International Journal of Image and Graphics Vol. 0, No. 0 (2002) 000–000

CORRECTING DISTORTION OF IMAGE BY IMAGE REGISTRATION WITH IMPLICIT FUNCTION THEOREM

TORU TAMAKI
Faculty of Engineering, Niigata University
8050 Ikarashininocho, Niigata-shi, Niigata 950-2181, Japan
E-mail: tamaki@ie.niigata-u.ac.jp

TSUYOSHI YAMAMURA
Faculty of Information Science and Technology, Aichi Prefectural University
1522-3 Ibaragabasama, Kumabari, Nagakute-cho Aichi-gun, Aichi 480-1198 Japan
E-mail: yamamura@ist.aichi-pu.ac.jp

NOBORU OHNISHI *
Center for Information Media Studies, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464–8603 Japan
E-mail: ohnishi@ohnishi.nuie.nagoya-u.ac.jp

Received (30 Nov. 2001)
Revised (revised date)

We propose a method for correcting image distortion due to camera lenses by calibrating intrinsic camera parameters. The proposed method is based on image registration and doesn't require point-to-point correspondence. Parameters of three successive transformations — view change, radial distortion and illumination change — are estimated using the Gauss-Newton method. Estimating all 19 unknowns simultaneously, we introduce the implicit function theorem for calculating the Jacobian. To avoid local minima, we first estimate parameters for view change and employ coarse-to-fine minimization. Experimental results using real images demonstrate the robustness and the usefulness of the proposed method.

Keywords: distortion correction, intrinsic camera parameters, camera calibration, nonlinear optimization

1. Introduction

1.1. *Background*

Calibrating a camera and correcting image distortion are important processes for computer vision. Many approaches on calibrating extrinsic camera parameters or

*He was with the laboratory of bio-mimetic sensory systems, Bio-Mimetic Control Research Center, RIKEN.

recovering 3D structure (for examples, see [1,2,3,4]) formulate the problems without considering distortion because of simplicity. However, distortion is inevitable when we use an ordinary lens installed on an inexpensive camera or a zoom lens; sometimes a point may be displaced more than ten pixels around the corner of the image due to distortion. Although self-calibration with a fundamental matrix[5] has been studied well recently, such studies don't take into account barrel or pin cushion distortion. Pre-calibration of intrinsic camera parameters and correction of distorted image are thus required preprocesses for such research and to produce quality images.

For researchers' convenience, some computer codes to calibrate intrinsic parameters have been made available through the internet (e.g., Tsai's method[6,7] implementation,[8] or Intel CV library[9]). However, such ordinary techniques[10,11,12,13] require a number of correspondences between a point in the image and a feature point at the known three-dimensional coordinates (on a plane or on some structure like a cube or a house) to estimate the parameters of the transformation of the corresponding points.

When the correspondences are established manually, errors can be generated by human operation and this compromises reliability. Moreover, it takes much time and patience. For example, it is too hard to measure distortion parameters repeatedly as changes in camera zooming.

An alternative procedure is to detect markers such as corners,[10,11] circles,[13] curves[14] or intersections. This can be performed by edge detection or template matching techniques that can be performed on a subpixel accuracy. However, another correspondence problem arises: When there are many feature points in a space, how do we decide which should correspond to a point on an image? The problem increases as the number of the markers increases in order to improve the estimation accuracy. Even if the problem can be avoided,[15] the number of points for the correspondence is still limited.

In this paper, we propose a new method for correcting image distortion due to camera lenses by calibrating intrinsic camera parameters. The proposed method establishes the correspondence between a calibration pattern (the ideal image) and a distorted picture of the pattern taken by a camera. The correspondence is based on an image registration that is often used for motion analysis. The estimation is expected to be more precise than approaches involving marker detection because this method uses all points of the image rather than just using relatively few marker points. The proposed method estimates the parameters of the transformations of a plane under perspective projection, radial distortion and spatial linear illumination change by a nonlinear optimization technique that minimizes the residuals between two images.

1.2. *Image registration for distortion*

The basic idea is that correspondence between points is necessary for calibra-

tion: registration can satisfy this requirement. The proposed method establishes the 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 three transformations (see Fig.1); view change $\boldsymbol{u}$, distortion $\boldsymbol{d}$, and illumination variation H. I_2 is regarded as an image generated from I_1 by applying the three functions. Using the image registration technique, the proposed method estimates the parameters of the functions by minimizing the difference between I_1 and I_2, that is, the sum of the squares of the intensity residuals between the two images.

The procedure of the proposed method is as follows. First, prepare calibration pattern I_1. Any digital image (taken by a digital camera, scanned photo or CG) can be used as the pattern. Second, print the pattern on a sheet of paper using a printer (we assume that the printer makes an ideal print). Third, use the camera to be calibrated to acquire an image I_2 of the printed pattern I_1. Finally, register the pattern I_1 and the observed image I_2 to determine the parameters of the functions $\boldsymbol{u}$, $\boldsymbol{d}$ and H.

Some researchers use the image registration to calibrate extrinsic camera parameters[1] or mosaicing.[2,16] The problems with employing the image registration technique in a straightforward manner are that a nonlinear function $\boldsymbol{f}$ (the inverse of $\boldsymbol{d}$) is often used to model the distortion, and that the function $\boldsymbol{d}$ is not expressed in a closed-form but is rather implemented by an iterative procedure. We require that the gradient of $\boldsymbol{d}$ is known to be used by gradient-based optimization, however, it is difficult to obtain the gradient because $\boldsymbol{d}$ is a procedure. Although this problem occurs when the distortion parameters are estimated, there is no account taken of it in fish-eye lens mosaicing.[16] We introduce the implicit function theorem[17,18] to obtain the Jacobian of $\boldsymbol{d}$, not $\boldsymbol{d}$ itself. This enables us to see the registration-based intrinsic/extrinsic parameter calibration as a unified approach including lens distortion and illumination variation.

Some registration-based methods have been developed,[16,19] but they require rotating a camera around a projection center for taking two (or more) pictures. In contrast, our method needs only one picture from any viewpoint.

In section 2, we explain models of image transformation including view change, distortion and illumination variation. We then describe the algorithm of registration based on a nonlinear optimization in section 3. Finally, we present the experimental results in section 4.

2. Models of Transformations

In this section, we describe the model of transformation between the calibration pattern I_1 and the observed image I_2. The transformation comprises three subsequent operations. The first is a change of view from pattern I_1 to a printed sheet in a 3D space, the second is the displacement from the projection of the sheet due to distortion, and the last is the illumination change that alters the intensity of the pattern.

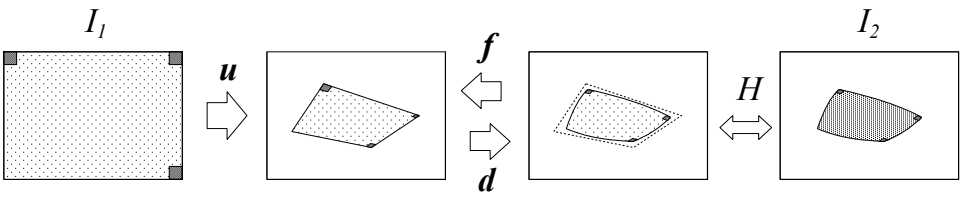


Figure 1: Observation I_2 of the calibration pattern I_1 modeled by three transformations.

