Affine registration of three-dimensional point sets for improving the accuracy of eye position trackers

Donghoon Kang,* **Jinwook Kim, and Sung-Kyu Kim**
Korea Institute of Science and Technology (KIST), Imaging Media Research Center, Seoul, Republic of Korea

Abstract. Existing methods for tracking three-dimensional (3-D) eye positions with a monocular color camera mostly rely on a generic 3-D face model and a certain face database. However, the performance of these methods is susceptible to the variations of head poses. For this reason, existing methods for estimating 3-D eye position from a single two-dimensional face image may yield erroneous results. To improve the accuracy of 3-D eye position trackers using a monocular camera, we present a compensation method as a postprocessing technique. We address the problem of determining an optimal registration function for fitting 3-D data consisting of the inaccurate estimates from the eye position tracker and their corresponding ground truths. To obtain the ground truths of 3-D eye positions, we propose two different systems by combining an optical motion capture system and checkerboards, which construct the form of the hand-eye and robot-world calibration. By solving a least-squares optimization problem, we can determine the optimal registration function in an affine form. Real experiments demonstrate that the proposed method can considerably improve the accuracy of 3-D eye position trackers using a single color camera. [DOI: 10.1117/1.OE.56.4.043105]

Keywords: monocular camera; eye positions; hand-eye and robot-world calibration; affine registration.

Paper 161474 received Sep. 21, 2016; accepted for publication Apr. 5, 2017; published online Apr. 20, 2017.

1 Introduction

Autostereoscopic display systems can provide three-dimensional (3-D) images for viewers without requiring any special glasses.[1–3] In such systems, the information about viewers' 3-D eye positions relative to the display screen can provide a useful clue to reduce crosstalk and extend viewing zones.[4–6] In this regard, the information of viewers' 3-D eye positions can be an important ingredient for realistic autostereoscopic visualization.

In theory, 3-D eye positions can be easily obtained by using stereo triangulation if two calibrated cameras are available and a pair of salient two-dimensional (2-D) feature points corresponding to the eye can be correctly detected by some methods.[7,8] From epipolar geometry, the physical length of the baseline–the line connecting two camera centers–can give useful information about the absolute scale of the observed environment. For this reason, a dedicated camera system consisting of a stereo high-resolution camera and some active infrared (IR) illumination devices is commonly used to track 3-D positions of pupils and estimate gaze directions.[8,9] However, when considering that eye tracking sensors–typically cameras–are usually attached to display devices, the camera systems under active IR illumination are not suitable for some autostereoscopic display systems with large viewing distances between the display screen and viewers. This is because the working volume of the camera systems with active IR illumination devices for tracking small-sized pupils are mostly small. To resolve the issue of the small working volume inherent in these types of systems employing the camera and active IR illumination devices, some researchers use several additional devices such as a zoom-lens camera[10] or a pan-tilt unit[11] in order to track remote pupils.

Rather than directly tracking small-sized pupils by using high-resolution cameras with active IR illumination devices, there exist other methods for estimating 3-D eye positions by tracking boundary feature points of the eyes including eye corners with a monocular low-resolution color camera. However, when using a single low-resolution color camera, 2-D feature points around the eyes can be erroneously detected by feature extraction algorithms owing to illumination changes, cluttered background images, or nonfrontal head poses relative to the cameras. Moreover, without reasonable assumptions and constraints, it is an ill-posed problem to obtain 3-D eye positions on the absolute scale by using only a monocular color camera.[12] Common approaches for resolving the issue of inferring the absolute scale of 3-D facial features by using a single 2-D facial image are data-driven, learning-based, and dependent on deformable 3-D face models.[13–15] In these approaches, the accuracy of the estimated 3-D eye positions from a single 2-D face image is susceptible to the variations of head poses related to the employed 3-D face models and face databases.[16] Although there are numerous methods[16,17] for constructing sophisticated 3-D face models using a single camera, there exists almost no face database consisting of monocular 2-D face images annotated by the metric information about the corresponding ground truths of 3-D eye positions. For this reason, existing methods for estimating 3-D eye positions on the absolute scale from monocular 2-D images may yield erroneous results. To improve the accuracy of eye trackers using a single color camera, some compensation methods are required.

