Hand–Eye Calibration Applied to Viewpoint Selection for Robotic Vision

Yuichi Motai, *Member, IEEE*, and Akio Kosaka, *Member, IEEE*

***Abstract*—Viewpoint calibration is a method to manipulate hand–eye for generating calibration parameters for active viewpoint control and object grasping. In robot vision applications, accurate vision sensor calibration and robust vision-based robot control are essential for developing an intelligent and autonomous robotic system. This paper presents a new approach to hand–eye robotic calibration for vision-based object modeling and grasping. Our method provides a 1.0-pixel level of image registration accuracy when a standard Puma/Kawasaki robot generates an arbitrary viewpoint. To attain this accuracy, our new formalism of hand–eye calibration deals with a lens distortion model of a vision sensor. Our most distinguished approach of optimizing intrinsic parameters is to utilize a new parameter estimation algorithm using an extended Kalman filter. Most previous approaches did not even consider the optimal estimates of the intrinsic and extrinsic camera parameters, or chose one of the estimates obtained from multiple solutions, which caused a large amount of estimation error in hand–eye calibration. We demonstrate the power of this new method for: 1) generating 3-D object models using an interactive 3-D modeling editor; 2) recognizing 3-D objects using stereovision systems; and 3) grasping 3-D objects using a manipulator. Experimental results using Puma and Kawasaki robots are shown.**

***Index Terms*—Hand–eye calibration, Kalman filter, lens distortion, multiple viewpoints, robot vision.**

I. Introduction

WE REPORT here an improved technique for calibrating hand–eye robotic vision systems. Our approach allows the system to compute robotic calibration parameters of the camera and the end-effector for multiple viewpoint generation after the system performs the vision-based camera calibration for only a small number of viewpoints. Ordinarily, if one wishes to integrate multiple views for acquiring a 3-D object model, one would separately carry out camera calibration for each viewpoint. An alternative approach consists of carrying out calibration for a certain number of designated viewpoints and using interpolation for other viewpoints. The disadvantages of this approach are: 1) overall low accuracy (partially dominant errors) and 2) spatial limitation of the viewpoint generation. In our proposed scheme, we use multiple viewpoints (at least three viewpoints) for calibration of the camera mounted on the gripper and optimally estimate all the necessary parameters for active control of viewpoints and precise object grasping. The advantages of our method are to minimize the camera calibration error by: 1) applying a lens distortion model and 2) optimizing the camera parameters with robotic arm kinematics.

Since camera calibration is fundamental to all phases of research in robot vision, much work has been done so far. Some of the earliest references to camera calibration in the computer vision literature are [7] and [11]. These assume a pinhole model for the camera, and, therefore, this assumption restricts the use of a wide range of views for 3-D reconstruction since the lens distortion can be severe in the peripheral of the camera view. Whereas few of these references deal with lens distortion of camera optics [23], others address the issue of efficient calculation of various camera parameters [25], [26].

Hand–eye calibration utilizes techniques of camera calibration and motion control of the robot hand. We have developed algorithms for hand–eye calibration by moving the camera to different viewpoints through robot arm motion and taking images of 3-D calibration patterns. The algorithms first estimate intrinsic and extrinsic camera parameters for each viewpoint and select the intrinsic camera parameters of one viewpoint as the truth-value of the intrinsic camera parameters for all viewpoints. The algorithms then estimate various transformation matrices for hand–eye calibration, including the transformation from the camera to the end-effector and the transformation from the robot base to the world coordinate frame.

When the lens distortion of the camera becomes severe, however, the estimated intrinsic camera parameters can greatly differ between views due to the instability of intrinsic camera parameter estimation (including lens distortion parameters). The selection of the intrinsic camera parameters from a single view leads to a large hand–eye calibration error, particularly for arbitrary viewpoints used in 3-D modeling of objects.

In this paper, to achieve accurate and reliable performance of active vision control, we have modeled a camera with lens distortion as well as optimally estimated camera calibration parameters by integrating those estimates obtained from multiple viewpoints using the Kalman filter approach. Kalman filters have been widely used in many industrial electronics, and some robotic applications similar to ours are found in [4], [15], [16], and [28]–[43].

Fig. 1 shows the image registration error of the overall hand–eye calibration for viewpoint generation. Small black circles are used for the camera calibration patterns to deal with lens distortion, and white cross-bars are superimposed to verify the accuracy of the method. As shown in this figure, we attain

Manuscript received December 20, 2006; revised February 5, 2008. First published March 21, 2008; current version published October 1, 2008. This work was supported by the Advanced Manufacturing Technology Department, Ford Motor Corporation.

Y. Motai is with the School of Engineering, University of Vermont, Burlington, VT 05405 USA (e-mail: ymotai@uvm.edu).

A. Kosaka is with Purdue University, West Lafayette, IN 47907 USA, and also with Future Creation Laboratory, Olympus Corporation, Shinjuku 163-0914, Japan (e-mail: kosaka@purdue.edu).

Digital Object Identifier 10.1109/TIE.2008.921255

0278-0046/$25.00 © 2008 IEEE

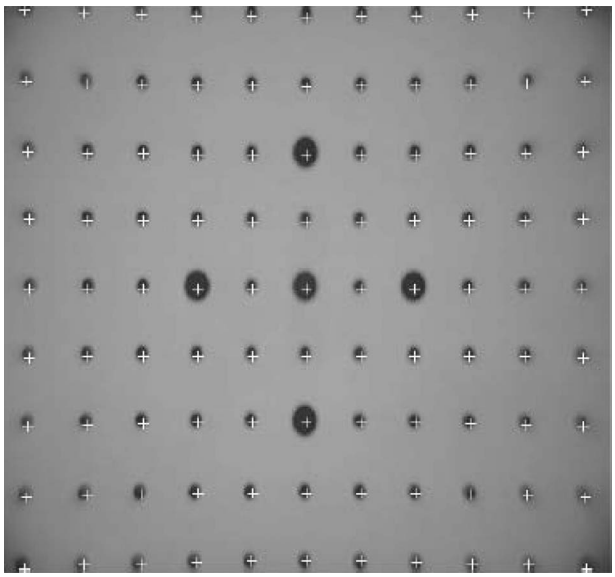
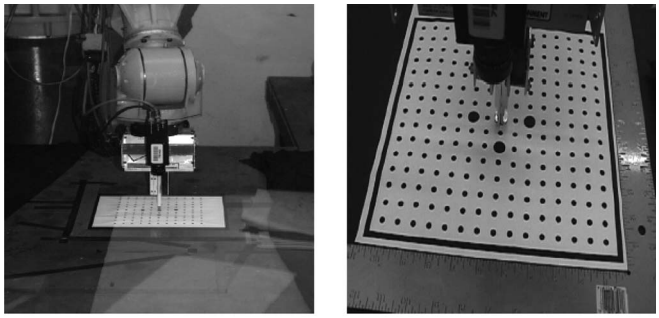


Fig. 1. Camera mounted on the gripper takes an image of calibration patterns of radius 2.0 mm at the distance of 0.2 m. The estimated calibration parameters from our method are used to produce the white cross-hairs, superimposed at the expected pattern centers.



(a) (b)

Fig. 2. Verification of hand–eye calibration using a stylus is shown. (a) Puma and (b) Kawasaki robots attempt to locate the tip of the stylus to the origin of the world coordinate frame by using the estimated calibration parameters.

the 1.0-pixel level of image registration accuracy (1-mm level of accuracy in the 3-D workspace) when the camera is moved to a position that was not used for calibration. Fig. 2 also shows the verification of our hand–eye calibration for object grasping. The robot attempts to move the tip of the stylus mounted on the gripper to the center of calibration patterns.

As several early studies demonstrated, the solution of the kinematics can be derived as a closed form using an inverse transformation matrix [10]. To determine the position and the orientation of the hand–eye camera with respect to the workspace, more recent issues are to solve the unknown transformation of the robot hand, which is formed as a homogeneous matrix equation $AX = XB$ [6], [21], [24] and to solve this equation using a quaternion approach [6], [8], [24]. In our method, we propose a modified algorithm using a nonlinear iteration method and then evaluate the new solution using our robot manipulation systems.