2.1. *Modeling view change*

Given two images of the same planar object from different viewpoints, the relationship between them is described by a planar perspective motion model with eight parameters.[2,20] As shown in Fig.2, I_1 and I_2 can be considered as the different views of the same plane because of the following reason. Since I_1 is just a digital image, I_1 is a plane exactly identical to the image plane. The printed sheet is regarded as a plane transformed from the plane of I_1, and I_2 is the projection of the sheet onto the image plane having a slight displacement due to the distortion.

The model warps a point $\boldsymbol{p} = (x, y)^T$ on I_1 into the corresponding point on I_2, $\boldsymbol{p}^u = (x^u, y^u)^T$, using the function $\boldsymbol{u}$ of $\boldsymbol{\theta}^u = (\theta^u_1, \ldots, \theta^u_8)^T$ as follows.[2]

$$\boldsymbol{p}^u = \boldsymbol{u}(\boldsymbol{p}, \boldsymbol{\theta}^u) = \frac{1}{\theta^u_1 x + \theta^u_2 y + 1} \begin{pmatrix} \theta^u_3 x + \theta^u_4 y + \theta^u_5 \\ \theta^u_6 x + \theta^u_7 y + \theta^u_8 \end{pmatrix} \tag{1}$$

The Jacobian of $\boldsymbol{u}$ is given[2] by

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}^u} = \begin{pmatrix} -x^2 & -xy & x & y & 1 & 0 & 0 & 0 \\ -xy & -y^2 & 0 & 0 & 0 & x & y & 1 \end{pmatrix} \tag{2}$$

2.2. *Modeling distortion*

The relationships between undistorted and distorted coordinates in an image (shown in Fig.3) are often modeled by five intrinsic camera parameters[21,6]: the radial distortion parameters κ_1 and κ_2, the coordinates of image center $(c_x, c_y)^T$, and the horizontal scale factor s_x. We write these parameters as $\boldsymbol{\theta}^d = (\theta^d_1, \ldots, \theta^d_5) = (\kappa_1, \kappa_2, c_x, c_y, s_x)^T$. Although we consider only the radial distortion, the following discussion can also be applied when another model involving decentering distortion[22,23] is employed.

Distortion is represented with respect to the image center $(c_x, c_y)^T$. Let $\boldsymbol{p}^u = (x^u, y^u)^T$ be a point in I_2 without considering distortion, that is, $\boldsymbol{p}^u = \boldsymbol{u}(\boldsymbol{p})$; $\boldsymbol{p}^u$ is moved to $\boldsymbol{p}^d = (x^d, y^d)^T$ by the radial distortion. Here we have two functions between $\boldsymbol{p}^u$ and $\boldsymbol{p}^d$.

$$\boldsymbol{p}^d = \boldsymbol{d}(\boldsymbol{p}^u, \boldsymbol{\theta}^d) \tag{3}$$

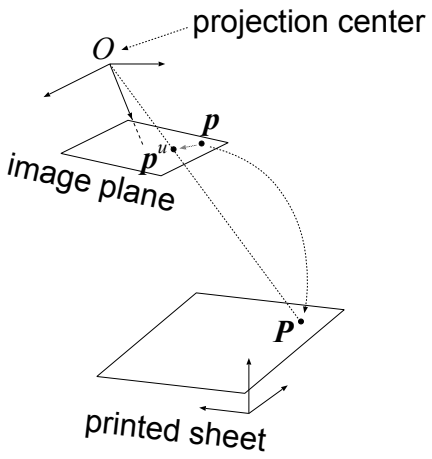


Figure 2: Relationship between I_1 and I_2.

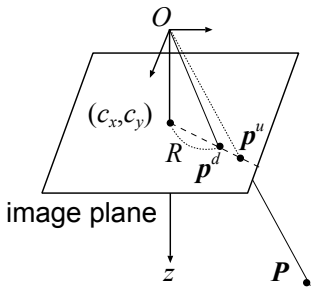


Figure 3: Distortion model

$$\boldsymbol{p}^u \quad = \quad \boldsymbol{f}(\boldsymbol{p}^d, \boldsymbol{\theta}^d) = \begin{pmatrix} \frac{x^d - c_x}{s_x}(1+\kappa_1 R^2 + \kappa_2 R^4) + c_x \\ (y^d - c_y)(1+\kappa_1 R^2 + \kappa_2 R^4) + c_y \end{pmatrix} \tag{4}$$

where $R = \sqrt{\left((x^d - c_x)/s_x\right)^2 + (y^d - c_y)^2}$. $\boldsymbol{f}$ and $\boldsymbol{d}$ are the inverse of each other, and $\boldsymbol{d}$ is not a closed-form function of $\boldsymbol{p}^u$ but is implemented by an iterative procedure[21] (see Appendix A).

In addition to the Jacobian of $\boldsymbol{u}$, the Jacobian of $\boldsymbol{d}$ is also needed for a gradient method. Here we introduce the *implicit function theorem*[17] for systems[18]. This theorem can represent the Jacobian of $\boldsymbol{d}$ as an explicit form through $\boldsymbol{f}$. Let $\boldsymbol{F}$ be a function of $\boldsymbol{q} = (\boldsymbol{p}^u, \boldsymbol{\theta}^d)$ and $\boldsymbol{p}^d$ represented by

$$\boldsymbol{F}(\boldsymbol{q}, \boldsymbol{p}^d) = \boldsymbol{p}^u - \boldsymbol{f}(\boldsymbol{p}^d, \boldsymbol{\theta}^d) \tag{5}$$

If $\boldsymbol{F}(\boldsymbol{q}, \boldsymbol{d}(\boldsymbol{q})) = 0$ is satisfied for $\forall \boldsymbol{q}$, then $\boldsymbol{p}^d = \boldsymbol{d}(\boldsymbol{q})$ is called an implicit function determined by $\boldsymbol{F}(\boldsymbol{q}, \boldsymbol{p}^d) = 0$. In our case, the condition is theoretically always satisfied because we defined $\boldsymbol{d}$ as the inverse of $\boldsymbol{f}$, and numerically Eq.(5) is almost 0 (it can be less than 10^{-10}).

According to the theorem, the Jacobian is given by the following equations.

$$\begin{aligned} \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{q}} &= -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}} = -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \begin{pmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^u} & \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}^d} \end{pmatrix} \\ &= -\begin{pmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^u} & \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}^d} \end{pmatrix} \end{aligned} \tag{6}$$

unless $\dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}$ is singular. On the other hand, the Jacobian can also be decomposed

into two parts as follows.

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{q}} = \left(\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{p}^u} \quad \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\theta}^d} \right) \tag{7}$$

Therefore, the second part is the desired gradient of $\boldsymbol{d}$.

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\theta}^d} = -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}^d} = -\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}^d} \tag{8}$$

The first part is the differential of $\boldsymbol{d}$ with respect to $\boldsymbol{p}^u = \boldsymbol{u}(\boldsymbol{p})$ (this is also used in the later formulation).