*Address all correspondence to: Donghoon Kang, E-mail: kimbab.moowoo@gmail.com

0091-3286/2017/$25.00 © 2017 SPIE

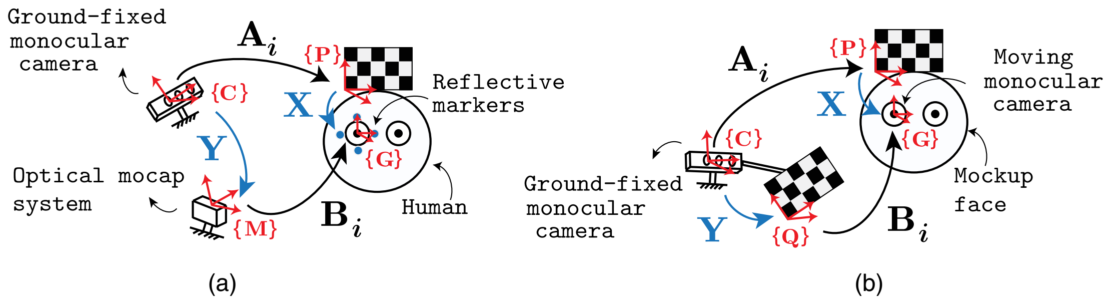

Fig. 1 Two different systems to obtain the ground truths of 3-D eye positions. (a) Reflective markers around a human eye and (b) mockup face with a checkerboard.

In this paper, we propose a systematic compensation method for improving the accuracy of 3-D eye position trackers using a "monocular" low-resolution color camera by determining an optimal registration function for data fitting. To our knowledge, the problem of improving the accuracy of eye trackers using a monocular camera as a postprocessing technique is first addressed in this paper. The problem of improving the accuracy of an eye position tracker can be viewed as the problem of fitting two sets of 3-D point data consisting of the inaccurate 3-D eye position estimates from the eye tracker with a single camera and the ground truths of the corresponding 3-D eye positions. To perform 3-D data fitting, we first need to find the ground truths of 3-D eye positions and then determine a registration function that maps the 3-D eye position estimate onto the ground truth of the corresponding 3-D eye position.

To obtain the ground truths of 3-D eye positions with a single color camera, we propose two different types of systems as shown in Figs. 1(a) and 1(b), which combine an optical motion capture (mocap) system and checkerboards. From these setups, we can construct the system as the form of the hand-eye and robot-world calibration[18–20] to obtain the ground truths of 3-D eye positions. As shown in Fig. 1(a), several reflective markers around an eye are attached and a hat can be rigidly combined with a checkerboard on the head. Since the optical mocap system with multiple infrared cameras is generally expensive, we devise a relatively inexpensive apparatus as an alternative device to obtain the ground truths of eye positions by using an additional camera and checkerboards [see Fig. 1(b)].

After obtaining the ground truths of 3-D eye positions by composing the systems as in Fig. 1 and using the algorithm for the hand-eye and robot-world calibration, the next task is to find a registration function, which fits the inaccurate 3-D eye position estimates into the ground truth of 3-D eye positions. To confine the solution space and reduce overfitting side-effects, we assume that the registration function for fitting 3-D data has the form of an affine function, which can be derived from the first-order approximation. By solving a least-squares optimization problem, we can determine the optimal affine function in an analytic form. Determination of the registration function can be regarded as the offline process for parameter estimation. With the registration function in an affine form, the 3-D eye positions estimated by the eye tracker using a single camera can be compensated in real time. To demonstrate the generality and effectiveness of the proposed method regarding the approximation of the registration function in an affine form, we perform extensive experiments by setting up a system as illustrated in Fig. 1(b). Furthermore, we show that the crosstalk can be reduced when applying the proposed method to actual autostereoscopic display systems.