Here, we apply our hand–eye calibration technique to 3-D modeling and grasping of industrial objects. Our robotic calibration can be applied to develop a human-assisted model acquisition system [17]. In this project, objects are placed in the work area by a human, who then "guides" the system into establishing image-to-image and pose-to-pose correspondences. After a model is acquired in this manner during the test phase, the same object in a random pose is viewed from two viewpoints for pose calculation if the object is placed inside a known workspace or in the view of the single camera mounted on the robot hand. In the acquired model, the object surfaces that are appropriate for grasping with the robotic hand are marked. The computer vision method identifies which surface has grasping capabilities before using the robotic hand to pick up the object. Lastly, robotic manipulation is executed to grasp the object for which the grasping points have already been acquired.

We also seek an efficient viewpoint selection for imaging, so that our robot vision system can reliably compute an object's pose. The choice of an imaging viewpoint has been examined in the field of active vision [1], [2]. In these studies, a camera is manipulated to improve the quality of the perceptual results. Because we can compute calibration parameters at any camera position, having active control over the viewpoint direction increases machine perception ability [22].

In this paper, we will first present our new approach for the hand–eye calibration system and will then apply this calibration result to the human-assisted model acquisition system. Lastly, experimental results that verify the power of our approach will be shown.

II. Problem Statement of Hand–Eye Calibration

To generate a 3-D object model, we mount a monocular camera on the gripper of the robotic manipulator to capture views of the object from multiple viewpoints. It is desired that all multiple views be automatically captured using the robot, followed by calibration parameter calculation for model acquisition. For this strategy in 3-D modeling, hand–eye calibration is an important task to perform. To formulate the hand–eye calibration problem, we define the following coordinate frames as shown in Fig. 3: *Base* (x_B, y_B, z_B), *Tool* (x_T, y_T, z_T), *Approach* (x_A, y_A, z_A), *Grasping* (x_G, y_G, z_G), *Object* (x_O, y_O, z_O), *World* (x_W, y_W, z_W), *Camera* (x_C, y_C, z_C), and *Image* (u, v).

In our laboratory platform, robotic hand motion is fully determined using a Puma 761 or Kawasaki JS10 controller with the positioning commands $XYZOAT$ (which is a platform-dependent control parameter similar to $XYZ\varphi\theta\psi$). We will be able to determine the homogeneous transformation $\boldsymbol{H}_{\text{Base}}^{\text{Tool}}$ from the robot tool to the robot base coordinate frames [see later in (33) and (34)]. Thus, our hand–eye calibration problems are, first, to solve a camera (eye) calibration problem and then to solve a robotic (hand) calibration problem, described as follows:

1) determine the camera-image transformation g, namely

$$(u, v) = g(x_C, y_C, z_C) \tag{1}$$

2) determine the homogeneous transformations of $\boldsymbol{H}_{\text{Camera}}^{\text{Tool}}$ and $\boldsymbol{H}_{\text{Base}}^{\text{World}}$.

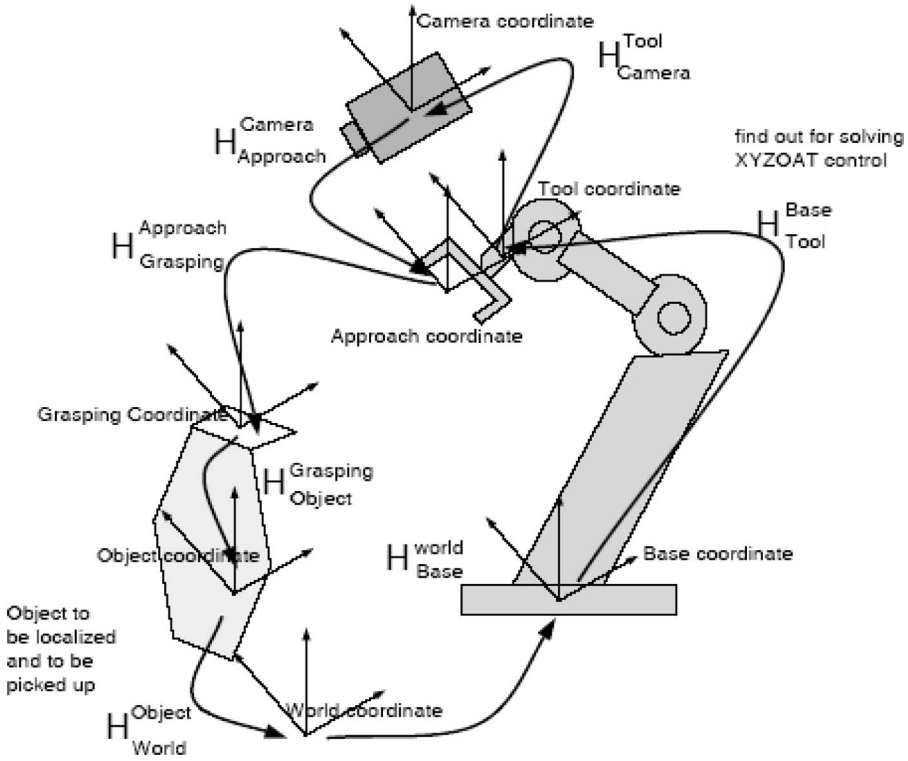


Fig. 3. Coordinate transitional illustration for robot manipulation.

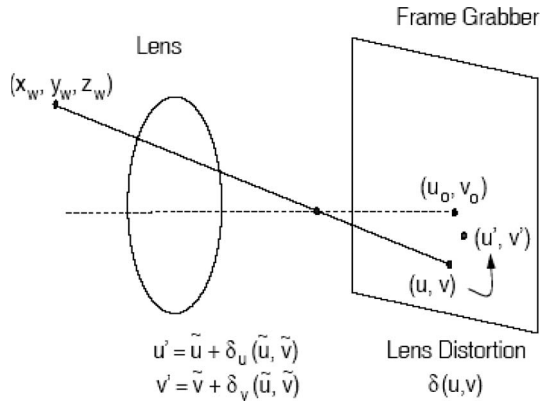


Fig. 4. Camera distortion model.

A. Camera Distortion Model

Before discussing the details of our calibration procedure, we will present the process of camera image formation that deals with lens distortion (Fig. 4). Let us first assume that the camera is modeled by a pinhole camera. In this case, the camera image coordinates (u_p, v_p) can be mapped into the normalized pinhole camera image coordinates (u', v') by the intrinsic camera parameters, the magnification factors α_u and α_v, and the image center coordinates (u_0, v_0) as follows:

$$u' = \frac{u_p - u_0}{\alpha_u} = \frac{x_C}{z_C}$$
$$v' = \frac{v_p - v_0}{\alpha v} = \frac{y_C}{z_C}. \tag{2}$$

Please note that, in general, the camera image coordinates (u_p, v_p) under the pinhole camera image coordinates are not observable. Lens distortion adds to the normalized pinhole camera image coordinates a deviation of (u', v'). Let $(\tilde{u}, \tilde{v})$ be the normalized camera image coordinate frame with lens distortion. Then, we can represent the relationship between (u', v') and $(\tilde{u}, \tilde{v})$ by the following [25]:

$$u' = \frac{x_C}{z_C} \approx \tilde{u} + k_1 \tilde{u}(\tilde{u}^2 + \tilde{v}^2)$$
$$v' = \frac{y_C}{z_C} \approx \tilde{v} + k_1 \tilde{v}(\tilde{u}^2 + \tilde{v}^2) \tag{3}$$

where the coefficient k_1 for the deviation represents the parameter for the radial distortion of the camera, and $(\tilde{u}, \tilde{v})$ is computed from the actual camera image coordinates (u, v) as follows:

$$\tilde{u} = \frac{u - u_0}{\alpha_u}$$
$$\tilde{v} = \frac{v - v_0}{\alpha_v}. \tag{4}$$

In this paper, we will call $\boldsymbol{s} = [\alpha_u, \alpha_v, u_0, v_0, k_1]^T$ the intrinsic camera parameters. Note that these intrinsic camera parameters will remain constant during the hand–eye calibration process. As shown in [25], tangential distortion is not dominant for robotic applications. Therefore, we only model here a radial distortion case. However, our method is applicable to lens distortion models dealing with radial and tangential cases.