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{p}^u} = -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}^u} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}^d}^{-1} \tag{9}$$

According to Eq.(4), the elements of the Jacobian are as follows.[24]

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}^d} = \begin{pmatrix} R^2 \dfrac{x_d - c_x}{s_x} & R^2(y_d - c_y) \\ R^4 \dfrac{x_d - c_x}{s_x} & R^4(y_d - c_y) \\ \dfrac{\partial x_u}{\partial c_x} & \dfrac{\partial y_u}{\partial c_x} \\ \dfrac{\partial x_u}{\partial c_y} & \dfrac{\partial y_u}{\partial c_y} \\ \dfrac{\partial x_u}{\partial s_x} & \dfrac{\partial y_u}{\partial s_x} \end{pmatrix}, \qquad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}^d} = \begin{pmatrix} 1 - \dfrac{\partial x_u}{\partial c_x} & -\dfrac{\partial y_u}{\partial c_x} \\ -\dfrac{\partial x_u}{\partial c_y} & 1 - \dfrac{\partial y_u}{\partial c_y} \end{pmatrix} \tag{10}$$

where

$$\frac{\partial x_u}{\partial c_x} = 1 - \frac{1}{s_x}(1 + \kappa_1 R^2 + \kappa_2 R^4) - 2(\kappa_1 + 2\kappa_2 R^2)\frac{(x_d - c_x)^2}{{s_x}^3} \tag{11}$$

$$\frac{\partial y_u}{\partial c_x} = -2(\kappa_1 + 2\kappa_2 R^2)\frac{x_d - c_x}{{s_x}^2}(y_d - c_y) \tag{12}$$

$$\frac{\partial x_u}{\partial c_y} = -2(\kappa_1 + 2\kappa_2 R^2)\frac{x_d - c_x}{s_x}(y_d - c_y) \tag{13}$$

$$\frac{\partial y_u}{\partial c_y} = 1 - (1 + \kappa_1 R^2 + \kappa_2 R^4) - 2(y_d - c_y)^2(\kappa_1 + 2\kappa_2 R^2) \tag{14}$$

$$\frac{\partial x_u}{\partial s_x} = \frac{-(x_d - c_x)}{{s_x}^2}(1 + \kappa_1 R^2 + \kappa_2 R^4) - 2(\kappa_1 + 2\kappa_2 R^2)\frac{(x_d - c_x)^3}{{s_x}^4} \tag{15}$$

$$\frac{\partial y_u}{\partial s_x} = -2(y_d - c_y)(\kappa_1 + 2\kappa_2 R^2)\frac{(x_d - c_x)^2}{{s_x}^3} \tag{16}$$

2.3. *Modeling illumination variation*

A point $\boldsymbol{p}$ in I_1 is transformed into the point $\boldsymbol{p}^d$ in I_2 by view change and distortion. However, the intensities of the corresponding points will not be identical because gray level of the sheet on which I_1 is printed is different from the original

one. The intensity of I_2 also depends on the exposure of the camera, illumination of the environment, and so on.

We take the following function as a model of intensity change:

$$H(I(\boldsymbol{p}), \boldsymbol{p}, \boldsymbol{\theta}^h) = (\theta_1^h + \theta_2^h x + \theta_3^h y)I(\boldsymbol{p}) + (\theta_4^h + \theta_5^h x + \theta_6^h y) \tag{17}$$

where $\boldsymbol{\theta}^h = (\theta_1^h, \ldots, \theta_6^h)^T$. This model represents both the gain and bias as spatial linear functions. This is not an exact model, but it can deal with simple changes in illumination.[25] There has been an attempt to eliminate the variations in illumination,[26] although it is not applicable to our case because it finds basis images from a set of images containing at least three different planes in a scene.

3. Minimization with Some Arrangements

In this section, we describe how to estimate the parameters of the functions $\boldsymbol{u}$, $\boldsymbol{d}$ and H. Image registration seeks to minimize the residuals r_i of intensities of the two images, I_1 and I_2.

$$r_i = I_1(\boldsymbol{p}_i) - H(I_2(\boldsymbol{p}_i^d), \boldsymbol{p}_i^d, \boldsymbol{\theta}^h) \tag{18}$$

$$\boldsymbol{p}_i^d = \boldsymbol{d}(\boldsymbol{p}_i^u, \boldsymbol{\theta}^d) \tag{19}$$

$$\boldsymbol{p}_i^u = \boldsymbol{u}(\boldsymbol{p}_i, \boldsymbol{\theta}^u) \tag{20}$$

The function to be totally minimized is the sum of squares of the residuals over the image I_1.

$$\min_{\boldsymbol{\theta}} \sum_i {r_i}^2 \ , \quad \boldsymbol{p}_i \in I_1 \tag{21}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{19})^T = (\boldsymbol{\theta}^u, \boldsymbol{\theta}^d, \boldsymbol{\theta}^h)^T$.

Estimating the parameters $\boldsymbol{\theta}$, the objective function is minimized by the Gauss-Newton method, a nonlinear optimization technique.[27] The parameters are updated with some initial value by the following rule.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \ \delta\boldsymbol{\theta} \tag{22}$$

The decent direction $\delta\boldsymbol{\theta} = (\delta\theta_1, \ldots, \delta\theta_{19})^T$ is calculated as follows[27]:

$$\delta\boldsymbol{\theta} = -(J^T J)^{-1} J^T \boldsymbol{r} \tag{23}$$

$$J = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\theta}} \tag{24}$$

where $\boldsymbol{r} = (r_1, r_2, \ldots)^T$. This is the same as the least square formulation, that is, the system of linear equations[28] written as

$$\sum_i \sum_l \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \delta\theta_l = -\sum_i r_i \frac{\partial r_i}{\partial \theta_k} \tag{25}$$

for $k = 1, \ldots, 19$ (the number of parameters). The partial derivatives are the elements of the following Jacobian obtained by the chain rule of vector differentiation.[27]

$$\frac{\partial r}{\partial \boldsymbol{\theta}} = \left(\frac{\partial r}{\partial \boldsymbol{\theta}^u} \quad \frac{\partial r}{\partial \boldsymbol{\theta}^d} \quad \frac{\partial r}{\partial \boldsymbol{\theta}^h} \right) \tag{26}$$

$$\begin{aligned} \frac{\partial r}{\partial \boldsymbol{\theta}^u} &= \frac{\partial r}{\partial H} \left(\frac{\partial H}{\partial I_2} \frac{\partial I_2}{\partial \boldsymbol{p}^d} + \frac{\partial H}{\partial \boldsymbol{p}^d} \right) \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{p}^u} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}^u} \\ &= -\left(\frac{\partial H}{\partial I_2} \nabla I_2(\boldsymbol{p}^d) + \frac{\partial H}{\partial \boldsymbol{p}^d} \right) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\theta}^u} \end{aligned} \tag{27}$$

$$\begin{aligned} \frac{\partial r}{\partial \boldsymbol{\theta}^d} &= \frac{\partial r}{\partial H} \left(\frac{\partial H}{\partial I_2} \frac{\partial I_2}{\partial \boldsymbol{p}^d} + \frac{\partial H}{\partial \boldsymbol{p}^d} \right) \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\theta}^d} \\ &= \left(\frac{\partial H}{\partial I_2} \nabla I_2(\boldsymbol{p}^d) + \frac{\partial H}{\partial \boldsymbol{p}^d} \right) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}^d}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}^d} \end{aligned} \tag{28}$$

$$\frac{\partial r}{\partial \boldsymbol{\theta}^h} = -(I_2(\boldsymbol{p}^d) \quad x^d I_2(\boldsymbol{p}^d) \quad y^d I_2(\boldsymbol{p}^d) \quad 1 \quad x^d \quad y^d) \tag{29}$$

Once the direction is decided by solving the system of equations in Eq.(25), the step length α is optimized by line minimization.[29] Update by Eq.(22) is repeated until it converges. At each iteration, the parameters estimated in the previous iteration are used for the current Jacobian.

