PAPER *Special Issue on Image Recognition and Understanding*

A Method for Compensation of Image Distortion with Image Registration Technique*

Toru TAMAKI[†,††], ***Student Member***, **Tsuyoshi YAMAMURA**[†††], ***and*** **Noboru OHNISHI**[††,††††], ***Regular Members***

SUMMARY We propose a method for compensating distortion of image by calibrating intrinsic camera parameters by image registration which does not need point-to-point correspondence. The proposed method divides the registration between a calibration pattern and a distorted image observed by a camera into two steps. The first step is the straightforward registration from the pattern in order to correct the displacement due to projection. The second step is the backward registration from the observed image for compensating the distortion of the image. Both of the steps use Gauss-Newton method, a nonlinear optimization technique, to minimize residuals of intensities so that the pattern and the observed image become the same. Experimental results show the usefulness of the proposed method. Finally we discuss the convergence of the proposed method which consists of the two registration steps.

key words: *camera calibration, image distortion, image registration, intrinsic camera parameters, nonlinear optimization*

1. Introduction

1.1 Background

Calibrating a camera and compensating distortion of image are important processes for computer vision. Many researches (for examples, [2], [3]) formulate their problems without considering distortion because of simplicity. However, distortion is inevitable when we use an ordinary camera lens, and the displacement of a point due to the distortion is sometimes more than ten pixels. Although self-calibration have been studied recently, they don't consider barrel/pin-cushion distortion. So pre-calibration of intrinsic camera parameters and correcting distorted image are required.

Some codes of calibration have been available via internet (e.g., Tsai's method [4] is available from [5]), however, such ordinary techniques require a lot of correspondences between points on an image and known three-dimensional coordinates (on a plane or on some structure like a cube or a house) to estimate parameters of transformation of the corresponding points.

When the correspondences should be done manually, such a method is not reliable because of human errors. Moreover, it takes much time and patience, so it is too hard to measure a change of the distortion parameters, for example, as changing camera zooming.

An alternative procedure is detecting markers such as corners or intersections. It can be done by a template matching technique and maybe sub-pixel order can be achieved. However, another correspondence problem arises; which marker on the image corresponds to which point in a space. It is hard to be neglected as the number of the markers increases in order to improve the accuracy of estimation. If the problem can be avoided [6], the number of points used for correspondence is limited.

In this paper, we propose a new calibration method which compensates distortion of image due to change of intrinsic camera parameters. The proposed method makes the correspondence between images, so that more precise estimation than the marker detection is achieved because this method uses not several points of markers but all points of the image. This is an image registration technique which is used in the area of motion analysis. The proposed method consists of the following three procedures. The first step is to roughly transform the pattern onto the image, which is represented by affine parameters. Then the accurate parameters of transformation of plane under perspective projection are estimated with a nonlinear optimization technique minimizing residuals between two images. Finally the distortion parameters are estimated to minimize residuals that remain after applying the previous step due to distortion.

1.2 Image Registration for Distortion

The basic idea is that calibration needs point-to-point correspondence and image registration can supply such correspondences of all points in an image. The proposed method makes correspondence between an ideal calibration pattern I_1 and a distorted image I_2 of the printed pattern observed by a camera. The observation is modeled by two transformations (see Fig. 1); one is projection $\boldsymbol{u}$ and the other is distortion $\boldsymbol{d}$ (the inverse of $\boldsymbol{f}$). Then, using the image registration technique,

Manuscript received September 29, 2000.
Manuscript revised March 16, 2001.

[†] The author is with the Department of Information Engineering, Nagoya University, Nagoya-shi, 464–8603 Japan.

[††] The authors are with the BMC, RIKEN, Nagoya-shi, 463–0003 Japan.

[†††] The author is with the Faculty of Information Science Technology, Aichi Prefectural University, Aichi-ken, 480–1198 Japan.

[††††] The author is with the Center for Information Media Studies, Nagoya University, Nagoya-shi, 464–8601 Japan.

[*] This paper was presented in part at SCI2000 [1].

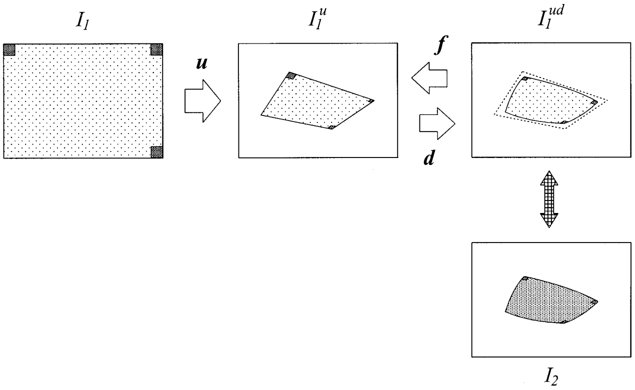


Fig. 1 Diagram of image transformation modeling. I_1 is the calibration pattern and I_2 is the observed image. Hatched arrow represents the registration.

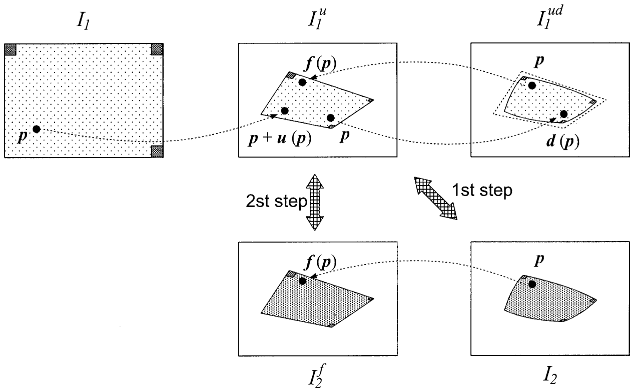


Fig. 2 Division of the registration into two steps.

the proposed method minimizes the difference between I_1 and I_2; that is, the square of residuals of intensities of the two images.

The procedure of the proposed method is as follows: At first, create the calibration pattern I_1. Any digital image (taken by a digital camera, scanned photo or CG) can be used as the pattern. Next, print the pattern on a sheet by a printer[†]. Then, take the image of the pattern, I_2, by the camera to be calibrated. Finally, we make registration between the pattern I_1 and the observed image I_2 through the two transformations $\boldsymbol{u}$ and $\boldsymbol{d}$. As shown in Fig. 1, let I_1^u be the image which is made from I_1 by applying $\boldsymbol{u}$, and I_1^{ud} be from I_1^u with $\boldsymbol{d}$. The registration finds the parameters of $\boldsymbol{u}$ and $\boldsymbol{d}$ so that the transformed calibration pattern I_1^{ud} becomes the same with the observed image I_2.