We now consider multiple viewpoints by moving a camera in the world coordinate frame. At viewpoint i, the camera location

in the world coordinate frame is specified by the homogeneous transformation $\boldsymbol{H}_{\text{Camera}}^{\text{Tool}}$, which consists of the rotation matrix $\boldsymbol{R}^i = (r_{km}^i)$ and the translation vector $\boldsymbol{t}^i = [t_x^i, t_y^i, t_z^i]^T$. Then from (3), the 3-D point (x_W, y_W, z_W) in the world coordinate frame is mapped onto the camera image frame as follows:

$$u' = \frac{r_{11}^i x_W + r_{12}^i y_W + r_{13}^i z_W + t_x^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} \approx \tilde{u} + k_1 \tilde{u}(\tilde{u}^2 + \tilde{v}^2) \tag{5}$$

$$v' = \frac{r_{21}^i x_W + r_{22}^i y_W + r_{23}^i z_W + t_{xy}^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} \approx \tilde{v} + k_1 \tilde{v}(\tilde{u}^2 + \tilde{v}^2) \tag{6}$$

where the rotation matrix $\boldsymbol{R}^i$ is specified by independent yaw–pitch–roll angles φ_x, φ_y, and φ_z. We will call $\boldsymbol{e}^i = [\varphi_x^i, \varphi_y^i, \varphi_z^i, t_x^i, t_y^i, t_z^i,]^T$ the *extrinsic camera parameters* for viewpoint i.

Having the intrinsic camera parameters $\boldsymbol{s}$ and the view-dependent extrinsic camera parameters $\boldsymbol{e}^i$, we can always determine the mapping of the world coordinate frame 3-D points into the camera coordinate frame. To compute the camera-image transformation, i.e., 2-D image coordinates (u, v) from the corresponding 3-D point (x_W, y_W, z_W) in the world frame, we take the following two steps.

1) Compute the normalized camera image coordinates $(\tilde{u}, \tilde{v})$ by solving the implicit forms of (5) and (6) using an appropriate iterative gradient method, for example, the Newton method, with initial values $(\tilde{u}, \tilde{v}) = (u', v')$.
2) Compute the actual image coordinates (u, v) from $(\tilde{u}, \tilde{v})$ using (4).

III. Solution to Hand–Eye Calibration

Most of the previous work on hand–eye calibration did not take into account the lens distortion for hand–eye calibration since their main interest was visual servoing for robotic manipulation [22]. In most cases, these methods only utilized central regions of camera images, where the pinhole camera is generally sufficient to model the 3-D–2-D point projection.

When using a camera with a small focal length and wider views for 3-D object modeling, lens distortion becomes a significant issue for calculating precise 3-D measurements or for object modeling using multiple view observations. We propose here an efficient and accurate approach that can deal with lens distortion for the camera model while still producing precise determination of robotic hand and camera transformations.

In our hand–eye calibration strategy, as shown in Fig. 5, we take the following three steps.

Step 1) We locate a calibration pattern board at different heights (z_W values) in the world coordinate frame and then take snapshots of the calibration pattern board from multiple viewpoints by moving the robot end-effector. We then independently estimate the intrinsic and extrinsic camera parameters for each viewpoint, assuming that intrinsic camera parameters are not actually equal.

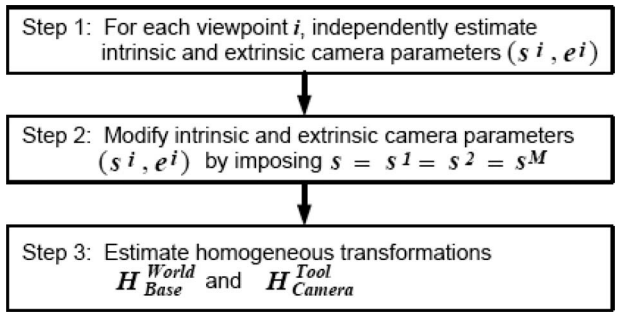


Fig. 5. Calibration steps.

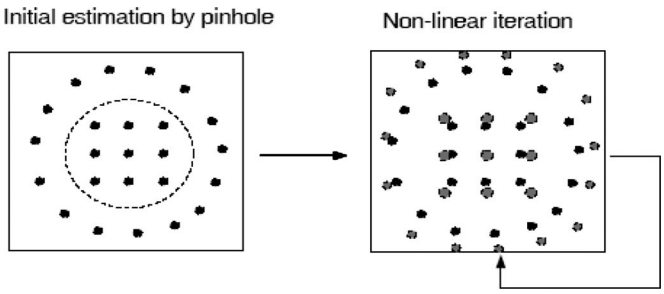


Fig. 6. Radial lens distortion illustration by iteratively minimizing (8). The left-hand side shows the initial estimation corresponding to the pinhole camera model. The right-hand side shows how the radial distortion is handled via nonlinear iterations.

Step 2) We reestimate intrinsic and extrinsic camera parameters by imposing the constraint that all intrinsic camera parameters be equal.

Step 3) We compute the homogeneous transformation between the robot tool and the camera coordinate frames as well as that between the robot base and the world coordinate frames.

Most of the previous work on hand–eye calibration did not take into account the lens distortion for hand–eye calibration since their main interest was visual servoing for robotic manipulation [22]. In most cases, these methods only utilized central regions of camera images, where the pinhole camera is generally sufficient to model the 3-D–2-D point projection (Fig. 6).

A. Step 1: Initial Estimation of Camera Parameters

In Step 1, we generate multiple viewpoints (M viewpoints in total) by moving the camera with the robot and taking snapshots of the planar calibration pattern board, which is located at different heights z_W specified in the world coordinate frame. The calibration patterns consist of small black circles (N circles in total), and the 3-D coordinates of the centroids of the calibration patterns are measured in the world frame. By analyzing the snapshots of the calibration patterns by a computer, we can estimate the 2-D image coordinates of the centroids in the camera image frame. For each viewpoint i and each circle j, let (x_j^i, y_j^i, z_j^i) and (u_j^i, v_j^i) be the 3-D coordinates of the calibration pattern centroids in the world coordinate frame and the corresponding 2-D image coordinates in the camera image frame measured and accumulated

from various heights, respectively. Then, from (5) and (6), we obtain

$$\begin{aligned}\frac{r_{11}^i x_W + r_{12}^i y_W + r_{13}^i z_W + t_x^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} &= \tilde{u}_j^i + k_1 \tilde{u}_j^i \left((\tilde{u}_j^i)^2 + (\tilde{v}_j^i)^2\right) \\ \frac{r_{21}^i x_W + r_{22}^i y_W + r_{23}^i z_W + t_y^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} &= \tilde{v}_j^i + k_1 \tilde{v}_j^i \left((\tilde{u}_j^i)^2 + (\tilde{v}_j^i)^2\right)\end{aligned} \tag{7}$$

$$\tilde{u}_j^i = \frac{u_j^i - u_0}{\alpha_u}$$

$$\tilde{v}_j^i = \frac{v_j^i - v_0}{\alpha_v}.$$

In Step 1, we first independently estimate intrinsic camera parameters $\boldsymbol{s}^i = [\alpha^i u, \alpha^i v, u_0^i, v_0^i, k_1^i]^T$ and extrinsic camera parameters $\boldsymbol{e}^i = [\varphi^i x, \varphi^i y, \varphi^i z, t^i x, t^i y, t^i z]^T$ for different viewpoints i. To do this, we apply Weng's algorithm, a nonlinear iterative method that is described in detail in [25]. The main contribution of the algorithm is that this problem is converted into an optimization problem that minimizes the objective function of the image registration error between the 2-D measurement coordinates $(u^i j, v_j^i)$ and the 2-D projected coordinates $(u^i j, v_j^i)$ based on the parametric representation of intrinsic and extrinsic camera models $\boldsymbol{s}^i$ and $\boldsymbol{e}^i$, respectively, computed from (7).

The objective function is defined by

$$f = \sum_j \left\{ \left(u_j^i - u_j^{-i}\right)^2 + \left(v_j^i - v_j^{-i}\right)^2 \right\} \tag{8}$$

which should be minimized with respect to $\boldsymbol{s}^i$ and $\boldsymbol{e}^i$.