3.1. *Interpolating pixel value*

When we want to obtain the intensity of a pixel whose coordinate is not on the integer grid (as frequently occurs), we need to interpolate the intensity using the values of the pixels that are already located on the grid. For this purpose, we use the bilinear interpolation[30] , a simple and fast method, which interpolates using the values of four neighboring pixels on a rectilinear grid.

3.2. *Initial state*

At the beginning of the iteration, we use the following initial value for each of the parameters:

$$\boldsymbol{\theta}^u \leftarrow (0, 0, 1, 0, 0, 0, 1, 0)^T \tag{30}$$

$$\boldsymbol{\theta}^d \leftarrow (\kappa_1^0, 0, \frac{w}{2}, \frac{h}{2}, 1)^T \tag{31}$$

$$\boldsymbol{\theta}^h \leftarrow (1, 0, 0, 0, 0, 0)^T \tag{32}$$

where w and h represent the width and height of I_2, and κ_1^0 is the initial value randomly selected for κ_1 to prevent the differentials of κ_1 and κ_2 (Eqs.(11)$\sim$(14)) from always being 0 by initializing $\kappa_1 = \kappa_2 = 0$. We empirically choose $\kappa_1^0 \in [-10^{-7}, 10^{-7}]$.

3.3. *Partial optimization*

If there is a large error in estimates at the beginning of the minimization, the estimation is unstable because the number of parameters is relatively large, and $\boldsymbol{\theta}^h$ changes the intensity of the observed image.

Even though, the initial values of the estimates are always set to the values mentioned above and the minimization sometimes falls into a local minimum. To avoid the instability in the early stage of estimation, we perform the optimization for only some of the parameters, not all of them; only $\boldsymbol{\theta}^u$ is estimated, while $\boldsymbol{\theta}^d$ and $\boldsymbol{\theta}^h$ are unchanged. This is called search space decomposition.[31] There is no guarantee of convergence, but it reduces the dimension of the search space and stabilize the estimation. After the estimation of $\boldsymbol{\theta}^u$ converges, we estimate all of the parameters.

However, a good initialization is needed for a large distortion (see section 4.2).

3.4. *Coarse-to-fine*

To reduce computational time and to perform precise estimation even when there is a relatively large error in the initial state, we employ a strategy for shifting from a coarse resolution to a fine resolution (known as coarse-to-fine strategy.[19]) First, we use blurred images with a large Gaussian kernel then change the filter to a smaller Gaussian kernel and repeat the optimization.

3.5. *Correcting distortion*

After estimating the parameters of the distortion $\boldsymbol{\theta}^d$, we can correct the distortion and obtain the corrected image I_2' by using the following relation.

$$I_2'(\boldsymbol{p}) = I_2(\boldsymbol{d}(\boldsymbol{p}, \boldsymbol{\theta}^d)) \tag{33}$$

Once we obtain the distortion parameters, we can use them for correction as long as the lens zoom is unchanged.

4. Experimental Results

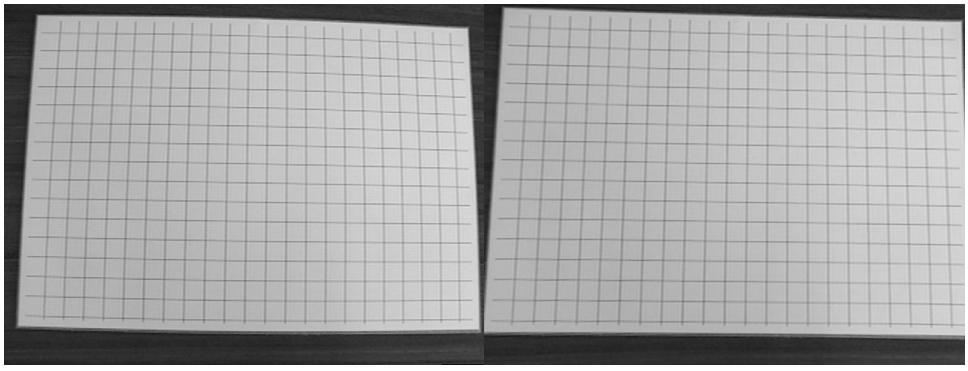
4.1. *Correcting distortion*

We conducted experiments with the proposed method using real images taken by a camera having a zoom lens. We used a scanned photograph as the calibration pattern (shown in Fig.4), printed it with a laser monochrome printer (Apple Laserwriter 16/600 PS), and then captured images of the printed sheet by a CCD video camera (Sony EVI-D30) with a video capture device (IO-DATA GV-VCP2M/PCI) installed in a PC. We placed the printed pattern in front of the camera almost parallel to the image plane. The captured image of the pattern is shown in Fig.5(a). We also took an image of a grid pattern (shown in Fig.5(c)) to help visualize the correction of distortion.

Figure 5(b) and (d) show the images corrected by Eq.(33) with the estimated parameters using Fig.5(a) as I_2. In the corrected image Fig.5(d), the curved lines

Figure 4: Calibration pattern (640×480)

(a) (b)



(c) (d)

Figure 5: Experiment results of the proposed method. (a) Image of the calibration pattern taken by the camera at the widest view angle (640×480). (b) Corrected image of (a). (c) Image of the grid pattern. (d) Corrected image of (c).

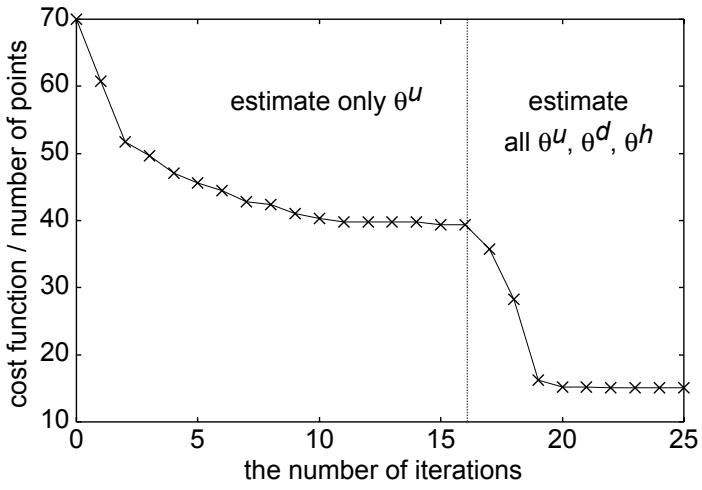


Figure 6: Convergence of the estimation. Horizontal axis is the number of iterations to update the estimates; vertical axis represents the sum of squares divided by the number of points in I_1.

on the grid pattern in the distorted image are corrected to straight lines, so the proposed method works well. The computational time was about 20 minutes on a PC (866MHz CPU, GNU C++ and CLAPACK). However, the optimization had almost converged after fewer than 30 iterations.

We can see the convergence in Fig.6, which shows the sum of the squares of the intensity residuals of the first 25 iterations. As we mentioned in section 3.3, only $\boldsymbol{\theta}^u$ was estimated in the early stage of the iteration while $\boldsymbol{\theta}^d$ and $\boldsymbol{\theta}^h$ were fixed to their initial value. After the estimation of $\boldsymbol{\theta}^u$ had converged (16 iterations), the minimization using all parameters began and converged after several iterations.