Some researchers use image registration to estimate extrinsic camera parameters for calibration [7] or mosaicing [8]. The problem to employ image registration strategy in a straightforward manner is that $\boldsymbol{f}$, the transformation from I_1^{ud} to I_1^u, is a nonlinear function (as you see in later section). This means that the inverse $\boldsymbol{d}$ is not a closed-form but something implemented by an iterative procedure. Usually a gradient based nonlinear optimization method is used for image registration, however, the gradient/Jacobian of the transformation $\boldsymbol{d}$ is not available because of the reason mentioned above. Therefore, we can not perform the registration between I_1 and I_2 directly.

To overcome this problem, we propose dividing the registration into two steps as shown in Fig. 2; one is from I_1, the other is from I_2. At the first step, the registration is straightforward and doesn't consider the effect of distortion. It finds the parameters of $\boldsymbol{u}$ so that the transformed calibration pattern I_1^u becomes similar to the observed image I_2.

On the other hand, at the second step the registration is performed in inverse direction and only considers the effect of distortion. It finds the parameters of $\boldsymbol{f}$ so that I_2^f, the transformed image of I_2 by applying $\boldsymbol{f}$, becomes similar to I_1^u which is obtained by the first registration.

This means that the comparisons of images are done at the intermediate stages (middle column of Fig. 2) from both sides of the transformations. Therefore, this approach enable us to avoid the unavailability of gradient/Jacobian and to make registration between the pattern and the observed image indirectly.

In Sect. 2, we explain about the preliminary step performed before the registration steps. Then the first and the second step of the registrations are described in Sects. 3 and 4. Experimental results are shown in Sect. 7, and the convergence of the proposed method is discussed in Sect. 8.

2. Rough Matching with Affine Transformation

At first, a calibration pattern image I_1 should be created. It must have three color (r, g, b) markers at coordinates $\boldsymbol{m}_r$, $\boldsymbol{m}_g$, $\boldsymbol{m}_b$ on the corners, then the pattern is printed with colors.

Then the camera, which is to be calibrated, takes an image of the printed pattern, I_2. The coordinates of the markers $\boldsymbol{m}'_r, \boldsymbol{m}'_g, \boldsymbol{m}'_b$ in the image I_2 are detected by thresholding or template matching.

Here we calculate the parameters of the transformation from the pattern I_1 into the image I_2. This transformation is represented by six affine parameters $\boldsymbol{\theta}^a = (\theta_1^a, \dots, \theta_6^a)^T$.

Let $\boldsymbol{p} = (x, y)^T$ be a point on I_1, and $\boldsymbol{p} + \boldsymbol{a}(\boldsymbol{p}; \boldsymbol{\theta}^a)$ be the point on I_2 corresponding to $\boldsymbol{p}$, where

$$\boldsymbol{a}(\boldsymbol{p}; \boldsymbol{\theta}^a) = \begin{pmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{pmatrix} \boldsymbol{\theta}^a \tag{1}$$

and solve the following system of linear equations

$$\boldsymbol{m}'_i = \boldsymbol{m}_i + \boldsymbol{a}(\boldsymbol{m}_i; \boldsymbol{\theta}^a), \quad i = r, g, b \tag{2}$$

to obtain the affine parameters $\boldsymbol{\theta}^a$. Here we model the

[†] We assume that a printer provide us an ideal print.

displacement of two corresponding points as an affine transformation (a modification of ordinary affine form) because it is convenient to use affine parameters as the initial value of the first registration step when the model of this step and that of the first step have the same form.

In fact, this preliminary step can be skipped when I_2 is similar enough to I_1.

3. First Step; Forward Registration of Projection

The preliminary step described above uses only three corresponding points for the affine parameters. Here we make an image registration, using plane projective transformation.

As mentioned above, the registration in this step doesn't consider the effect of distortion but deals with the displacement of the pattern in the image. The parameters of $\boldsymbol{u}$ are estimated so that the transformed calibration pattern I_1^u becomes similar to the observed image I_2.

3.1 Modeling and Formulation

Suppose that a point $\boldsymbol{p}$ in I_1 is transformed by $\boldsymbol{u}$ to $\boldsymbol{p}+\boldsymbol{u}$ in I_1^u (the intensities of the two points are the same). The difference between the intensity of $\boldsymbol{p}+\boldsymbol{u}$ in I_1^u and that of $\boldsymbol{p}+\boldsymbol{u}$ in I_2 (at the same location) is expected to be small.

Hence, what to do is minimizing the residuals of intensities between $\boldsymbol{p}_i$ in I_1 and $\boldsymbol{p}_i+\boldsymbol{u}(\boldsymbol{p}_i;\boldsymbol{\theta}^u)$ in I_2;

$$r_i^u = I_1(\boldsymbol{p}_i) - I_2(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i;\boldsymbol{\theta}^u)) \tag{3}$$

where $\boldsymbol{\theta}^u = (\theta_1^u, \dots, \theta_8^u)^T$ and

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{p};\boldsymbol{\theta}^u) &= \begin{pmatrix} x & y & 0 & 0 & 1 & 0 & x^2 & xy \\ 0 & 0 & x & y & 0 & 1 & xy & y^2 \end{pmatrix}\boldsymbol{\theta}^u \\ &= M^u(\boldsymbol{p})\boldsymbol{\theta}^u, \quad \boldsymbol{p} = (x,y)^T \end{aligned} \tag{4}$$

The function to be minimized is the square of the residuals.

$$\min_{\boldsymbol{\theta}^u} \sum_i \rho(r_i^u), \quad \rho(r) = r^2 \tag{5}$$

Here $\boldsymbol{u}()$ is a 2D quadratic model with eight parameters of instant motion of a planar object and often used for motion analysis [9]. This model can deal with the case that the rotation of two viewpoints is small and the depth change of the planar object is relatively small as compared with the distance between the object and the camera. Thus $\boldsymbol{u}$ is not an exact planar perspective motion model which has anther parameter sets [10]. Although we have not evaluated the error to employ this model, the effect can be negligible because in fact, in order to make the area for the registration large, the printed pattern appeared in I_2 becomes large enough (almost fills in the image) and the pattern is placed almost parallel to the image plane so that the difference between I_1 and I_2 is small. And also we don't need extrinsic camera parameters themselves to compensate distortion.

3.2 Minimization

Estimating the parameters $\boldsymbol{\theta}^u$, the objective function (5) is minimized by Gauss-Newton method, an nonlinear optimization technique [11]. The parameters are updated from some initial value by the following rule.

$$\boldsymbol{\theta}^u \leftarrow \boldsymbol{\theta}^u + \delta\boldsymbol{\theta}^u \tag{6}$$

We use $\boldsymbol{\theta}^a$, the affine parameters obtained at the preliminary step, as the initial value of the first six elements of $\boldsymbol{\theta}^u$. The last two of $\boldsymbol{\theta}^u$ are initialized to 0. If the preliminary step was skipped, all elements of $\boldsymbol{\theta}^u$ are initialized to 0.