B. Step 2: Integration of Camera Parameter Estimates From Multiple Views

In the previous step, intrinsic camera parameters $\boldsymbol{s}^i$ $(i = 1, 2, \ldots, M)$ are not necessarily equal since, for this algorithm, the parameters are independently estimated for viewpoint i. In Step 2, we integrate all estimates to obtain an optimal estimate of intrinsic and extrinsic camera parameters $\boldsymbol{s}$ and $\boldsymbol{e}^i$ by a nonlinear iteration algorithm. As for the initial estimates for the iteration, we use the estimates $\boldsymbol{s}^i$ and $\boldsymbol{e}^i$ obtained in Step 1. More specifically, we attempt to minimize f in (8) under the following additional constraint:

$$\boldsymbol{s} = \boldsymbol{s}^1 = \boldsymbol{s}^2 = \boldsymbol{s}^M. \tag{9}$$

We also apply an iterative technique of nonlinear optimization to attain a robust and accurate estimate. In this case, we have a good initial estimate for $\boldsymbol{s}$ and $\boldsymbol{e}^i$ $(i = 1, 2, \ldots, M)$; then, we apply an extended Kalman filter-based updating scheme [13], so that sequential updating of parameters can be attained along with outlier elimination. In our implementation of the extended Kalman filtering, for each image measurement point (u_j^i, v_j^i), we have a pair of constraint equations as follows:

$$\boldsymbol{f} \equiv \begin{bmatrix} \frac{r_{11}^i x_W + r_{12}^i y_W + r_{13}^i z_W + t_x^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} - \left(\tilde{u}_j^i + k_1 \tilde{u}_j^i \left((\tilde{u}_j^i)^2 + (\tilde{v}_j^i)^2\right)\right) \\ \frac{r_{21}^i x_W + r_{22}^i y_W + r_{23}^i z_W + t_x^i}{r_{31}^i x_W + r_{32}^i y_W + r_{33}^i z_W + t_z^i} - \left(\tilde{v}_j^i + k_1 \tilde{v}_j^i \left((\tilde{u}_j^i)^2 + (\tilde{v}_j^i)^2\right)\right) \end{bmatrix} = 0 \tag{10}$$

where the image measurement $(u^i j, v^i j)$ is converted to the normalized image point $(\tilde{u}^i j, \tilde{v}^i j)$ as (7). We define the parameter vector $\boldsymbol{p}$ to be estimated as follows:

$$\boldsymbol{p} = [s \quad e^1 \quad e^2 \quad \cdots \quad e^m] \tag{11}$$

and measurement vector $z = (u^i j, v^i j)$, and we associate the mean $\overline{\boldsymbol{p}}$ and the error covariance matrix Σ. Then, the extended Kalman filter updating from $(\overline{\boldsymbol{p}}, \Sigma)$ to $(\overline{\boldsymbol{p}}_{\text{new}}, \Sigma_{\text{new}})$ is, for each iteration, expressed by

$$\overline{\boldsymbol{p}}_{\text{new}} = \overline{p} - K(M\overline{p} - y) \tag{12}$$

$$\Sigma_{\text{new}} = (I - KM)\Sigma \tag{13}$$

where

$$\boldsymbol{f} = \boldsymbol{M}\overline{\boldsymbol{p}} - \boldsymbol{y} \tag{14}$$

$$\boldsymbol{K} = \Sigma \boldsymbol{M}^T (\boldsymbol{W} + \boldsymbol{M}\Sigma\boldsymbol{M}^T) \tag{15}$$

$$\boldsymbol{M} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} \tag{16}$$

$$\boldsymbol{W} = \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}\right) \boldsymbol{R} \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}\right)^T. \tag{17}$$

Note that $\boldsymbol{y}$ is obtained as the linearization of the constraint equation $\boldsymbol{f}$, $\boldsymbol{K}$ is the Kalman gain, and $\boldsymbol{R}$ is the error measurement covariance matrix. To stabilize the estimation, Kalman updating is sequentially performed for each measurement point $(u^i j, v^i j)$ by randomizing the order of indexes i and j. During the sequential updating, we also check whether outliers exist in the image measurements, as described in [13]. In our case, we have a large number of measurements of image calibration points (uj, vj) $(j = 1, 2, \ldots, N)$, where N is typically around 1000. The extended Kalman filter-based updating can be implemented for a small number of constraints for each iteration of updating; more specifically, only two degrees of freedom for the constraints are necessary as seen in the constraint equation (10). Note that the parameters to be estimated are $(\boldsymbol{s}, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e_m})$ for m multiple views, and the dimension of the parameter vector $\boldsymbol{p}$ is $5 + 6m$. The constraint equation acquired from (10) is simply 2-D associated with the image measurements (uj, vj) for each sequential updating. This reduction of dimensionality greatly helps the reduction of computational complexity.

We would like to mention here that our approach of optimizing intrinsic parameters from multiple views is important. Compared to previous approaches that do not consider the optimal estimates of the unique intrinsic camera parameters and multiple extrinsic camera parameters in multiple views, our proposed method is chosen as a good tool to move the camera to an arbitrary viewpoint. Many studies on visual servoing

for robotic manipulation simply choose one of the intrinsic parameters from multiple solutions. This causes a large amount of an estimation error for hand–eye calibration, particularly for scenarios with severe lens distortion. As our experimental results later show, this step significantly improves the accuracy of camera calibration.

C. Step 3: Robotic Calibration

Once the experimental process has provided the camera parameters $\boldsymbol{s}$ and $\boldsymbol{e}^i$ $(i = 1, 2, \ldots, M)$, the relationship between coordinates in Fig. 3 will be obtained as each component transformation matrix such as $\boldsymbol{H}^{\mathrm{Tool}}_{\mathrm{Camera}}$, $\boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}}$, and $\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}$. Note that $\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Camera}_i}$ is derived from the extrinsic camera parameters $\boldsymbol{e}^i$. The transitional relationship between those transformation matrices is shown in the following:

$$\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Camera}_i} = \boldsymbol{H}^{\mathrm{Tool}_i}_{\mathrm{Camera}_i} \boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}_i} \boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}. \tag{18}$$

Since one camera setting is fixed in the tool position, the images of different viewpoints are produced by controlling the tool position. Given the positioning control parameters, we can specify the transformation matrix $\boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}_i}$ [10]. The two unknown transformation matrices in (18) are $\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}$ and $\boldsymbol{H}^{\mathrm{Tool}_i}_{\mathrm{Camera}_i}$. Note that since the transformation from Tool coordinates to Camera coordinates is viewpoint independent, we can denote an invariant matrix as $\boldsymbol{H}^{\mathrm{Tool}}_{\mathrm{Camera}}$. Using the multiple camera calibration results, we solve $\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}$ first, use this matrix, and then compute the last unknown matrix $\boldsymbol{H}^{\mathrm{Tool}}_{\mathrm{Camera}}$. Since we approximately know that the world z-axis and the robot base z-axis are almost parallel, $\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}$ is nearly equal to an identity matrix in a rotational part and some values in a translational part. For simplicity, in (19), we show the two pairs of viewpoint cases for five viewpoints, denoted with i and j (thus, ${}_5\mathrm{C}_2$ combinatory possibilities), for the transitional relationship, i.e.,

$$\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Camera}_{i/j}} = \boldsymbol{H}^{\mathrm{Tool}}_{\mathrm{Camera}} \boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}_{i/j}} \boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}}. \tag{19}$$

We eliminate $\boldsymbol{H}^{\mathrm{Tool}}_{\mathrm{Camera}}$ from viewpoints i and j in (19) and obtain

$$\boldsymbol{A}_{ij}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}_{ij} = \boldsymbol{O} \tag{20}$$

where the notations are defined as follows:

$$\boldsymbol{X} = \boldsymbol{H}^{\mathrm{World}}_{\mathrm{Base}} \tag{21}$$

$$\boldsymbol{A}_{ij} = \left(\boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}_i}\right)^{-1} \boldsymbol{H}^{\mathrm{Base}}_{\mathrm{Tool}_j} \tag{22}$$

$$\boldsymbol{B}_{ij} = \left(\boldsymbol{H}^{\mathrm{World}}_{\mathrm{Camera}_i}\right)^{-1} \boldsymbol{H}^{\mathrm{World}}_{\mathrm{Camera}_j}. \tag{23}$$

In the above notation, the 3×3 rotation matrix $\boldsymbol{R}$ and the 3×1 translation vector $\boldsymbol{t}$ can be separately expressed in a homogeneous form, so that the equation can be decomposed as follows:

$$\boldsymbol{R}^A_{ij}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^B_{ij} \tag{24}$$

$$\boldsymbol{R}^A_{ij}\boldsymbol{t} + \boldsymbol{t}^A_{ij} = \boldsymbol{R}\boldsymbol{t}^B_{ij} + \boldsymbol{t}. \tag{25}$$

Therefore, our first problem is to estimate $\boldsymbol{R}$. Although an analytical derivation of solutions to $\boldsymbol{R}$ in (24) is generally difficult [6], [21], we can apply a simple iterative approach using the following method. Let ψ_x, ψ_y, and ψ_z be the yaw–pitch–roll angles associated with the rotational transformation from the world coordinate to the base coordinate associated with $\boldsymbol{R}$ defined by (26), shown at the bottom of the page. We apply the Broyden–Fletcher–Goldfarb–Shanno optimization algorithm [5] to obtain the solution of $\boldsymbol{q} = (\psi_x, \psi_y, \psi_z)^T$, which minimizes the following objective function:

$$f(\boldsymbol{q}) = \sum_{i,j(i\neq j)} \left\| \boldsymbol{R}^A_{ij}\boldsymbol{R} - \boldsymbol{R}\boldsymbol{R}^B_{ij} \right\|^2. \tag{27}$$

For this nonlinear iteration method, we can appropriately select an initial estimate for $\boldsymbol{q}$ since we know the approximate position of the robot base with respect to the world coordinate frame. Our initial estimate may not be sufficiently close to the true value for us to apply a simple gradient-descent method like the Newton method; therefore, we utilize the Broyden–Fletcher–Goldfarb–Shanno optimization method, which is known to have a stable convergence [3].