To visualize the convergence, we produced a synthetic image which is transformed from I_1 by using $\boldsymbol{u}$, $\boldsymbol{d}$ and H with current estimates at every step. Each image of Fig.7 illustrates the difference between I_2 and the synthesized image. At the first iteration, the two images were quite different and the difference image had many dark pixels. After 25 iterations, the estimation had converged and the subtraction image had few dark pixels, which means that the synthetic image and I_2 became quite similar to each other and the estimation result was good.

4.2. *Distortion parameters while changing zoom*

The advantage of the proposed method is convenience for the human operator. The requirements are just a printed pattern and one captured image of it; a batch process is then called without any manual operations. This simplicity enables us to see the distortion parameter change that arises due to changing the zoom of the camera, while point correspondence-based conventional methods require an enor-

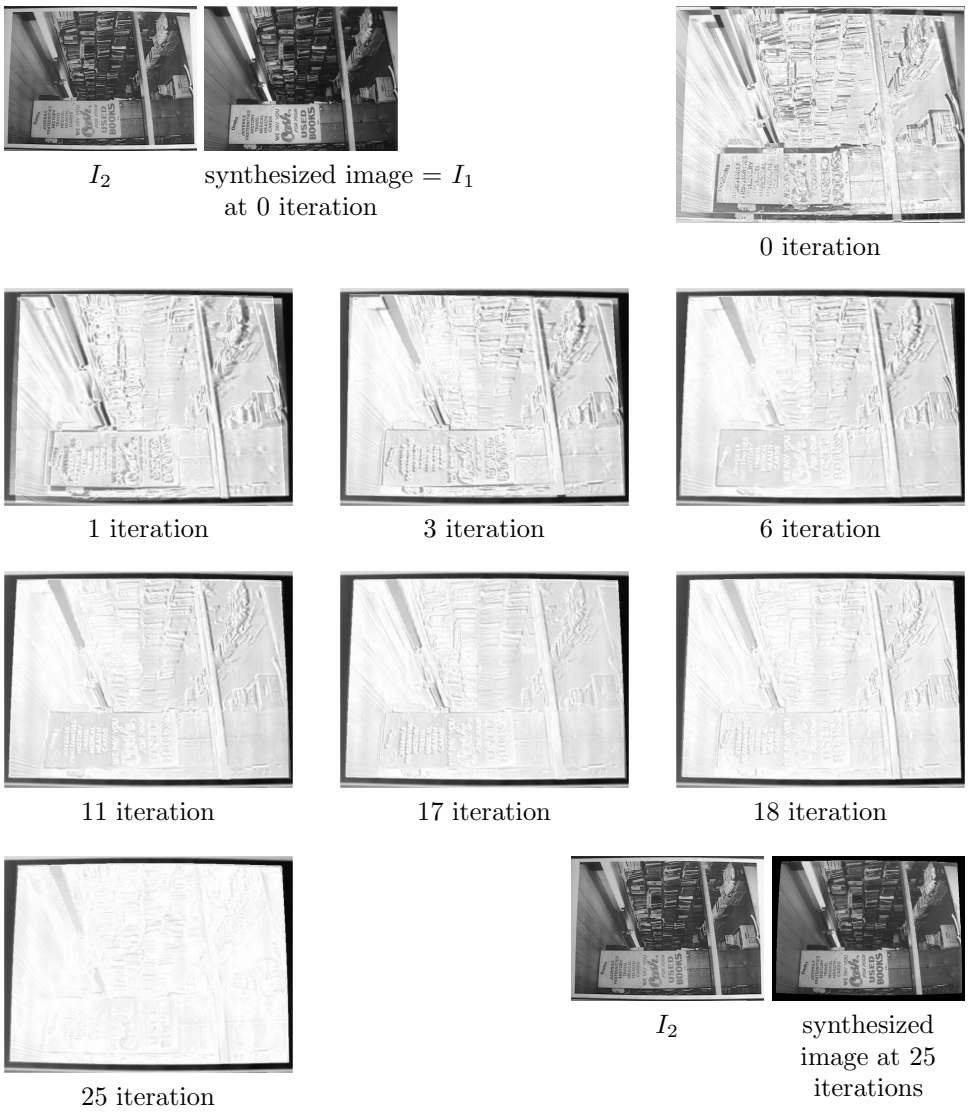
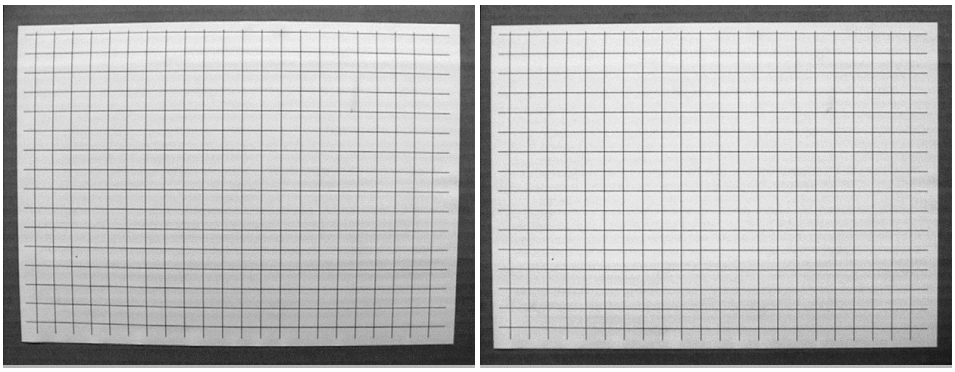


Figure 7: Visualizations of the convergence of the estimation. Each image shows the subtraction image between the observed image I_2 and the synthesized image with the current estimates. The more the subtraction is white, the better the parameters are estimated.



(a) zoom=0 (b) zoom=455

Figure 8: Captured images with different zoom parameters.

mous number of selection of points input by a mouse. The camera that we used in the experiment above has a zoom lens whose zooming can be controlled by receiving a command[32] through a serial communication port, so that we can accurately plot the distortion parameters against zooming.

In general, an image captured by a zoom lens with wide view angle has barrel distortion ($\kappa > 0$). At the tele side of the zoom the distortion is pin cushion ($\kappa < 0$), and there is a range of zoom with no (or very small) distortion between them. As shown in Fig.8(a) and (b), this zoom lens has large barrel distortion at the widest view angle (zoom=0), and very small distortion at the middle view angle (zoom=455). The estimated parameters reflect this observation.

Figures 9(a)∼(e) show the estimated parameters of 44 zoom settings. The horizontal axis represents the zooming between 0 and 455; 0 is the widest view angle (48.8 degrees), and the maximum zoom of this lens is 1023 (4.4 degrees).

In Fig.9(a), we can see that the distortion parameter κ_1 monotonically decreases as the camera zooms in, while in Fig.9(b) κ_2 increases and changes its sign from negative to positive. c_x, c_y and s_x are shown in Figs.9(c)(d) and (e). Apparently, c_x and s_x change and c_y stays, however, these three parameters become less precise when κ_1 (and κ_2) is small because, if there is no distortion ($\kappa_1 = \kappa_2 = 0$), the image center and the scale factor are indefinite. Furthermore, the estimation of s_x becomes inaccurate as the distortion becomes small. The reason is the combined use of the view change and the distortion transformations; the view change stretches the image horizontally while the distortion makes the stretched image shrink with $s_x < 1$. Therefore, the calibration of the scale factor is difficult for small distortion.

