A Basic Study of Hand Eye Calibration using a Tablet Computer

Junya Sato
Department of Mechanical Engineering, Gifu University,
1-1, Yanagido, Gifu-shi, Gifu, 501-1193, Japan

Takayoshi Yamada
Department of Mechanical Engineering, Gifu University,
1-1, Yanagido, Gifu-shi, Gifu, 501-1193, Japan

Kazuaki Ito
Department of Mechanical Engineering, Gifu University,
1-1, Yanagido, Gifu-shi, Gifu, 501-1193, Japan

E-mail: jsato@gifu-u.ac.jp, yamat@gifu-u.ac.jp, kazu_it@gifu-u.ac.jp

Abstract

In this study, we describe a hand eye calibration method for calibrating an attached camera and a handmade end-effector to a robot by utilizing a tablet computer. By iterating the touching black dots displayed in the computer and optimizing robot parameters to minimize the touching error, the hand eye calibration is achieved without the end-effector information such as the dimensions and attached position. We finally achieved 1.2 mm touching error.

Keywords: hand eye calibration, differential evolution, evolutionary computation, tablet computer

1. Introduction

To pick an object using a robot and vision system, it is necessary to calibrate a camera and robot in advance. More specifically, homogeneous transformation matrices between the camera and robot base or hand should be estimated accurately via the calibration. This is known as hand eye calibration. The general approach is to use a calibration board such as a checkerboard. However, this approach does not consider information such as dimensions of an end-effector and its attached position to the robot. Therefore, manual measurements of them are required, especially for a handmade end-effector. However, because the manual measurements contain errors, adjustment of the homogeneous transformation matrices is necessary. This is a tedious work. To address this problem, this study describes a method to achieve hand eye calibration without end-effector information by utilizing a tablet computer.

2. Related works

Cao *et al*. proposed a method that uses a neural network[1]. There is a direct approach that uses a laser tracker to compensate for the absolute position errors of a robot. However, such measurement devices are expensive. For this reason, Cao *et al*. proposed an indirect approach that does not use any device, but instead use the neural network to compensate the error. Mišeikis *et al*. achieved a wide range of accurate calibration by attaching a checkerboard to a robot and captured images of it using three RGB-D cameras[2]. However, this requires space to install the three cameras, which incurs high cost.

These previous works use a calibration board with a specific pattern, such as a checkerboard. Hence, this kind of calibration is a general approach. In contrast, Lee *et al*. proposed a method that does not use any markers, such as checkerboards[3]. First, the robot is captured by an RGB camera placed next to the robot, and each joint position of the robot is recognized using deep learning. The calibration is achieved by using joint positions, forward kinematics, and camera intrinsic. Nevertheless, some space is required for the camera to be installed to capture the robot.

As described, most previous works do not consider an end-effector such as its dimensions and attached position to a robot. Thus, if a handmade end-effector is used, additional calibration is required for the accurate manipulation. In order to avoid it, a novel hand eye calibration method using a tablet computer is proposed in this study.

3. Proposed method

The overview and appearance of the proposed method are shown in Figs. 1, 2 and 3, respectively. A tablet computer was used for the proposed method. The robot is installed with an RGB-D camera and a pen of the tablet as the end-effector. First, the screen of the tablet is captured by the

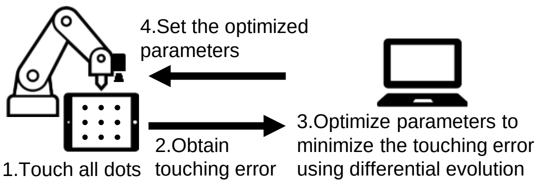

Fig. 1. Overview of the proposed method.

Fig. 2. Appearance of the proposed system.

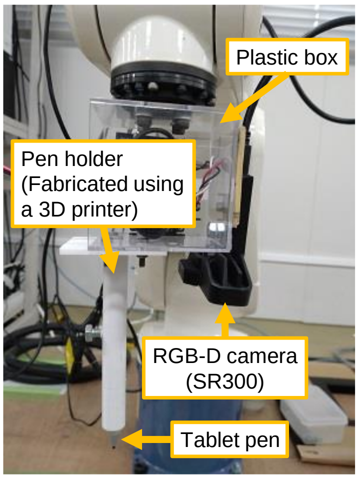

Fig. 3. Details of the end-effector in our method.