Once $\boldsymbol{R}$ is estimated from (27), we have the following translation equation from (25):

$$\left(\boldsymbol{I} - \boldsymbol{R}^A_{ij}\right) t = \boldsymbol{t}^A_{ij} - \boldsymbol{R}\boldsymbol{t}^B_{ij}. \tag{28}$$

Then, we have the following:

$$\boldsymbol{C}\,\boldsymbol{t} = \boldsymbol{d} \tag{29}$$

where the following notations are used:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I} - \boldsymbol{R}^A_{1,2} \\ \vdots \\ \boldsymbol{I} - \boldsymbol{R}^A_{m-1,m} \end{bmatrix} \quad \boldsymbol{d} = \begin{bmatrix} \boldsymbol{t}^A_{1,2} - \boldsymbol{R}\boldsymbol{t}^B_{1,2} \\ \vdots \\ \boldsymbol{t}^A_{m-1,m} - \boldsymbol{R}\boldsymbol{t}^B_{m-1,m} \end{bmatrix} \tag{30}$$

where $\boldsymbol{m}$ is the number of viewpoints. The solution to (29) with respect to $\boldsymbol{t}$ is given by

$$\boldsymbol{t} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{d} \tag{31}$$

which is equivalent to the solution that minimizes $\|\boldsymbol{C}\boldsymbol{t} - \boldsymbol{d}\|^2$. This completes the estimates of $\boldsymbol{X}$ in (20).

$$\boldsymbol{R} = \begin{bmatrix} \cos\psi_z \cos\psi_y & \cos\psi_z \sin\psi_y \sin\psi_x - \sin\psi_z \cos\psi_x & \cos\psi_z \sin\psi_y \cos\psi_x + \sin\psi_z \sin\psi_x \\ \sin\psi_z \cos\psi_y & \sin\psi_z \sin\psi_y \sin\psi_x + \cos\psi_z \cos\psi_x & \sin\psi_z \sin\psi_y \cos\psi_z - \cos\psi_z \sin\psi_x \\ -\sin\psi_y & \cos\psi_y \sin\psi_x & \cos\psi_y \cos\psi_x \end{bmatrix} \tag{26}$$

Note that the estimation of $\boldsymbol{H}^{\text{Tool}}_{\text{Camera}}$ is derived as the same from (20) in the case of $\boldsymbol{H}^{\text{World}}_{\text{Base}}$. When the robot tool is moving toward a specific position, we have the robot positioning control parameter and its corresponding homogeneous transformation $\boldsymbol{H}^{\text{Base}}_{\text{Tool}_i}$. From (18), the three transformation matrices $\boldsymbol{H}^{\text{Tool}}_{\text{Camera}}$, $\boldsymbol{H}^{\text{Base}}_{\text{Tool}_i}$, and $\boldsymbol{H}^{\text{World}}_{\text{Base}}$ will produce the desired transformation matrix $\boldsymbol{H}^{\text{World}}_{\text{Camera}}$. Therefore, we can now generate both intrinsic and extrinsic camera parameters for any tool positions if we have the end-effector positioning parameters. The output of our solution is the optimized intrinsic camera parameters $\boldsymbol{s}$ and the computed extrinsic camera parameters $\boldsymbol{e}^i$ at any arbitrary tool position at i. This eliminates the need for recalibration as the robot end-effector is moved to different positions.

IV. Viewpoint Generation by Robotic Motion

The purpose of hand–eye calibration is to generate viewpoints and to obtain all parameters for transformations that are associated with viewpoints. In our case, we would like to model 3-D objects located in the world coordinate frame by moving the camera to arbitrary viewing positions specified in the world coordinate frame. To achieve this, we need the inverse kinematics to locate the camera at arbitrary positions and orientations in the world coordinate frame by moving the end-effector of the robot. Therefore, the problem is specified as follows.

- Given a homogeneous transformation from the world coordinate frame to the camera coordinate frame, generate the robot motion commands.

Therefore, the first step of our viewpoint generation problem is how to compute $\boldsymbol{H}^{\text{Tool}}_{\text{Base}}$ using $\boldsymbol{H}^{\text{Camera}}_{\text{World}}$ based on extrinsic camera parameters $\boldsymbol{e}$. Because the robot hand is controlled by $\boldsymbol{H}^{\text{Tool}}_{\text{Base}}$, this can be done by computing

$$\boldsymbol{H}^{\text{Tool}}_{\text{Base}} = \boldsymbol{H}^{\text{World}}_{\text{Base}} \boldsymbol{H}^{\text{Camera}}_{\text{World}} \boldsymbol{H}^{\text{Tool}}_{\text{Camera}} \tag{32}$$

where $\boldsymbol{H}^{\text{World}}_{\text{Base}}$ and $\boldsymbol{H}^{\text{Tool}}_{\text{Camera}}$ are determined using the camera calibration methods discussed in Step 3 in Section III.

The second step of our viewpoint generation problem is to compute the robotic motion parameters from $\boldsymbol{H}^{\text{Tool}}_{\text{Base}}$. This is a well-known inverse kinematics problem for various robotic manipulators. In our case, we apply our method with the following two robots.

A. Puma 761

Puma 761 is a Puma robot hand, whose end-effector tool position is controlled by the six parameters $XYZOAT$, and is expressed in terms of the homogeneous transformation matrix in (33), shown at the bottom of the page.

Given $\boldsymbol{H}^{\text{Tool}}_{\text{Base}}$, $XYZOAT$ parameters are computed for each element of (33).

A detailed element computation for the Puma robotic platform can be found in [10].

B. Kawasaki js10

In the Kawasaki JS10 Trellis controller, the 3×3 rotation matrix portion in (33) is represented by roll φ, pitch θ, and yaw ψ as given in (34), shown at the bottom of the page.

The parameters φ, θ, and ψ were derived by transforming the elements of the above matrix with respect to the arc tangent.

V. Experimental Results

We implemented the proposed calibration method on a SUN Workstation (the implemented source code in this paper is available by request). We performed the evaluation experiments for the Puma and Kawasaki platforms. Several minutes of computation were required to optimally estimate the camera parameters $\boldsymbol{s}$ and $\boldsymbol{e}^i$ and homogeneous transformation matrices $\boldsymbol{H}^{\text{Tool}}_{\text{Camera}}$ and $\boldsymbol{H}^{\text{World}}_{\text{Base}}$. In this experiment, we used a Sony DC-47 monocular 1/3 in a charge-coupled device camera with a Pulnix lens of focal length 16 mm. We selected five viewpoints to estimate the hand–eye calibration parameters. All of five viewpoints were set up in such a manner that each was directed to the world origin at the same height in the world coordinate frame. One camera viewpoint was located directly above a pattern board, and the other viewpoints were symmetrically offset along the $\boldsymbol{X}$ and $\boldsymbol{Y}$ world coordinates, as shown in Fig. 7.