To confirm that the estimated parameters are consistent with what we see in Fig.8, we computed the displacement of the corner (640,480) due to the distortion using the estimated parameters[†](see Fig.10(a)). The displacement is defined as the

[†]we forced $s_x = 1$ because of the reason described above.

distance between $\boldsymbol{p}^d$ and $\boldsymbol{p}^u$ using Eq.(3) and Eq.(4), and the negative value means the pin-cushion effect that the point is moved toward to the image center. We can see the contribution of κ_1 to the distortion by letting $\kappa_2 = 0$, and vice versa. Actually, κ_1 is dominant when the zooming is wide, but the effect of κ_2 becomes larger than that of κ_1 when the zooming goes to the tele side. Although the effect of κ_2 is the pin cushion distortion at wide view angle, totally the radial distortion is the barrel distortion for all zooming as shown in Fig.10(b).

Note that the horizontal axis in Fig.9 and Fig.10(a) is not identical to the focal length of the camera but is just a parameter to control zooming of the lens. The zooming parameter is related to the focal length, but it doesn't mean that the parameter is linearly proportional to the focal length.

4.3. *Severe distortion and convergence*

Another result of correcting a distorted image is shown in Fig.11. Figure 11(a) is a grid pattern captured by another camera (Sony DXC-200A) with a wide angle lens (Sony VCL-4V10XEA). Figure 11(b) is the image corrected using the proposed method. For such severe distortion, the proposed method requires an appropriate initial state for the parameters to avoid falling into local minima. Instead of using the values in section 3.2, in this case, we decided the initial state as follows. We chose the coordinates of four corners of the pattern in the captured image and solved Eq.(1) as a system of equations with eight unknowns, then used the solution as the initial state of $\boldsymbol{\theta}^u$. We also set κ_1^0 to 1e-6; this larger positive value represents severe barrel distortion. Partial optimization is also slightly changed. We first fixed $\boldsymbol{\theta}^d$ and $\boldsymbol{\theta}^h$, then fixed $\boldsymbol{\theta}^h$, and finally estimated all parameters.

These devices seem to be ad hoc. Therefore, much more sophisticated global optimization techniques should be used for large distortion. Even for small distortion, the optimization may fall into local minima because there are 19 unknowns. Nevertheless, the proposed method described in section 3 was executed 44 times to plot Fig.9 without any change or any interaction, and worked well throughout the experiment. The reason is that the distortion is relatively small (κ_1 is less than 4e-7) and the pattern occupied a large area in the captured image.

4.4. *Mosaicing using corrected images*

One of the applications of this method is improving the mosaicing of images. Using two distorted images (Fig.12(a) and (b)), mosaicing is imperfect as seen in Fig.13(a) (at the corner of the original image in Fig14(a) and (b) where the displacement of distortion is severe). In contrast, the mosaic shown in Fig.13(b) composed of two images (Fig.12(c) and (d)) corrected by the proposed method is correctly generated. The images were taken by a digital camera (Olympus CAMEDIA C-960ZOOM).

5. Conclusions

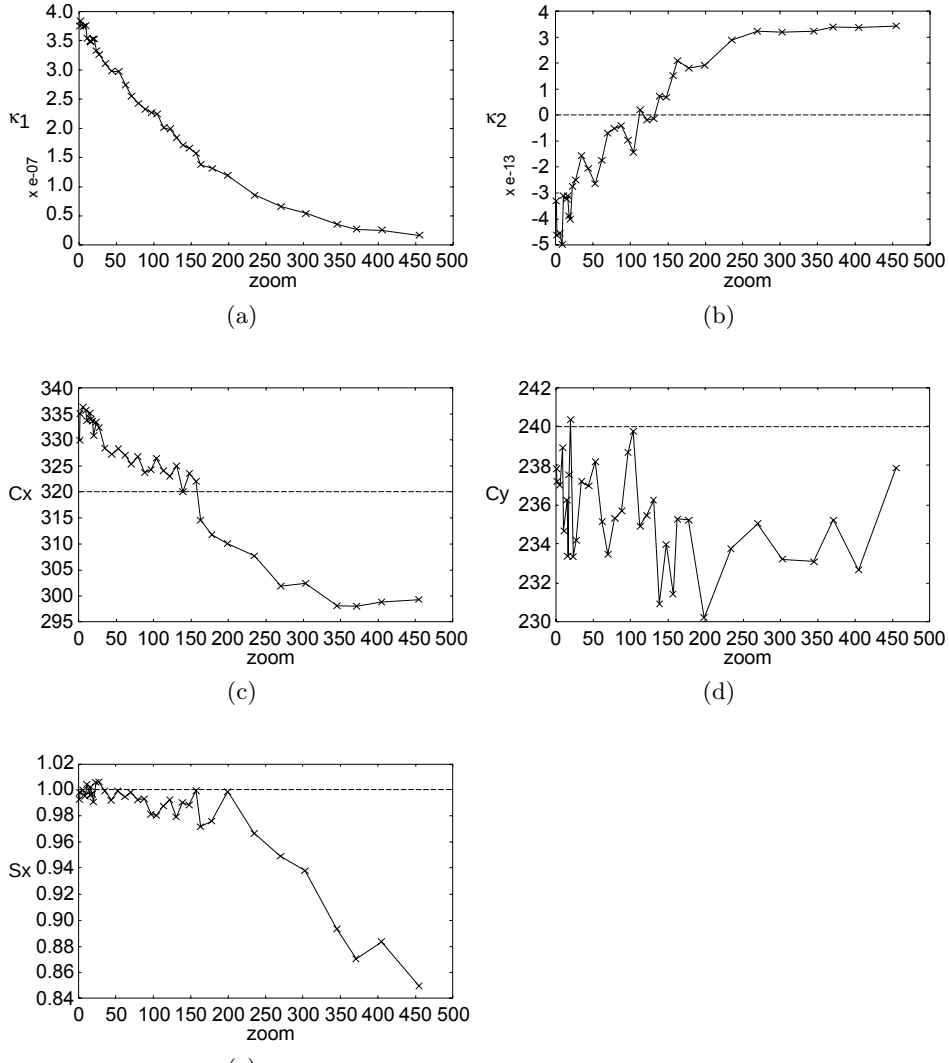
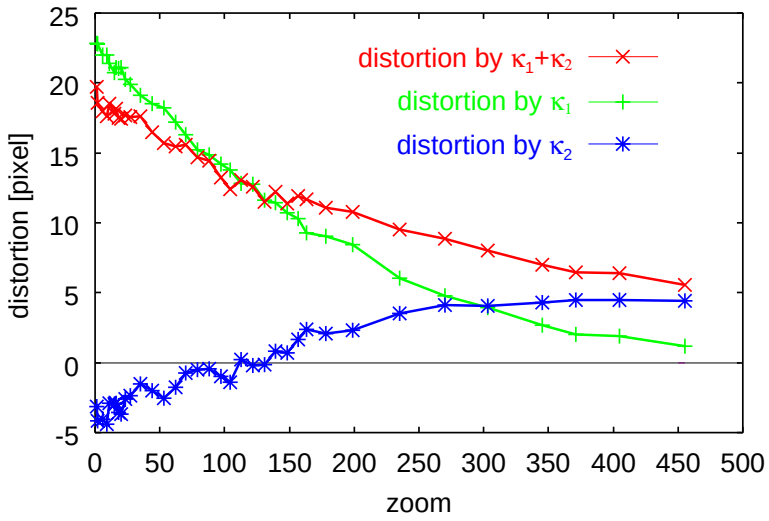
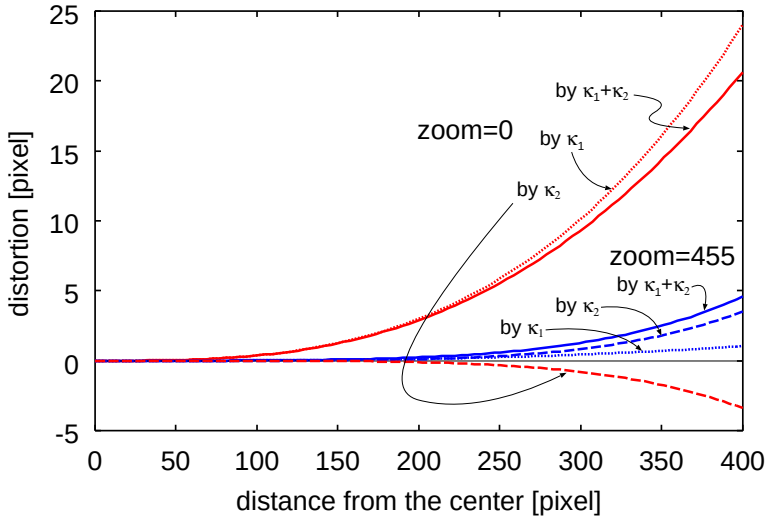