The decent direction $\delta\boldsymbol{\theta}^u$ is calculated as follows [11]; †

$$\delta\boldsymbol{\theta}^u = -(J\tilde{D}J^T)^{-1}J\tilde{D}\boldsymbol{r}^u \tag{7}$$

$$J = \frac{\partial \boldsymbol{r}^u}{\partial \boldsymbol{\theta}^u} = \left[\frac{\partial r_i^u}{\partial \theta_j^u}\right] \tag{8}$$

$$\tilde{D} = \mathrm{diag}\left[\frac{\dot{\rho}(r_i^u)}{r_i^u}\right] \tag{9}$$

$$\dot{\rho}(r_i^u) = \left.\frac{\partial \rho(r)}{\partial r}\right|_{r=r_i^u} \tag{10}$$

This is the same as the least square formulation, that is, the system of linear equations [9] which is written as

$$\sum_{l,i} \frac{\dot{\rho}(r_i^u)}{r_i^u}\frac{\partial r_i^u}{\partial \theta_k^u}\frac{\partial r_i^u}{\partial \theta_l^u}\delta\theta_l^u = -\sum_i \frac{\dot{\rho}(r_i^u)}{r_i^u} r_i^u \frac{\partial r_i^u}{\partial \theta_k^u} \tag{11}$$

for $k = 1, \dots, 8$. The partial derivatives are the elements of the following Jacobian obtained by the chain rule of differential.

$$\frac{\partial r^u}{\partial \boldsymbol{\theta}^u} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}^u}\frac{\partial r^u}{\partial \boldsymbol{u}} = -\{M^u(\boldsymbol{p})\}^T \nabla I_2(\boldsymbol{p}+\boldsymbol{u}(\boldsymbol{p})) \tag{12}$$

The iteration of calculating $\delta\boldsymbol{\theta}^u$ in Eq. (6) is repeated until it converges. At each iteration, the parameters estimated at the previous iteration are used for the calculation of $\boldsymbol{u}(\boldsymbol{p})$.

In the following section, let $\hat{\boldsymbol{\theta}}^u$ be the estimates of $\boldsymbol{\theta}^u$ after the iteration stops.

4. Second Step; Backward Registration of Distortion

At the end of the first step, the image registration be-

† The reason to introduce $\tilde{D}$ is to make it easy to deal with a robust function as ρ instead of least-square when the pattern in the image are partially occluded or out of scene.

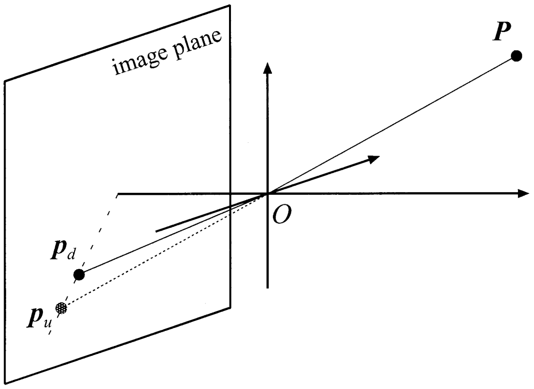


Fig. 3 Model of distortion.

tween the pattern image and the pattern in the distorted image is performed without considering the effect of distortion.

In fact, the effect of approximation of 2D motion model exists, however, the radial distortion still remains even if the other model is employed. Modeling distortion of image usually consists of two successive processes [12]–[14] as shown in Fig. 3; at first a point $\boldsymbol{P}$ in three dimensional space is projected onto the image plane, then the projected point $\boldsymbol{p}_u$ in the image plane is slightly moved to $\boldsymbol{p}_d$ because of the radial distortion.

The effect of 2D model arises from the difference between the model of projection, however, it can be negligible as described before.

4.1 Modeling of Distortion

The relationship between undistorted and distorted coordinates in an image is usually modeled by the following five intrinsic camera parameters [12], [13], [15]; the radial distortion parameters κ_1 and κ_2, coordinate of image center $(c_x, c_y)^T$, and the aspect ratio (horizontal scale) s_x. We write these parameters as $\boldsymbol{\theta}^d = (\kappa_1, \kappa_2, c_x, c_y, s_x)^T$. Here we consider only radial distortion, however, the following discussion can be applied when another model involving decentering distortion [14] is employed.

The distortion is represented usually with a coordinate system which has its origin at $(c_x, c_y)^T$, while the system used in the previous section has the origin at top-left corner. Therefore, we introduce another notation. Let $\boldsymbol{p}_u = (x_u, y_u)^T$ and $\boldsymbol{p}_d = (x_d, y_d)^T$ be points in the un-distorted and distorted images, both of them have their origins at the top-left corner of their images. Here we have two functions between $\boldsymbol{p}_u$ and $\boldsymbol{p}_d$ about the system of top-left corner origin,

$$\boldsymbol{p}_d = \boldsymbol{d}(\boldsymbol{p}_u; \boldsymbol{\theta}^d) \tag{13}$$

$$\begin{aligned}\boldsymbol{p}_u &= \boldsymbol{f}(\boldsymbol{p}_d; \boldsymbol{\theta}^d)\\ &= \begin{pmatrix} \frac{x_d - c_x}{s_x}(1 + \kappa_1 R'^2 + \kappa_2 R'^4) + c_x \\ (y_d - c_y)(1 + \kappa_1 R'^2 + \kappa_2 R'^4) + c_y \end{pmatrix}\end{aligned} \tag{14}$$

where

$$R' = \sqrt{\left(\frac{x_d - c_x}{s_x}\right)^2 + (y_d - c_y)^2} \tag{15}$$

$\boldsymbol{f}$ and $\boldsymbol{d}$ are the inverse of each other, but $\boldsymbol{d}$ is not a closed-form function of $\boldsymbol{p}_u$ and $\boldsymbol{d}$ is implemented by an iterative procedure [12] as we addressed previously.