For each viewpoint, we took snapshots of a planar calibration pattern board at three different levels of z-values specified in the world frame. The calibration plane was moved in the z_w axis to the z_W values 0.001, 0.12325, and 0.2465 m. The spacing of the points in the calibration pattern spacing was 0.0175 m.

$$\boldsymbol{H}^{\text{Tool}}_{\text{Base}} = \begin{bmatrix} \cos O \sin T - \sin O \sin A \cos T & \cos O \cos T + \sin O \sin A \sin T & \sin O \cos A & X \\ \sin O \sin T + \cos O \sin A \cos T & \sin O \cos T - \cos O \sin A \sin T & -\cos O \cos A & Y \\ -\cos A \cos T & \cos A \cos T & -\sin A & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

$$\boldsymbol{H}^{\text{Tool}}_{\text{Base}} = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos + \sin\phi\sin\psi & X \\ \sin\phi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & Y \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{34}$$

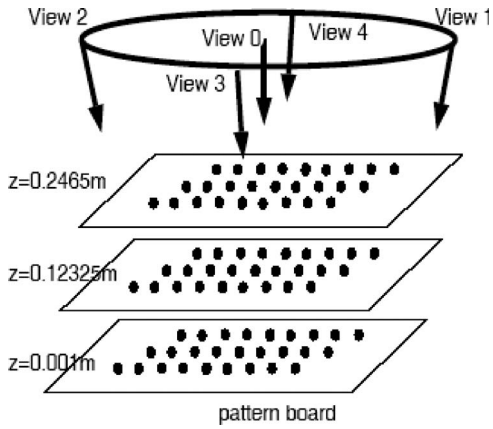


Fig. 7. Five image viewpoints used for imaging an object.

As long as the noncoplanar points are located in the roughly 3×3 m^2 workspace, then all of the camera parameters can be estimated. This is different from typical self-calibration because the z_w motion is not necessarily parallel to the camera optic axis. We then independently estimated the intrinsic camera parameters $\boldsymbol{s}^i$ and the extrinsic camera parameters $\boldsymbol{e}^i$. Next, we integrated multiple view-dependent camera parameters to estimate the optimal intrinsic camera parameters $\boldsymbol{s}$ and to modify individual extrinsic camera parameters $\boldsymbol{e}^i$. Last, the hand–eye homogeneous transformations, such as $\boldsymbol{H}_{\text{Camera}}^{\text{Tool}}$ and $\boldsymbol{H}_{\text{Base}}^{\text{World}}$, were estimated.

A. Evaluation of Hand–Eye Calibration by Image Registration

The accuracy of the estimated calibration parameters was evaluated in the 2-D image plane by generating arbitrary robot hand motions and by projecting known 3-D coordinates of the calibration patterns (white cross-bars) onto the originally captured images of calibration patterns. Fig. 8 shows the results of image registration accuracy. We numerically analyzed the error of calibration parameters with pixel deviation between the projected points and the actual image points. In the Puma platform, the average error was 1.38 ± 0.79 pixel/marker, as the typical image shown in Fig. 1 (Section I). In the Kawasaki platform, several other viewpoint images were also evaluated as in Fig. 8.

The image registration errors for all views are shown in Fig. 9 (the Kawasaki robot platform). The image registration results were compared to alternative solutions in Table I, which included the method without lens distortion. Modeling the registration results with lens distortion provides the best accuracy, particularly if wider camera image views are used, as shown in Fig. 8 and Table I.

B. Evaluation for 3-D Reconstruction

The calibration results were also evaluated for 3-D reconstruction by integrating views from multiple viewpoints. We placed a calibration pattern board with 225 black markers at three different heights in the world coordinates and took snapshots of this calibration board from five different viewpoints. We then extracted markers and estimated the 3-D coordinates of the black markers in the world coordinate frame using the stereo

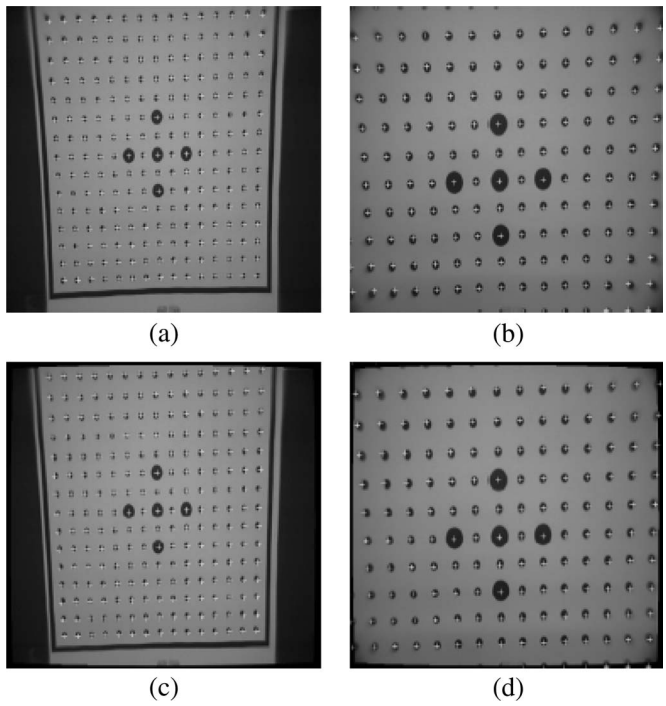


Fig. 8. Verification of superimposed maker patterns at two different z values in the Kawasaki platform. For (a) and (b), the projection matrix is determined using the results of the pinhole camera model. For (c) and (d), the results of the radial distortion camera model are used.

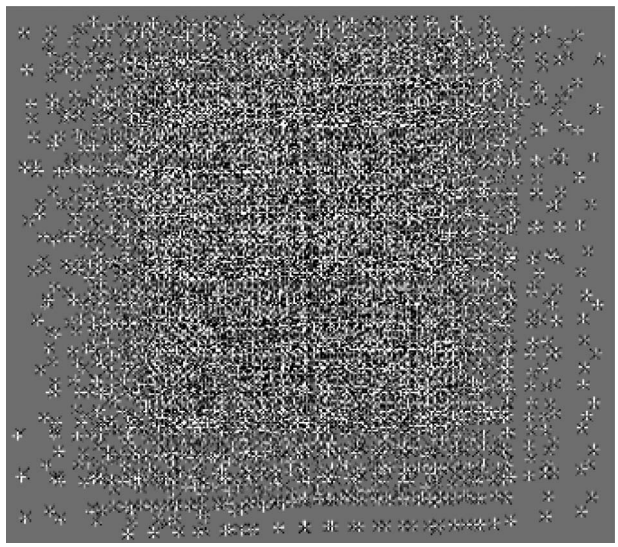


Fig. 9. Integration of all superimposed images using estimated distortion camera parameters in the Kawasaki platform. The white cross-bars "+" represent the superimposition of the pattern using the estimated transformation. The black overlaps "×" represent the actual extraction of the patterns.

TABLE I
IMAGE REGISTRATION ERROR

Platform	Method	Average Error (pixel)
Puma	Our method with distortion	1.38 ± 0.79
	Our method without distortion	1.71 ± 0.91
Kawasaki	Our method with distortion	1.50 ± 0.54
	Our method without distortion	1.68 ± 1.21

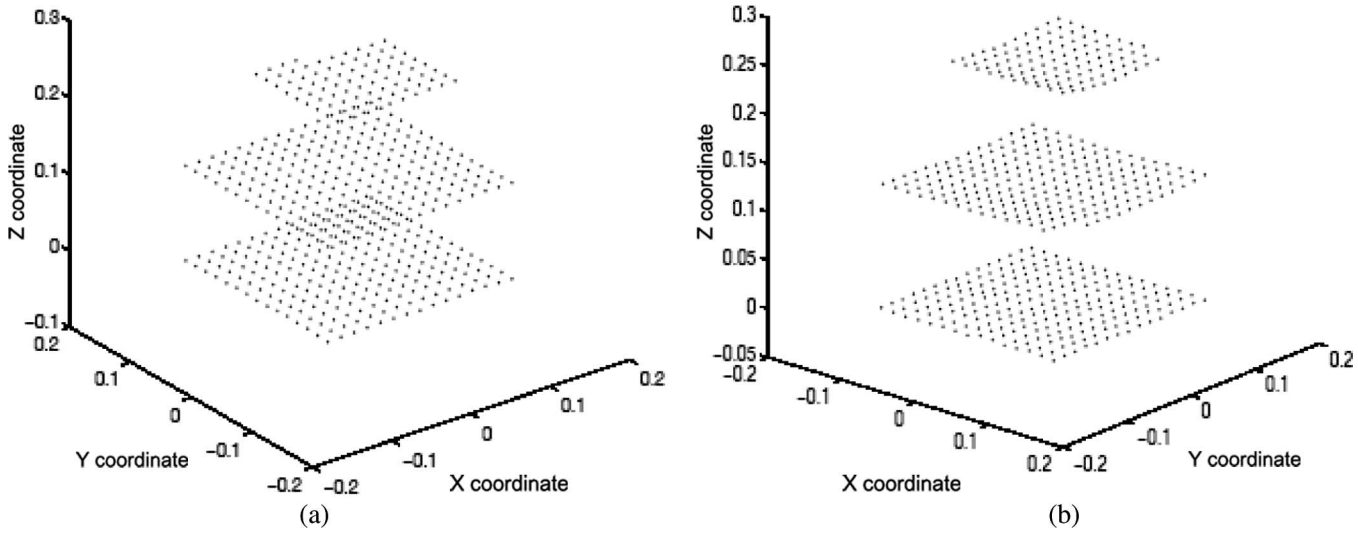


Fig. 10. Three-dimensional reconstruction of the maker patterns from three viewpoint images using hand–eye calibration results (a) with eliminating distortion and (b) affected by distortion. The stereo correspondence pattern points in the three images were identified and computed the world coordinate values (in meters) with depth value z. The marker point spacing in 3-D was $z = 0.001$, 0.12325, and 0.2465 m. The standard deviations of z were (a) 0.00698 m and (b) 0.00885 m, which demonstrates a 21.1% improvement when lens distortion is taken into account.