Figure 9: Change of the intrinsic parameters. (a) Distortion parameters κ_1 and (b) κ_2. (c) Image center c_x and (d) c_y. (e) Scale factor s_x. The horizontal axis represents the zooming of the camera; left edge of the graph is the wide side and right is the tele side of zoom.

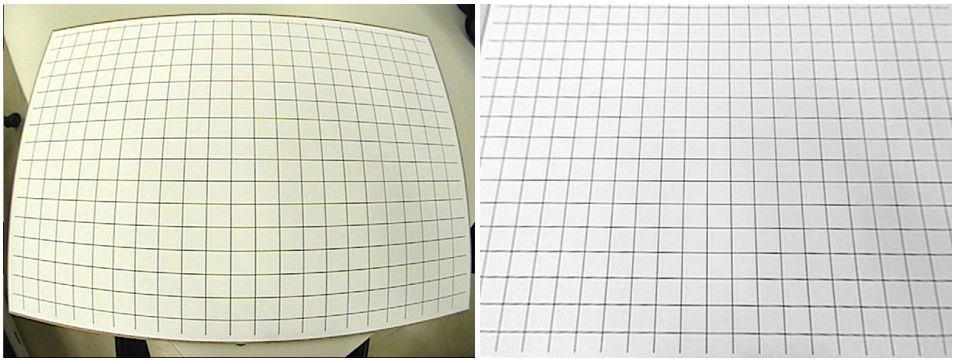


(a)



(b)

Figure 10: Contribution to the distortion by the distortion parameters. (a) Changes of the distortion contributed by κ_1, κ_2 and both κ_1 and κ_2 at the corner (640,480). (b) Magnitude of the distortion with two zoom parameters, and the horizontal axis is the distance from the image center. The distance from the center to the corner (640,480) is 400.



(a) (b)

Figure 11: (a) Severely distorted image (about 85 degrees horizontal and 62 degrees vertical view angles). (b) Corrected image. Estimated parameters : $\boldsymbol{\theta}^d = (1.72\text{e-}6, 1.09\text{e-}11, 305, 227, 1.026)$

We have proposed a new method for correcting image distortion due to camera lenses by calibrating intrinsic camera parameters without any manual operations. The proposed method is based on image registration and consists of a nonlinear optimization to estimate parameters including view change, distortion, and illumination variations. Experimental results using real images demonstrated the efficiency of the proposed method. The nonlinear optimization takes some time but is fast enough to run as a batch process. We have shown two applications of the proposed method: measuring distortion parameters with changes in camera zooming and image mosaicing using corrected images.

So far the results of the correction have been evaluated qualitatively because the actual intrinsic camera parameters are unknown. A quantitative evaluation of the estimates is planned for the future.

Acknowledgements

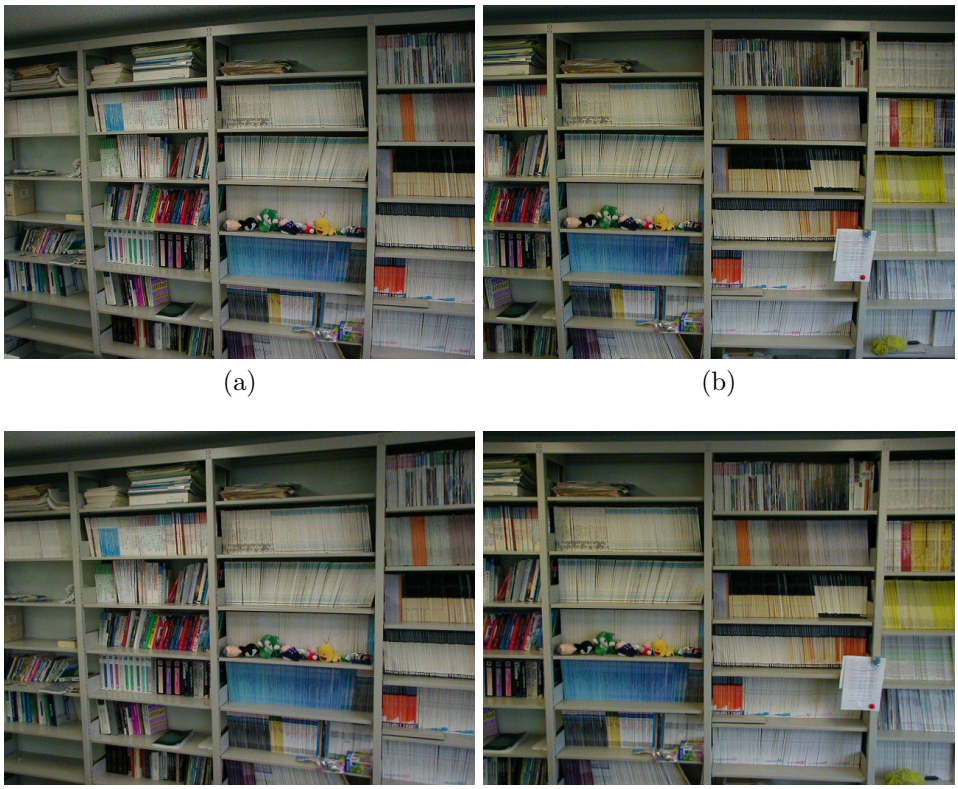
This research was supported in part by NEC Foundation For C&C Promotion.