Anyway, we can write the transformation between images using Eqs. (13) and (14), and the following equations about intensities of corresponding points hold.

$$I_1(\boldsymbol{p}) = I_1^u(\boldsymbol{p} + \boldsymbol{u}(\boldsymbol{p}; \boldsymbol{\theta}^u); \boldsymbol{\theta}^u)) \tag{16}$$

$$I_1^u(\boldsymbol{p}; \boldsymbol{\theta}^u)) = I_1^{ud}(\boldsymbol{d}(\boldsymbol{p}; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u, \boldsymbol{\theta}^d) \tag{17}$$

$$I_1^u(\boldsymbol{f}(\boldsymbol{p}; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u) = I_1^{ud}(\boldsymbol{p}; \boldsymbol{\theta}^u, \boldsymbol{\theta}^d) \tag{18}$$

4.2 Minimization with Backward Registration

Since the transformation from I_1 to I_1^{ud} is not closed-form, now consider a backward registration. As we described in Sect. 1, the difference between I_2 with I_1^{ud} is expected to be small, that is, minimize the square of residuals of intensities between two images;

$$r_i^d = I_2(\boldsymbol{p}_i) - I_1^{ud}(\boldsymbol{p}_i; \boldsymbol{\theta}^u, \boldsymbol{\theta}^d) \tag{19}$$

But the point $\boldsymbol{p}_i$ in I_1^{ud} corresponds to $\boldsymbol{f}(\boldsymbol{p})$ in I_1^u as shown in Fig. 2. So using Eq. (18), this equation can be rewritten as follows;

$$r_i^d = I_2(\boldsymbol{p}_i) - I_1^u(\boldsymbol{f}(\boldsymbol{p}_i; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u) \tag{20}$$

Note that since a point $\boldsymbol{p}$ in I_2 is transformed to $\boldsymbol{f}(\boldsymbol{p})$ in I_2^f, the residual defined by the intensity of the point $\boldsymbol{f}(\boldsymbol{p})$ in I_1^u and that of $\boldsymbol{f}(\boldsymbol{p})$ in I_2^f is equal to Eq. (20). This is shown in Fig. 4.

Hence, the estimation method becomes the same one in the previous step. The minimization is done about the following function with Gauss-Newton method.

$$\min_{\boldsymbol{\theta}^d} \sum_{i \in \Omega} \rho(r_i^d) \tag{21}$$

where $\Omega = \{i \mid \boldsymbol{p}_i \in I_2, \exists \boldsymbol{p} \in I_1, \boldsymbol{f}(\boldsymbol{p}_i) = \boldsymbol{p} + \boldsymbol{u}(\boldsymbol{p})\}$, which means that the minimization should use points in I_2 within the region corresponding to the pattern I_1 (the gray area in the figure).

The system of equations to be solved is the same form with Eq. (11);

$$\sum_{l,\ i \in \Omega} \frac{\dot{\rho}(r_i^d)}{r_i^d} \frac{\partial r_i^d}{\partial \theta_k^d} \frac{\partial r_i^d}{\partial \theta_l^d} \delta\theta_l^d = -\sum_{i \in \Omega} \frac{\dot{\rho}(r_i^d)}{r_i^d} r_i^d \frac{\partial r_i^d}{\partial \theta_k^d} \tag{22}$$

and the derivatives in Eq. (22) are derived as follows;

$$\frac{\partial r^d}{\partial \boldsymbol{\theta}^d} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}^d} \frac{\partial r^d}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}^d}(-\nabla I_1^u(\boldsymbol{f}(\boldsymbol{p}); \hat{\boldsymbol{\theta}}^u)) \tag{23}$$

According to Eq. (14), the Jacobian is

$$\frac{\partial \boldsymbol{f}(\boldsymbol{p}_d)}{\partial \boldsymbol{\theta}^d} = \begin{pmatrix} R'^2 \frac{x_d - c_x}{s_x} & R'^2(y_d - c_y) \\ R'^4 \frac{x_d - c_x}{s_x} & R'^4(y_d - c_y) \\ \frac{\partial x_u}{\partial c_x} & \frac{\partial y_u}{\partial c_x} \\ \frac{\partial x_u}{\partial c_y} & \frac{\partial y_u}{\partial c_y} \\ \frac{\partial x_u}{\partial s_x} & \frac{\partial y_u}{\partial s_x} \end{pmatrix} \tag{24}$$

where

$$\frac{\partial x_u}{\partial c_x} = 1 - \frac{1}{s_x}(1 + \kappa_1 R'^2 + \kappa_2 R'^4) - 2(\kappa_1 + 2\kappa_2 R'^2)\frac{(x_d - c_x)^2}{{s_x}^3} \tag{25}$$

$$\frac{\partial y_u}{\partial c_x} = -2(\kappa_1 + 2\kappa_2 R'^2)\frac{x_d - c_x}{{s_x}^2}(y_d - c_y) \tag{26}$$

$$\frac{\partial x_u}{\partial c_y} = -2(\kappa_1 + 2\kappa_2 R'^2)\frac{x_d - c_x}{s_x}(y_d - c_y) \tag{27}$$

$$\frac{\partial y_u}{\partial c_y} = 1 - (1 + \kappa_1 R'^2 + \kappa_2 R'^4) - 2(y_d - c_y)^2(\kappa_1 + 2\kappa_2 R'^2) \tag{28}$$

$$\frac{\partial x_u}{\partial s_x} = \frac{-(x_d - c_x)}{{s_x}^2}(1 + \kappa_1 R'^2 + \kappa_2 R'^4) - 2(\kappa_1 + 2\kappa_2 R'^2)\frac{(x_d - c_x)^3}{{s_x}^4} \tag{29}$$

$$\frac{\partial y_u}{\partial s_x} = -2(y_d - c_y)(\kappa_1 + 2\kappa_2 R'^2)\frac{(x_d - c_x)^2}{{s_x}^3} \tag{30}$$

Initial parameters to solve Eq. (22) are set as follows; c_x and c_y to half of the width and the height of I_2, $\kappa_2 = 0$ and $s_x = 1$. On the other hand, κ_1 is randomly initialized to avoid that all of Eqs. (25) – (28) and (30) become 0 by initializing $\kappa_1 = \kappa_2 = 0$. We choose empirically $\kappa_1 \in [-10^{-7}, 10^{-7}]$.

In the following section, let $\hat{\boldsymbol{\theta}}^d$ be the estimates of $\boldsymbol{\theta}^d$ after the iteration stops.

5. Iteration of Two Steps with Coarse-to-Fine

To reduce computation time, and to obtain an accurate estimation even when there is a relatively large residual in the initial state, the coarse-to-fine strategy is employed. The procedures mentioned above are applied to a filtered image which is much blurred at first and then gradually becomes sharpened. Therefore, the first and the second steps are repeated in turn as changing the resolution of the images I_1 and I_2.

The first step should be modified appropriately to involve the parameters $\hat{\boldsymbol{\theta}}^d$ estimated in the second step. In the previous formulation, the parameters of $\boldsymbol{u}$ are estimated so that I_1^u becomes similar to I_2, but now the estimation of the first step is changed so that I_1^u becomes similar to I_2^f, the corrected image of I_2 (see Fig. 4).

