
translation (cm)	16.1
rotation (deg)	0.151