Appendix A

The iterative procedure[21] to compute $\boldsymbol{p}^d$ from $\boldsymbol{p}^u$ for given parameters $\boldsymbol{\theta}^d$ is shown below. Unlike the formulation in section 2.2, coordinate system of the points is represented with respect to the image center (not to the top-left corner). Starting with $(x_{v0}, y_{v0}) = (x_u, y_u)$, the following equations are used to update (x_{vi}, y_{vi}).

$$x_{vi} = \frac{x_u}{1 + \kappa_1 r_{i-1}^2 + \kappa_2 r_{i-1}^4} \quad (1)$$

$$y_{vi} = \frac{y_u}{1 + \kappa_1 r_{i-1}^2 + \kappa_2 r_{i-1}^4} \quad (2)$$



(a) (b)

(c) (d)

Figure 12: (a) Left side and (b) right side of the original images of a bookshelf. (c) Left side and (d) right side of the corrected images. Estimated parameters : $\boldsymbol{\theta}^d = (5.15\text{e-}7, -5.06\text{e-}13, 298, 242, 0.975)$

where $r_i = \sqrt{x_{vi}^2 + y_{vi}^2}$. After k iterations the update converges (at most eight times iterations is enough[21] to achieve a good approximation), then we take $(x_d, y_d) = (s_x x_{vk}, y_{vk})$.

References

1. Y. Shibuya and I. Kumazawa, "Estimation of camera parameters and compensation of image distortion matching a priori known shape," *IEICE D-II*, vol. J83-D-II, no. 6, pp. 1460–1468, 2000. (in Japanese).
2. 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.
3. 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," in *Proc. of ICCV'99*, pp. 411–417, 1999.

(a)

(b)

Figure 13: Mosaicing of a bookshelf using (a) original images and (b) corrected images.

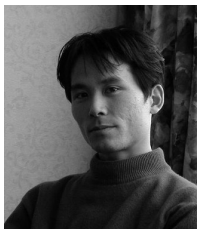
(a) (b)

Figure 14: Magnifications of the mosaic using (a) original distorted images and (b) corrected images.

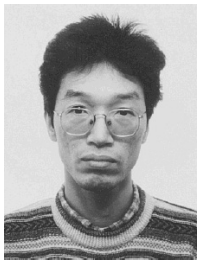
4. J. B. Shim, T. Mukai, and N. Ohnishi, "Improving the accuracy of 3D shape by fusing shapes obtained from optical flow," in *Proc. of CISST'99*, pp. 196–202, 1999.
5. O. D. Faugeras, ed., *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1993.
6. R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proc. of CVPR'86*, pp. 364–374, 1986.
7. R. K. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-d machine," *IEEE Trans. on PAMI.*, vol. 10, no. 5, pp. 713–720, 1988.
8. R. Willson, "Camera calibration using Tsai's method." ftp://ftp.vislist.com/SHAREWARE/CODE/CALIBRATION/Tsai-method-v3.0b3/, 1995.
9. G. R. Bradski and V. Pisarevsky, "Intel's computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture and object recognition," in *Proc. of CVPR2000*, vol. 2, pp. 796–797, 2000. http://www.intel.com/research/mrl/research/opencv/.
10. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on PAMI.*, vol. 22, no. 11, pp. 1330–1334, 2000.
11. Z. Zhang, "A flexible new technique for camera calibration," Tech. Rep. MSR-TR-98-71, Microsoft Research, 1998. http://research.microsoft.com/~zhang/.
12. G. P. Stein, "Lens distortion calibration using point correspondences," in *Proc. of CVPR'97*, pp. 602–608, 1997.
13. J. Heikkilä, "Geometric camera calibration using circular control points," *IEEE Trans. on PAMI.*, vol. 22, no. 10, pp. 1066–1077, 2000.
14. R. Swaminathan and S. K. Nayar, "Nonmetric calibration of wide-angle lenses and polycameras," *IEEE Trans. on PAMI.*, vol. 22, no. 10, pp. 1172–1178, 2000.
15. C. Matsunaga, Y. Kanazawa, and K. Kanatani, "Optimal grid pattern for automated camera calibration using cross ratio," *IEICE Trans. on Fundamentals*, vol. E83-A, no. 10, pp. 1921–1928, 2000.
16. Y. Xiong and K. Turkowski, "Creating image-based VR using a self-calibrating fisheye lens," in *Proc. of CVPR'97*, pp. 237–243, 1997.
17. G. Pedrick, *A First Course in Analysis*. Springer-Verlag, 1994.

18. M. H. Protter and C. B. Morrey, *A First Course in Real Analysis*. Springer-Verlag, 1977.
19. H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion correction," *IEEE Trans. on PAMI.*, vol. 21, no. 3, pp. 235–243, 1999.
20. R. Szeliski, "Video mosaics for virtual environment," *IEEE Computer Graphics and Applications*, vol. 16, no. 3, pp. 22–30, 1996.
21. R. Klette, K. Schlüns, and A. Koschan, *Computer Vision Three-Dimensional Data from Images*. Singapore: Springer-Verlag, 1998.
22. J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. on PAMI.*, vol. 14, no. 10, pp. 965–980, 1992.
23. C. C. Slama, ed., *Manual of Photogrammetry*. American Society of Photogrammetry, 1980.
24. T. Tamaki, T. Yamamura, and N. Ohnishi, "A method for compensation of image distortion with image registration," *IEICE Trans. Inf. Syst.*, vol. E84-D, no. 8, pp. 990–998, 2001. http://search.ieice.org/2001/files/e000d08.htm#e84-d,8,990.
25. M. J. Black, D. J. Fleet, and Y. Yacoob, "Robustly estimating changes in image appearance," *CVIU*, vol. 78, no. 1, pp. 8–31, 2000.
26. P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?," *IJCV*, vol. 28, no. 3, pp. 1–16, 1998.
27. G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. New York: Wiley, 1989.
28. H. S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. on PAMI.*, vol. 18, no. 8, pp. 814–830, 1996.
29. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*. Cambridge University Press, 1992.
30. R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York: McGraw-Hill, 1995.
31. D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proc. of ICPR'96*, pp. 73–80, 1996.
32. Sony Co., *EVI-D30/D31 Command List*, 1998. http://www.sony.co.jp/en/Products/ISP/docu_soft/index.html#manual.

Photo and Bibliography



Toru Tamaki received his B.E., M.S. and Ph.D degrees in information engineering from Nagoya University, Japan, in 1996, 1998 and 2001, respectively. Currently, he is a Research Associate at the Department of Information Engineering, Faculty of Engineering, Niigata University, Japan. From April 1999 to March 2001, he was with the Bio-Mimetic Control Research Center, RIKEN, Japan, as a Junior Research Associate. His research interests include computer vision and image recognition. He is a member of IEICE and ITEJ.



Tsuyoshi Yamamura received his B.E., M.E. and Ph.D degrees from Nagoya University, Japan, 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, Japan. He is engaged in research on Natural Language Processing and Visual Information Processing. He is a member of IEEE, IEICE and IPSJ.

Noboru Ohnishi received his B.E., M.S. and Ph.D degrees in electrical engineering from Nagoya university, Japan, in 1973, 1975 and 1984, respectively. In 1975, he was a Researcher of the Rehabilitation Engineering Center under the Ministry of Labor, Japan. From September 1992 to January 1993, he was with the Laboratory for Artificial Intelligence, the University of Michigan, as a Visiting Researcher. 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 is a Professor in the Center for Information Media Studies, Nagoya University. From 1993 to 2001, he concurrently held a Head of Laboratory for Bio-mimetic Sensory System at the 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, IEICE, ITE and IPSJ.