The residual Eq. (3) is modified as

$$r_i^u = I_1(\boldsymbol{p}_i) - I_2^f(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i; \boldsymbol{\theta}^u); \hat{\boldsymbol{\theta}}^d) \tag{31}$$

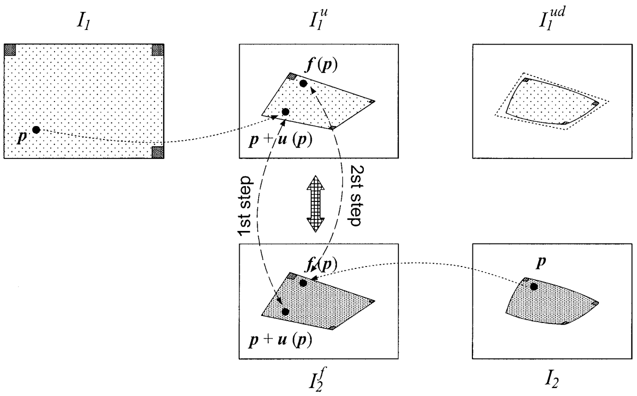

I_1
I_1^u
I_1^{ud}
$\boldsymbol{p}$
$\boldsymbol{f}(\boldsymbol{p})$
$\boldsymbol{p} + \boldsymbol{u}(\boldsymbol{p})$
1st step
2st step
$\boldsymbol{f}(\boldsymbol{p})$
$\boldsymbol{p} + \boldsymbol{u}(\boldsymbol{p})$
$\boldsymbol{p}$
I_2^f
I_2

Fig. 4 Relation between two steps repeated by coarse-to-fine. The arrows indicate points corresponding each other used in each step.

and the Jacobian Eq. (12) as

$$\frac{\partial r^u}{\partial \boldsymbol{\theta}^u} = -\{M^u(\boldsymbol{p})\}^T \nabla I_2^f(\boldsymbol{p} + \boldsymbol{u}(\boldsymbol{p}; \boldsymbol{\theta}^u); \hat{\boldsymbol{\theta}}^d) \tag{32}$$

where I_2^f is the image transformed by applying $\boldsymbol{f}$ with $\hat{\boldsymbol{\theta}}^d$ to I_2 as;

$$I_2^f(\boldsymbol{f}(\boldsymbol{p}; \hat{\boldsymbol{\theta}}^d); \hat{\boldsymbol{\theta}}^d) = I_2(\boldsymbol{p}) \tag{33}$$

$$I_2^f(\boldsymbol{p}; \hat{\boldsymbol{\theta}}^d) = I_2(\boldsymbol{d}(\boldsymbol{p}; \hat{\boldsymbol{\theta}}^d)) \tag{34}$$

6. Some Strategies

6.1 Interpolation of Pixel Value

When we want to obtain an intensity of pixel whose coordinate is not on the integer grid, and it occurs frequently, we need to interpolate the intensity using the values of the pixels that are already located on the grid. We use the bilinear interpolation[16, p.382], an easy and simple method, which interpolate with the values of four neighbor pixels on a rectilinear grid.

6.2 Histogram Matching

In general, intensity of the pattern in the image observed by a camera is different from that of the original one, and I_1^{ud} can not be identical to I_2 in a strict sense. So we use histogram matching technique[17, p.93] which changes the histogram of the observed image I_2 so that it becomes identical to that of the distorted pattern I_1^{ud}.

7. Experimental Results

We conducted experiments with the proposed method using real images which is taken by a camera with zoom lens. We use a scanned photograph as the calibration pattern (shown in Fig. 5), and print it on a sheet by a

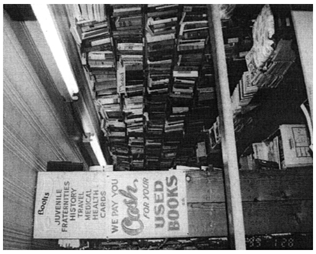


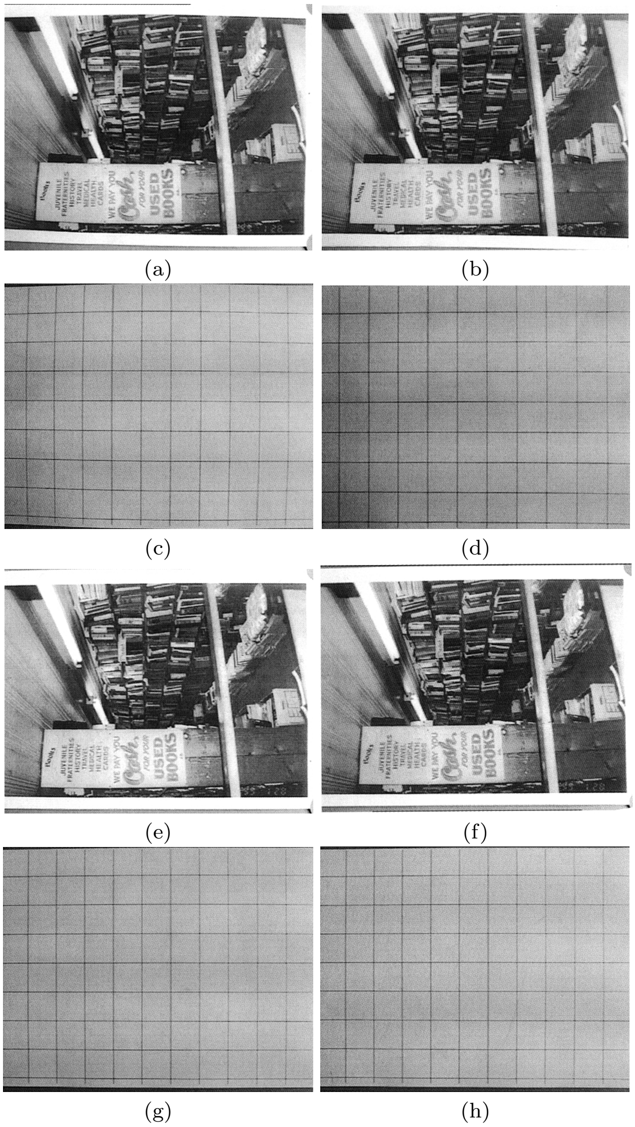
Fig. 5 The calibration pattern (640 × 480).

laser monochrome printer (EPSON LP-9200PS2), then capture images of the printed output by a CCD camera (Sony EVI-D30) with a capturing software (of SGI O2). As fixing the camera zooming, we placed the sheet of the pattern in front of the camera almost parallel to the image plane so that the image is filled with the pattern. We took two images of the printed pattern as changing the zooming of the camera as shown in Figs. 6 (a) and (e). At each setting, we took also the image of a grid pattern (shown in Figs. 6 (c) and (g)) to show the effect of distortion more visually. As you see, the wider the view angle is, the larger the effect of barrel distortion becomes (the lines of the grid pattern in Fig. 6 (a) curve more tightly than those in Fig. 6 (g)). In these experiments, we skipped the preliminary step described in Sect. 2.