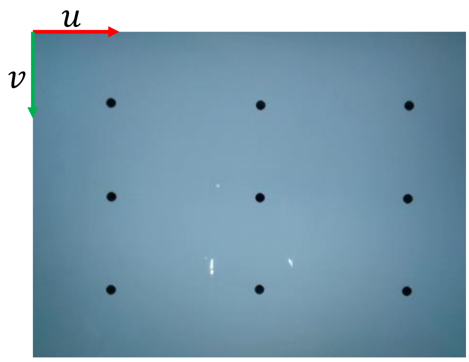

Fig. 4. Captured image by SR300, which is an RGB-D camera (640×480 pixels). The u and v axes represent image coordinate system, which is used in OpenCV for image processing.

camera (Fig. 4), and the coordinates of the nine black dots are obtained by image processing. Because their coordinate system is the image coordinate system, the obtained coordinates are transformed to the robot's base coordinate system using homogeneous transformation matrices. Second, the robot touches the dots with the attached pen. Third, the distance errors between the original and touched coordinates are obtained. Fourth, the average of the errors is calculated, and the parameters for homogeneous transformation matrices are optimized by differential evolution (DE) to minimize this error. Finally, the optimized parameters are set to the robot, and the same process is repeated. Using this iteration, the hand eye calibration is achieved. Because this approach does not require any operation by human while calibrating, smaller effort requires than existing approaches.

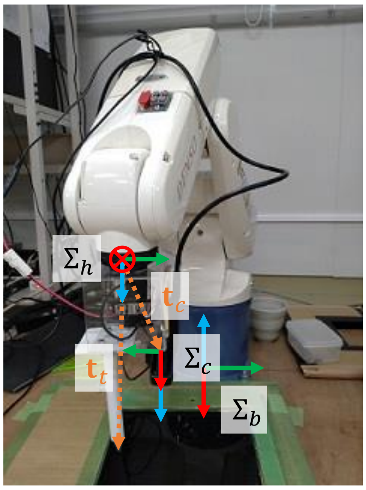

Fig. 5. Each coordinate system. The red, green, and blue arrows indicate *x*, *y*, and *z* axes, respectively.

3.1. *Coordinate system and optimized parameters*

Figure 5 shows each coordinate system in our method. The robot base coordinate, hand coordinate system, and camera coordinate system are denoted as Σ_b, Σ_h, and Σ_c, respectively. For the transformation from Σ_b to Σ_h, rotation of -180° in y axis and translation of $(x_h, y_h, z_h)^T$ are required. Similarly, for the transformation from Σ_h to Σ_c, rotations of 180° in y axis and 180° in x axis, and translation of $\mathbf{t}_c = (x_c, y_c, z_c)^T$ are necessary. An equation to transform a position of ith black dot in Σ_c $((x_i^c, y_i^c, z_i^c)^T)$, which is measured by the camera, to the robot base coordinate system $\left((x_i^b, y_i^b, z_i^b)^T\right)$ is:

$$\begin{bmatrix} x_i^b \\ y_i^b \\ z_i^b \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & x_h \\ 0 & 1 & 0 & y_h \\ 0 & 0 & -1 & z_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & x_c \\ 0 & -1 & 0 & y_c \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i^c \\ y_i^c \\ z_i^c \\ 1 \end{bmatrix} \quad (1)$$

Because $(x_i^c, y_i^c, z_i^c)^T$ cannot be acquired directly, they are obtained from image coordinate system using the following equation, which is based on the pinhole camera model.

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i^c / z_i^c \\ y_i^c / z_i^c \\ 1 \end{bmatrix} \quad (2)$$

Where u_i and v_i are ith position of the black dot in the camera coordinate system. The f_x and f_y are focal length of x and y axes, respectively. The c_x and c_y are coordinate of the principal point of x and y axes, respectively. From the above equation, the positions of each black dot in the image coordinate system (x_i^c and y_i^c) can be transformed to the camera coordinate system.

Because the camera and pen are not aligned to the rotation axis of the robot hand, offset $((x', y', z')^T)$ is necessary to touch with the attached pen. It can be calculated as below.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} \quad (3)$$

Where $\mathbf{t}_t = (x_t, y_t, z_t)^T$ is a translation vector from Σ_h to the tip of the tablet pen. The θ is a rotation angle of z axis in Σ_h. By combining Eq. (1) and (3), the position in the robot base coordinate system to touch each black dot using the tablet pen can be acquired.

$$\begin{bmatrix} x_i^{b'} \\ y_i^{b'} \\ z_i^{b'} \end{bmatrix} = \begin{bmatrix} x_i^b \\ y_i^b \\ z_i^b \end{bmatrix} - \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4)$$

Next, the parameters to be optimized are described here. The $(x_h, y_h, z_h)^T$ in Eq. (1) represents the initial robot hand position to capture the tablet display, and this is known. The $\mathbf{t}_c = (x_c, y_c, z_c)^T$ can be roughly obtained by manual measurement. However, this is not precise. Thus, the optimization is required. Similarly, $\mathbf{t}_t = (x_t, y_t, z_t)^T$ in Eq. (3) should be optimized. The f_x, f_y, c_x, and c_y can be obtained through a manual of an RGB-D camera or SDK (Software Development Kit). Hence, the optimization for them is not required. The z_i^c represents the distance between the tablet display and the camera at the initial robot hand position. This can be known from the depth information of the RGB-D camera. The both of z_c and z' are unknown. However, they can be ignored because the robot is controlled that the attached pen always touches the tablet display to obtain the touching error. Therefore, x_c, y_c, x_t, and y_t are the parameters to be optimized.