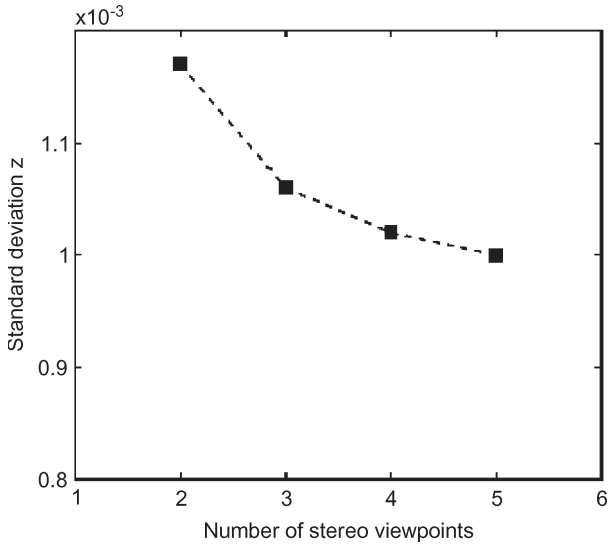


Fig. 11. Standard deviation of z values versus the number of stereo images used.

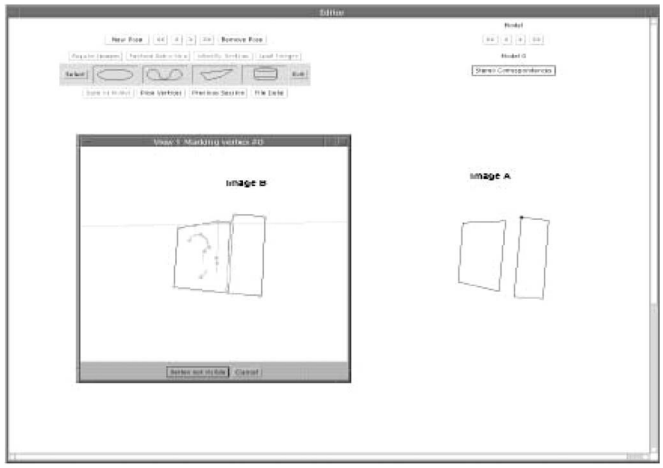


Fig. 12. Stereo correspondence in human–computer interaction editor. The feature extraction and correspondences in multiple views are established using the human-in-the-loop approach. The features are computed in the 3-D world coordinate frame using the camera parameters $\boldsymbol{s}$ and $\boldsymbol{e}$ by least-mean-square minimization. With the calibration results, each feature is represented in the 3-D world coordinates from multiple viewpoint images.

reconstruction technique described in [14]. Fig. 10 shows the 3-D reconstruction results from merging three views. Fig. 10(a) shows the results with respect to lens distortion, and Fig. 10(b) shows the results without taking lens distortion into account (using the pinhole camera model). As we can see from Fig. 10, modeling the lens distortion leads to better accuracy for 3-D reconstruction. We also evaluate the relationship between the number of viewpoints and the 3-D reconstruction error. Fig. 11 shows the result. We conclude that five views are sufficient for modeling 3-D reconstruction of the object.

VI. Applications of the Camera and Robot Calibration

We utilized our hand–eye calibration results for 3-D object modeling and robotic bin picking. As we have discussed in the previous sections, we have a good tool to move the camera to an arbitrary viewpoint in the world coordinate frame where the object to be modeled or to be grasped is located. In the following, we will present the experimental results for: 1) modeling 3-D objects and 2) recognizing and grasping such objects by controlling the viewpoints used by the camera.

A. Object Model Acquisition

In our laboratory, vision models of 3-D objects were generated using the human-assisted model-acquisition system [17], [18] along with the results from hand–eye calibration. In this robot teaching system, shown in Fig. 12, the objects were placed in the work area by a human, who then taught the system to establish the registration of image-to-image and pose-to-pose

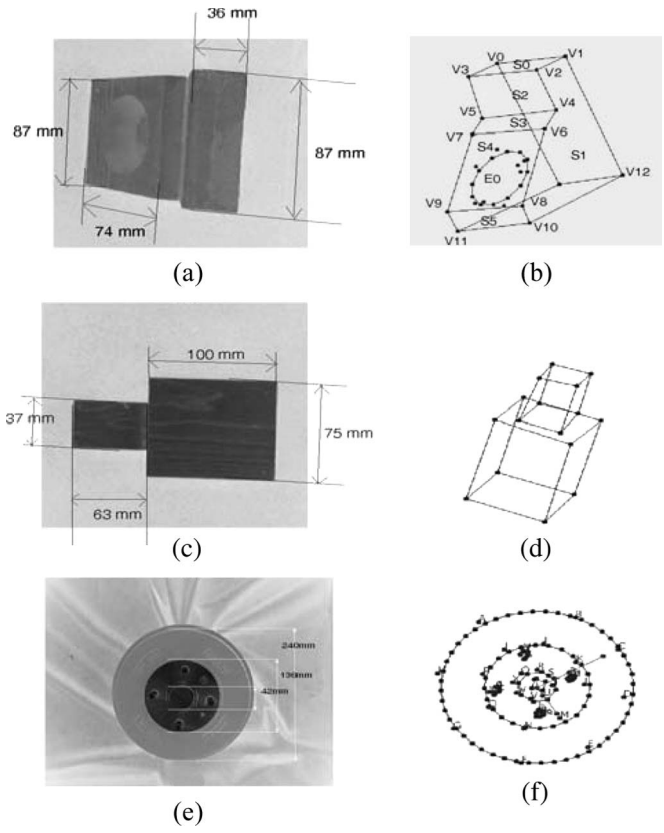


Fig. 13. Some results of acquiring object models. For the polyhedral objects (a), (b) and (c), (d), and the oval objects (e), (f). The geometrical shape was reconstructed using the calibration parameters.

correspondences. We generated a 3-D geometrical model of a 3-D object from salient features such as circles, polygons, ellipses, and free-form curves. The computer extracted 2-D salient features from the images taken from multiple views. The system then interacted with a human operator to represent the object with the extracted features for each 2-D view image. Then, the system estimated the 3-D coordinates of features in the world frame from the 2-D features in the image coordinates since the system registered the estimated 3-D features as a body of the 3-D object model. During the 3-D reconstruction process, the calibration results, such as the lens distortion parameters k_1 in (3), $\boldsymbol{s}$, and $\boldsymbol{e}$ and robotic kinematics, were required when the system integrated multiple views to generate 3-D object models, so that lines and curves in each image can be reliably and accurately reconstructed in 3-D features. Based on our evaluation experiment, five views were used for integration—this was verified by Fig. 11.

The example results of the two polyhedral object models are shown in Fig. 13(b) and (d). We tested our calibration method by modeling an artificially made object made of wood with polygon surfaces since the 3-D reconstruction for these objects easily evaluated the accuracy of our calibration results. The modeling error was less than 3% (around 3 mm) compared to the true values of measuring the actual objects [17], [18]. The industrial objects, including oval curve shapes, were also examined through a human computer interface editor (Fig. 12).