The results of estimation of the intrinsic camera parameters $\boldsymbol{\theta}^d$ are shown in Table 1. The right column of Fig. 6 shows the images compensated by Eq. (14) with the estimated parameters. In the compensated figures, the curved lines in the grid pattern are corrected to straight lines, so the proposed method works well. The computational time was about twenty minutes for about 50 iterations including both of the two steps, however, the optimization almost converged by 15 iterations.

The results have some errors as shown in the right column of Fig. 6. The lines of grid still slightly curves especially around the corners of the image, because the gradation of illumination of the image (the left side is darker and the right is brighter) can not be removed thoroughly by the simple histogram transformation. The histogram matching method should be replaced with an estimation of illumination change by some method, such as a linear brightness constraint [19].

To see the usefulness of the proposed method, we compared with the results by another method; the camera calibration toolbox for MATLAB included in the Open Source Computer Vision Library released by Intel [18]. This method needs several images of the checkerboard (no high/low limits are indicated but their example uses 20 images), and detects the corners of the squares based on the coordinates of the four corners of the checkerboard which are indicated by user (in this case, 4 corners × 20 images = 80 points on the images should be clicked by a mouse). Their model of distortion includes parameters of the forth order of radial and upto the second order of tangential distortion. Although the model is different from ours and the comparison of the estimates of intrinsic parameter



(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Fig. 6 Experimental results of the proposed method. (a) the image of the calibration pattern taken by the camera at the widest view angle. (b) compensated image of (a). (c) the image of the grid pattern taken at the widest view angle. (d) compensated image of (c). (e) the image of the calibration pattern taken by the camera at the second-widest view angle. (f) compensated image of (e). (g) the image of the grid pattern taken at the second-widest view angle. (h) compensated image of (g).

Table 1 Estimation results at two view angle.

	Fig. 6 (a)	Fig. 6 (e)
κ_1	2.804e-07	-6.7631e-08
κ_2	2.992e-13	5.219e-13
c_x	327.8	326.6
c_y	214.3	184.2
s_x	0.9954	0.9997

Table 2 Two steps of the estimations at step k and $k+1$.

	Estimate $\boldsymbol{\theta}^u$	Estimate $\boldsymbol{\theta}^d$
step k	$\boldsymbol{\theta}^u_k = \operatorname*{argmin}_{\boldsymbol{\theta}^u} L^u(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d_{k-1})$	$\boldsymbol{\theta}^d_k = \operatorname*{argmin}_{\boldsymbol{\theta}^d} L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d)$
step $k+1$	$\boldsymbol{\theta}^u_{k+1} = \operatorname*{argmin}_{\boldsymbol{\theta}^u} L^u(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d_k)$	$\boldsymbol{\theta}^d_{k+1} = \operatorname*{argmin}_{\boldsymbol{\theta}^d} L^d(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d)$

Table 3 Inequalities obtained by each estimation step.

$L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_{k-1})$	$>$	$L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_k)$	at step k of estimating $\boldsymbol{\theta}^d$ fixing $\boldsymbol{\theta}^u_k$
$L^u(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_k)$	$>$	$L^u(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d_k)$	at step $k+1$ of estimating $\boldsymbol{\theta}^u$ fixing $\boldsymbol{\theta}^d_k$
$L^d(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d_k)$	$>$	$L^d(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d_{k+1})$	at step $k+1$ of estimating $\boldsymbol{\theta}^d$ fixing $\boldsymbol{\theta}^u_{k+1}$

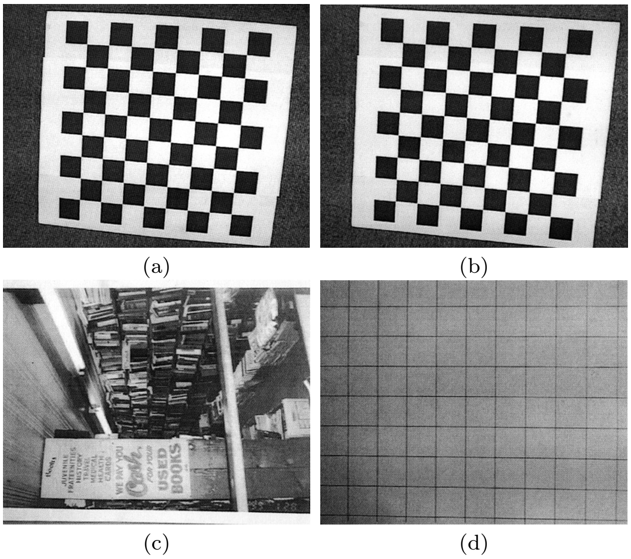

(a) (b) (c) (d)

Fig. 7 Experimental results by another method [18]. (a) checkerboard used for calibration. (b) corrected image of the checkerboard with estimated intrinsic parameters. (c) corrected image of Fig. 6 (a). (d) corrected image of Fig. 6 (c).

is difficult, the distorted images can be corrected by both method.

Figure 7 shows the experimental results of the method. Figure 7 (a) is one of the images used to calibration, and Fig. 7 (b) is the corrected image. Figure 7 (a) is captured with the same parameters with that of Fig. 6 (a) (that is, the widest view angle), so Figs. 6 (a) and (c) can be corrected with the parameters estimated by this method. Figures 7 (c) and (d) show the corrected images with the parameters of this method.

The computational time of our method is longer than that of this method because our proposed method is based on a nonlinear optimization. As for the corrected images, the results of both methods are similar each other. However, our method is much better for a batch process when the parameters are estimated as changing camera zooming. It is because the proposed method needs only one image of the calibration pattern and also our method doesn't need any involvements by user if the preliminary step can be skipped. Methods which need numbers of point correspondences would be a burden for a user.

8. Discussions

8.1 Convergence

In this section, we discuss the convergence of the proposed two-step method. In proposed method, the image registration is divided into two steps. So the result of estimation at the second step depends on that of the first step, and the estimation of this method may not converge. Here we discuss below the convergence of the method under some assumption.