3.2. *Optimization by differential evolution (DE)*

In this study, differential evolution (DE), which is one of the population-based evolutionary computation methods, is adopted for the optimization because this can optimize all parameters simultaneously and is easy to use for us. The fitness function is:

$$\text{Fitness} = \sum_{i=0}^{8} \sqrt{\left(u_i - u_i^{\text{tch}}\right)^2 + \left(v_i - v_i^{\text{tch}}\right)^2} \Big/ 9 \quad (5)$$

Where $\left(u_i^{\text{tch}}, v_i^{\text{tch}}\right)$ is ith touched position by the robot. Because the parameters that minimize the touching error are required, this fitness function is used.

4. Experimental Setup

In our experiments, a 6-axis robot (DENSO VP-6242), tablet computer (Microsoft Surface Pro 7), tablet pen (Surface Pen), and an RGB-D camera (Intel SR300) were used. The set values related to Eq. (1) to (4) were $(x_h, y_h, z_h) = (320, -70, 290)$, $z_i^c = 168$, $f_x = 617.732788$, $f_y = 617.732849$, $c_x = 316.517365$, and $c_y = 242.328247$. The population and generation sizes for DE were 10 and 30, respectively. The scaling factor and crossover probability were 0.5 and 0.9, respectively. In experiments, DE/rand/1 and DE/best/1 strategies were compared. The used crossover method was binomial crossover. The set search ranges for each optimized parameter were $x_c \in [-20, 20]$, $y_c \in [20, 50]$, $x_t \in [-20, 50]$, and $y_t \in [-40, -10]$. In order to check the effectiveness of DE, random sampling (RS) was also introduced and compared.

The rotation angle of the sixth axis when the black dots were touched was gradually rotated so that the first dot was 0 degrees and the ninth was 180 degrees. In other words, the six axis was rotated by 22.5 degrees to investigate the robot can touch the dots correctly even though the hand rotates.

5. Result and Consideration

Table 1 represents the experimental results. From the S1 to S5 mean use of different random seeds because they affect the optimization results by DE. Since the results of DE/rand/1 and DE/best/1 were better than RS, the effectiveness of DE was indicated. The average mean touching error of DE/rand/1 was smaller than DE/best/1. This is because the DE/rand/1 could keep higher diversity of the population.

6. Conclusion

In this study, we proposed a hand eye calibration method using a tablet computer. Because existing previous studies do not consider dimensions of the end-effector and attached position to the robot, additional calibration for it could be required if a handmade end-effector is used. On the other hand, since our approach calibrates based on the touching error, which is obtained from the tablet computer, the hand eye calibration is achieved without the information of the end-effector. In the experiment, our method achieved minimum touching error of 1.2 mm and mean touching error of 1.7 mm.

Table 1. Mean touching error in [mm].

Method	S1	S2	S3	S4	S5	Ave.
DE/rand/1	1.5	1.2	1.3	3.3	1.2	1.7
DE/best/1	3.8	1.3	1.8	1.9	2.6	2.3
RS	4.7	4.8	4.9	3.2	2.7	4.1

References

1. C. T. Cao, V. P. Do, B. R. Lee, "A novel indirect calibration approach for robot positioning error compensation based on neural network and hand-eye vision", applied sciences, Vol. 9, No. 9, pp. 1-17, 2019
2. J. Mišeikis, K. Glette, O. J. Elle, J. Torresen, "Automatic calibration of a robot manipulator and multi 3d camera", proceedings of IEEE/SICE international symposium on system integration, pp. 2474-2325, 2016
3. T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, S. Birchfield, "Camera-to-robot pose estimation from a single image", proceedings of IEEE international conference on robotics and automation, pp. 9426-9432, 2020

Authors Introduction

Dr. Junya Sato

He received the B.E., M.E., and Ph.D. degrees in Engineering from Iwate University, Japan, in 2012, 2014, and 2017, respectively. Since 2017, he has been an Assistant Professor with the Department of Mechanical Engineering, Faculty of Engineering, Gifu University, Japan.

Dr. Takayoshi Yamada

He received the B.E., M.E., and Ph.D. degrees in mechanical engineering from the Nagoya Institute of Technology, Japan, in 1991, 1993, and 1995, respectively. He is currently a Professor with the Department of Mechanical Engineering, Gifu University, Japan.

Dr. Kazuaki Ito

He received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the Nagoya Institute of Technology, Japan, in 1998, 2000, and 2003, respectively. He moved to the Department of Mechanical Engineering, Gifu University, Japan and has been a Professor since 2022.