B. Object Localization and Grasping

We also show other experimental results that verify that our calibration method for viewpoint generation can be applied to 3-D object recognition and pose estimation tasks. After 3-D object models were generated with the human-assisted model acquisition system described in Section VI-A, the actual objects were randomly placed in a bin in the robot's workspace. The robot's task was to recognize and grasp the objects using the vision model. To verify and/or reject the pose hypotheses, a new viewpoint was computed by following the method in Section IV, so that the desired view could be obtained at that viewpoint. If the hypothesis was not fully verified, the system tried to capture an additional image from a subsequent viewpoint, and the object pose was recomputed to match the features obtained at the new viewpoints. We applied a model-based stereovision algorithm developed by Kosaka and Kak [14] to extract salient image features for object localization in the world coordinate frame. In our strategy, the robot initially took two views of the object whose viewpoints are preassigned. The robot then attempted to compute the pose hypotheses of the object based on the correspondence between the model features. To verify the applications of our calibration accuracy, we made experiments for object localization and grasping tasks [19], [20].

VII. Conclusion

This paper presented a new method of hand–eye robotic calibration for vision-based object modeling and grasping. Our method provides pixel-level image registration accuracy for 3-D reconstruction when arbitrary viewpoints are generated using a standard Puma or a Kawasaki robot. To attain this accuracy, we developed a hand–eye calibration process that included a lens distortion camera model. Our method integrates estimates of camera parameters obtained from multiple viewpoints by utilizing an extended Kalman filter. Based on modeling and manipulation experiments using the robots, we showed that our vision system had sufficient power for active viewpoint generation to acquire 3-D models of objects and to recognize and grasp these objects in the robotic work cell.

Acknowledgment

The authors would like to thank J. B. Zurn for her comments.

References

[1] J. Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 333–356, 1988.
[2] D. H. Ballard, "Animate vision," *Artif. Intell.*, vol. 48, no. 1, pp. 57–86, Feb. 1991.
[3] C. G. Broyden, "Quasi-Newton methods," in *Optimization Methods in Electronics and Communications*, vol. 1, *Mathematical Topics in Telecommunications*, K. W. Cattermole and J. J. O'Relly, Eds. New York: Wiley, 1984, pp. 105–110.
[4] T. Bucher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. von Seelen, "Image processing and behavior planning for intelligent vehicles," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 62–75, Feb. 2003.
[5] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. New York: Wiley, 1996, pp. 147–165.

[6] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand–eye calibration," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 617–622, Aug. 1998.
[7] O. D. Faugeras and G. Toscani, "Calibration problem for stereo," in *Proc. Int. Conf. Pattern Recog.*, 1986, pp. 15–20.
[8] O. D. Faugeras and M. Herbert, "Representation, recognition, and locating of 3-D objects," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 27–52, 1986.
[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[10] K. S. Fu, R. C. Gonzales, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987, pp. 12–81.
[11] S. Ganapathy, "Decomposition of transformation matrices for robot vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1984, pp. 130–139.
[12] L. Grewe and A. C. Kak, "Interactive learning of a multi-attribute hash table classifier for fast object recognition," *Comput. Vis. Image Underst.*, vol. 61, no. 3, pp. 387–416, 1995.
[13] A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *Comput. Vis. Graph. Image Process.—Image Understanding*, vol. 56, no. 3, pp. 271–329, Nov. 1992.
[14] A. Kosaka and A. C. Kak, "Stereo vision for industrial applications," in *Handbook of Industrial Robotics*, S. Y. Nof, Ed. New York: Wiley, 1999, pp. 269–294.
[15] J. M. Lee, K. Son, M. C. Lee, J. W. Choi, S. H. Han, and M. H. Lee, "Localization of a mobile robot using the image of a moving object," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 612–619, Jun. 2003.
[16] J. Manigel and W. Leonhard, "Vehicle control by computer vision," *IEEE Trans. Ind. Electron.*, vol. 39, no. 3, pp. 181–188, Jun. 1992.
[17] Y. Motai and A. C. Kak, "An interactive framework for acquiring vision models of 3D objects from 2D images," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 566–578, Feb. 2004.
[18] Y. Motai, "Visual-based human–robotic interaction for extracting salient features of an industrial object for an automated assembly system," *Int. J. Comput. Ind.—Special Issue on Machine Vision*, vol. 56, no. 8/9, pp. 943–957, Dec. 2005.
[19] Y. Motai, "A new edge-grouping algorithm for multiple complex objects localization," in *Proc. Innovations Appl. Artif. Intell.*, 2004, vol. 3029, pp. 1194–1203.
[20] Y. Motai and A. Kosaka, "Concatenate feature extraction for robust 3D elliptic object localization," in *Proc. 19th ACM Symp. Appl. Comput.*, 2004, pp. 21–28.
[21] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.
[22] K. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Trans. Robot. Autom.*, vol. 11, no. 1, pp. 86–104, Feb. 1995.
[23] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.
[24] C. C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," *IEEE Trans. Robot. Autom.*, vol. 8, no. 2, pp. 161–175, Apr. 1992.
[25] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.
[26] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 666–673.
[27] *Camera Calibration Toolbox for Matlab*. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/
[28] G. Wei, K. Arbter, and G. Hirzinger, "Active self-calibration of robotic eyes and hand–eye relationships with model identification," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 158–166, Feb. 1998.
[29] F. Jatta, G. Legnani, and A. Visioli, "Friction compensation in hybrid force/velocity control of industrial manipulators," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 604–613, Apr. 2006.
[30] S. Katsura, Y. Matsumoto, and K. Ohnishi, "Analysis and experimental validation of force bandwidth for force control," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 922–928, Jun. 2006.
[31] H. Hu and P. Woo, "Fuzzy supervisory sliding-mode and neural-network control for robotic manipulators," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 929–940, Jun. 2006.
[32] Y. Matsumoto, S. Katsura, and K. Ohnishi, "Dexterous manipulation in constrained bilateral teleoperation using controlled supporting point," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1113–1121, Apr. 2007.
[33] G. Cheng and K. Peng, "Robust composite nonlinear feedback control with application to a servo positioning system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1132–1140, Apr. 2007.
[34] N. Kubota and K. Nishida, "Perceptual control based on prediction for natural communication of a partner robot," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 866–877, Apr. 2007.
[35] M. Chien and A. Huang, "Adaptive control for flexible-joint electrically driven robot with time-varying uncertainties," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1032–1038, Apr. 2007.
[36] R. Wai and P. Chen, "Robust neural-fuzzy-network control for robot manipulator including actuator dynamics," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1328–1349, Jun. 2006.
[37] K. Sim, K. Byun, and F. Harashima, "Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1362–1372, Jun. 2006.
[38] S. Katsura, J. Suzuki, and K. Ohnishi, "Pushing operation by flexible manipulator taking environmental information into account," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1688–1697, Oct. 2006.
[39] K. Khayati, P. Bigras, and L. Dessaint, "A multistage position/force control for constrained robotic systems with friction: Joint-space decomposition, linearization, and multiobjective observer/controller synthesis using LMI formalism," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1698–1712, Oct. 2006.
[40] T. N. Chang, B. Cheng, and P. Sriwilaijaroen, "Motion control firmware for high-speed robotic systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1713–1722, Oct. 2006.
[41] G. Kim and W. Chung, "Tripodal schematic control architecture for integration of multi-functional indoor service robots," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1723–1736, Oct. 2006.
[42] S. Katsura and K. Ohnishi, "A realization of haptic training system by multilateral control," *IEEE Trans. Ind. Electron.*, vol. 53, no. 6, pp. 1935–1942, Dec. 2006.
[43] H. Hu and P. Woo, "Fuzzy supervisory sliding-mode and neural-network control for robotic manipulators," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 929–940, Jun. 2006.

Yuichi Motai (M'01) received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991, the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2002.

He is currently an Assistant Professor of electrical and computer engineering with the University of Vermont, Burlington. His research interests include the broad area of sensory intelligence, particularly of medical imaging, computer vision, sensor-based robotics, and human–computer interaction.

Akio Kosaka (S'86–M'88) received the B.Eng. and M.Eng. degrees from the University of Tokyo, Tokyo, Japan, in 1981 and 1983, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1992.

He is currently an adjunct Associate Professor of electrical and computer engineering with Purdue University, as well as the chief Research Scientist with Future Creation Laboratory, Olympus Corporation, Shinjuku, Japan. He has conducted a wide variety of research projects, including computer-aided neurosurgical navigation systems, 6DOF haptic interfaces, wearable user interfaces, content-based image retrieval systems, and 3-D digital camera systems. His research interests include computer vision, 3-D medical imaging, distributed sensor networks, intelligent environments, and mobile robot navigation.