The two step estimation procedures at steps[†] k and $k+1$ are summarized in Table 2, where $\boldsymbol{\theta}^u_k$ and $\boldsymbol{\theta}^d_k$ are the estimates of $\boldsymbol{\theta}^u$ and $\boldsymbol{\theta}^d$ at step k, respectively (hats are omitted), and L^u, L^d are the residuals with the parameters as follows;

$$L^u(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = \sum_i \rho(r^u_i(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d)) \tag{35}$$

$$L^d(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = \sum_j \rho(r^d_j(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d)) \tag{36}$$

$$r^u_i(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = I_1(\boldsymbol{p}_i) - I^f_2(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i; \boldsymbol{\theta}^u); \boldsymbol{\theta}^d) \tag{37}$$

$$r^d_j(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = I_2(\boldsymbol{p}_j) - I^u_1(\boldsymbol{f}(\boldsymbol{p}_j; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u) \tag{38}$$

At each estimation step, for example, estimating $\boldsymbol{\theta}^d$ at step k, the minimization of $L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d)$ about $\boldsymbol{\theta}^d$ are done by fixing $\boldsymbol{\theta}^u_k$ with $\boldsymbol{\theta}^d_{k-1}$ as the initial state. Then we obtain the estimates $\boldsymbol{\theta}^d_k$, and $L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_k)$ should be smaller than $L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_{k-1})$. We obtain this condition at each estimation step, and they are shown in Table 3.

Here, consider whether both of the following two inequalities

$$L^u(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_{k-1}) > L^u(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d_k) \tag{39}$$

$$L^d(\boldsymbol{\theta}^u_k, \boldsymbol{\theta}^d_k) > L^d(\boldsymbol{\theta}^u_{k+1}, \boldsymbol{\theta}^d_{k+1}) \tag{40}$$

that is, whether the residuals become smaller If they hold, both of the residuals become small and the iteration of the proposed method converges.

[†] Here "steps" means the number of iterations, and one iteration is a pair of the first and the second steps of the estimations.

To derive the above inequalities, we assume that the following propositions

$$\boldsymbol{p}_i \in I_1 \Longleftrightarrow \boldsymbol{p}'_i \in I_1^u, \quad \boldsymbol{p}'_i = \boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i) \tag{41}$$

$$\boldsymbol{p}_j \in I_1^{ud} \Longleftrightarrow \boldsymbol{p}'_j \in I_1^u, \quad \boldsymbol{p}'_j = \boldsymbol{f}(\boldsymbol{p}_j) \tag{42}$$

are true so that the mapping between I_1 and I_1^{ud} are one-to-one mapping through I_1^u. These assumptions do not hold exactly about digital image because a pixel exists only on a grid. But it is reasonable if an image has an enough resolution.

Using the assumptions mentioned above and Eqs. (16), (18), we can rewrite the residuals as follows;

$$\begin{aligned} r_i^u(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) &= I_1(\boldsymbol{p}_i) - I_2^f(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i; \boldsymbol{\theta}^u); \boldsymbol{\theta}^d) \\ &= I_1^u(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i; \boldsymbol{\theta}^u); \boldsymbol{\theta}^u) \\ &\quad - I_2^f(\boldsymbol{p}_i + \boldsymbol{u}(\boldsymbol{p}_i; \boldsymbol{\theta}^u); \boldsymbol{\theta}^d) \\ &= I_1^u(\boldsymbol{p}'_i; \boldsymbol{\theta}^u) - I_2^f(\boldsymbol{p}'_i; \boldsymbol{\theta}^d) \end{aligned} \tag{43}$$

$$\begin{aligned} r_j^d(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) &= I_2(\boldsymbol{p}_j) - I_1^u(\boldsymbol{f}(\boldsymbol{p}_j; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u) \\ &= I_2^f(\boldsymbol{f}(\boldsymbol{p}_j; \boldsymbol{\theta}^d); \boldsymbol{\theta}^d) - I_1^u(\boldsymbol{f}(\boldsymbol{p}_j; \boldsymbol{\theta}^d); \boldsymbol{\theta}^u) \\ &= I_2^f(\boldsymbol{p}'_j; \boldsymbol{\theta}^d) - I_1^u(\boldsymbol{p}'_j; \boldsymbol{\theta}^u) \end{aligned} \tag{44}$$

Then L^u, L^d are rewritten as follows

$$L^u(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = \sum_i \rho(I_1^u(\boldsymbol{p}'_i; \boldsymbol{\theta}^u) - I_2^f(\boldsymbol{p}'_i; \boldsymbol{\theta}^d)) \tag{45}$$

$$L^d(\boldsymbol{\theta}^u, \boldsymbol{\theta}^d) = \sum_j \rho(I_2^f(\boldsymbol{p}'_j; \boldsymbol{\theta}^d) - I_1^u(\boldsymbol{p}'_j; \boldsymbol{\theta}^u)) \tag{46}$$

Since the order of points i, j doesn't matter and ρ is an even function, L^u and L^d are identical.

Therefore, the inequality Eq. (39) is derived from the first and second conditions in Table 3, and Eq. (40) is derived from the second and third conditions in Table 3. This means that both of the residuals L^u and L^d decrease at every estimation step and the iteration finally converges.

8.2 Local Minimum

In general, a result of the convergence of nonlinear optimization depends on its initial values, and sometimes the estimates converge into local minimum even though the value of cost function decreases every step as described above. Throughout the discussions we assume implicitly that the result of the first step is good enough for initial state of the second step. In fact, this is reasonable when the distortion is small, however, it leads us to local minimum.

Local minimum which occurs for barrel distortion is illustrated in Fig. 8. The figure shows how a rectangle is distorted. In Fig. 8 (a), a rectangle with no distortion is drawn with dashed line, and its distorted shape observed under barrel distortion is drawn with solid line. As you can see in the figure, and also in the model of distortion defined by Eq. (14), the size of the rectangle should be larger than that of distorted shape for barrel distortion, and the rectangular shrinks toward to the barrel shape along the arrows. Therefore, in this case, I_1^u is expected to be estimated as the dashed rectangular shape.

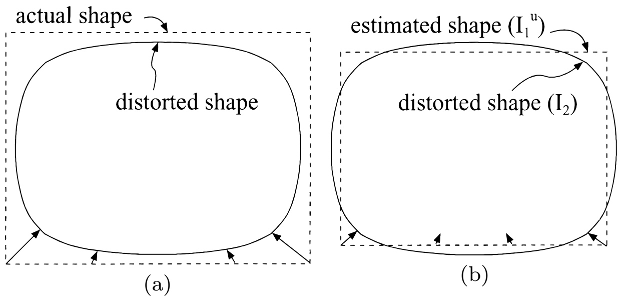


Fig. 8 Difference between estimated and actual shapes under barrel distortion.

However, I_1^u estimated in the first step is different from what it should be. As shown in Fig. 8 (b), in the first step the parameters $\boldsymbol{u}$ are estimated so that I_1^u (rectangular with dashed line) becomes similar to I_2 (barrel shape with solid line) and also the sizes become the same. When the size of I_1^u is similar to that of I_2 like Fig. 8 (b), the rectangular cannot shrink toward to the distorted shape. Then the size of I_2^f is always larger than that of I_1^u, and I_2^f will never be identical to I_1^u through the second step.

Local minimum is a big problem. The idea to avoid the size problem of this case is adding one more parameter, called κ_0, to the intrinsic parameters so that the size is also estimated; that is, the polynomial parts of the first term in Eq. (14), $1 + \kappa_1 R'^2 + \kappa_2 R'^4$, is replaced with $\kappa_0 + \kappa_1 R'^2 + \kappa_2 R'^4$. This parameter can deal with the size problem, but may affect the estimates of other parameters.

9. Conclusions

We have proposed a new technique of automated camera calibration method to obtain internal camera parameters in order to compensate the distortion of image. The proposed method is based on image registration and consists of two nonlinear optimization steps; forward registration step for projection and backward registration for distortion. Experimental results demonstrated the efficiency of the proposed method in comparison with the conventional calibration method. The convergence of the proposed method was shown under the condition that the image has enough resolution, and the problem of local minimum was discussed. We have not mentioned about the one step method [20], the extension of the proposed two-step method, which can be a way to avoid the local minimum. There are some methods that realize the one step method, and we are trying to develop them and compare features of them.

Since the gradation of illumination of the image affects the result of the estimates, the effect should be incorporated into the registration. So far the results of the compensation is evaluated qualitatively because the actual intrinsic camera parameters is unknown. The nonlinear optimization somewhat takes time, but it is enough to run as a batch process. Dealing with the change of illumination of image, avoiding local minimum and a quantitative evaluation of the estimates are the future plans.

References

[1] T. Tamaki, T. Yamamura, and N. Ohnishi, "An automatic camera calibration method with image registration technique," IIIS Proc. SCI2000, vol.5, pp.317–321, 2000.

[2] T. Mukai and N. Ohnishi, "The recovery of object shape and camera motion using a sensing system with a video camera and a gyro sensor," Proc. ICCV'99, pp.411–417, 1999.

[3] J.B. Shim, T. Mukai, and N. Ohnishi, "Improving the accuracy of 3D shape by fusing shapes obtained from optical flow," Proc. CISST'99, pp.196–202, 1999.

[4] R.Y. Tsai, "An efficient and accurate camera calibration tecnique for 3D machine vision," Proc. CVPR'86, pp.364–374, 1986.

[5] R. Willson, "Camera calibration using Tsai's method," ftp://ftp.vislist.com/SHAREWARE/CODE/CALIBRATION/Tsai-method-v3.0b3/ 1995.

[6] C. Matsunaga and K. Kanatani, "Optimal grid pattern for automated matching using cross ratio," Proc. MVA2000, pp.561–564, 2000.

[7] Y. Shibuya and I. Kumazawa, "Estimation of camera parameters and compensation of image distortion matching a priori known shape," IEICE Trans., vol.J83-D-II, no.6, pp.1460–1468, 2000.

[8] H.-Y. Shum and R. Szeliski, "Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment," IJCV, vol.36, no.2, pp.101–130, 2000.

[9] H.S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," T-PAMI, vol.18, no.8, pp.814–830, 1996.

[10] R. Szeliski, "Video mosaics for virtual environment," IEEE Computer Graphics and Applications, vol.16, no.3, pp.22–30, 1996.

[11] G.A.F. Seber and C. J. Wild, Nonlinear Regression, Wiley, New York, 1989.

[12] R. Klette, K. Schlüns, and A. Koschan, Computer Vision Three-Dimensional Data from Images, Springer-Verlag, Singapore, 1998.

[13] R.K. Lenz and R.Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-d machine," T-PAMI, vol.10, no.5, pp.713–720, 1988.

[14] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," T-PAMI, vol.14, no.10, pp.965–980, 1992.

[15] J. Heikkilä and O. Silvén, "A four-step camera calibration procedure with implicit image correction," Proc. CVPR'97, pp.1106–1112, 1997.

[16] R. Jain, R. Kasturi, and B.G. Schunck, Machine Vision, McGraw-Hill, New York, 1995.

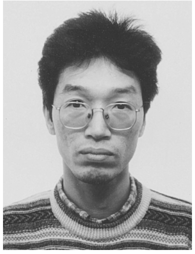
[17] K.R. Castleman, Digital Image Processing, Prentice-Hall, Inc., 1996.

[18] G.R. Bradski and V. Pisarevsky, "Intel's computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture and object recognition," Proc. CVPR2000, vol.2, pp.796–797, 2000. http://www.intel.com/research/mrl/research/cvlib/.

[19] M.J. Black, D.J. Fleet, and Y. Yacoob, "Robustly estimating changes in image appearance," CVIU, vol.78, no.1, pp.8–31, 2000.

[20] T. Tamaki, T. Yamamura, and N. Ohnishi, "Image distortion correction with nonlinear optimization," Proc. ITE Winter Annual Convention 2000, p.58, 2000.

Toru Tamaki received his B.E. and M.S. degrees in information engineering from Nagoya University, in 1996 and 1998, respectively. Currently, he is a Ph.D candidate at the Department of Information Engineering, Graduate School of Engineering, Nagoya University, and a junior research associate at the Bio-Mimetic Control Research Center, RIKEN. His research interests include computer vision and image recognition. He is a student member of ITEJ.



Tsuyoshi Yamamura received his B.E., M.E. and Ph.D degrees from Nagoya University in 1987, 1989 and 1994, respectively. In 1992, he was a Research Associate in the Department of Electronic Information Engineering at Nagoya University. In 1995, he was an Assistant Professor in the Department of Information Engineering, Graduate School of Engineering, Nagoya University. Currently he is an Associate Professor in the Faculty of Information Science and Technology, Aichi Prefectural University. He is engaged in research on Natural Language Processing and Visual Information Processing. He is a member of IEEE and IPSJ.

Noboru Ohnishi received his B.E., M.S. and Ph.D degrees in electrical engineering from Nagoya university, in 1973, 1975 and 1984, respectively. In 1975, he was a Researcher of the Rehabilitation Engineering Center under the Ministry of Labor. In 1986, he was an Assistant Professor, Department of Electrical Engineering, Nagoya University. In 1989, he was an Associate Professor, Department of Electrical Engineering, Nagoya University. In 1994, he was a Professor, Department of information Engineering, Nagoya University. Currently he was a Professor in the Center for Information Media Studies, Nagoya University, and concurrently a team leader of laboratory for bio-mimetic sensory system at bio-mimetic control research center of RIKEN. His research interests include computer-vision and -audition, robotics, bio-cybernetics, and rehabilitation engineering. He is a member of IEEE, RSJ, SICE, IEEJ, ITE and IPSJ.