The remainder of this paper is organized as follows: after introducing necessary notations in Sec. 2, we explicitly describe the problem in Sec. 3. In Sec. 4, we first show that the ground truths of 3-D eye positions can be obtained by setting up two different systems as in Figs. 1(a) and 1(b). Then we describe how to approximate the nonlinear registration function as the form of an affine function for fitting two sets of 3-D points. Experimental results using real data sets are presented in Sec. 5 to validate the proposed method.

2 Mathematical Preliminaries and Notations

Before beginning to describe the problem and present our solution, let us first provide mathematical preliminaries[21] and necessary notations. In mathematics, a Lie group is a finite-dimensional group that is a differentiable manifold where the product and inverse group operations are smooth. The group of rigid-body motions in $\mathbb{R}^3$ denoted by SE(3) is an example of a matrix Lie group and can be represented by 4×4 real matrices of the form $\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$, where $\mathbf{R} \in \mathrm{SO}(3)$, $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{0}$ denotes a row vector of three zeros. Here, SO(3) denotes the group of 3×3 rotation matrices, which is also an example of a matrix Lie group. Throughout this paper, $\mathbf{R}$ and $\mathbf{p}$ denote a 3×3 rotation matrix and a 3×1 translation (i.e., position) vector, respectively. For instance, the rotation matrix and translation vector consisting of $\mathbf{Z} \in \mathrm{SE}(3)$ can be, respectively, denoted by $\mathbf{R}_Z \in \mathrm{SO}(3)$ and $\mathbf{p}_\mathbf{Z} \in \mathbb{R}^3$.

The *Lie algebra* associated with the matrix Lie group $\mathcal{M}$ is defined as the tangent space at the identity element of $\mathcal{M}$. On SO(3), the associated Lie algebra so(3) is the set of 3×3 skew-symmetric matrices of the form $[\mathbf{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$, where $\mathbf{r} = (r_1, r_2, r_3)^\mathrm{T} \in \mathbb{R}^3$.

The fundamental concept related to matrix Lie groups is the *exponential mapping*. Given a matrix Lie group $\mathcal{M}$ and its associated Lie algebra $\mathfrak{m}$, the exponential mapping is the map $\exp : \mathfrak{m} \rightarrow \mathcal{M}$ defined by the matrix exponential: $\exp \Psi = \mathbf{I} + \Psi + \frac{1}{2!}\Psi^2 + \cdots$ for any $\Psi \in \mathfrak{m}$. Here, $\mathbf{I}$ denotes the identity matrix, of which size is clear from the context. The exponential mapping from so(3) to SO(3) is given by the following explicit equation:

$\exp[\mathbf{r}] = \mathbf{I} + a[\mathbf{r}] + b[\mathbf{r}]^2 \in \mathrm{SO}(3)$, where $a = (\sin \|\mathbf{r}\|)/\|\mathbf{r}\|$ and $b = (1 - \cos \|\mathbf{r}\|)/\|\mathbf{r}\|^2$.

The inverse of the exponential map, or matrix logarithm of SO(3) can also be expressed by the following equation: suppose $\mathbf{R} \in \mathrm{SO}(3)$ such that $\mathrm{tr}(\mathbf{R}) \neq -1$, where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Then

$$\log \mathbf{R} = \frac{\theta}{2 \sin \theta}(\mathbf{R} - \mathbf{R}^{\mathrm{T}}), \tag{1}$$

where θ satisfies $1 + 2 \cos \theta = \mathrm{tr}(\theta)$, $|\theta| < \pi$, and $\|\log \mathbf{R}\| = \theta$. In the event that $\mathrm{tr}(\mathbf{R}) = -1$, the logarithm $[\mathbf{r}] = \log \mathbf{R}$ has two antipodal solutions $\pm\mathbf{r}$ which can be determined from the relation $\mathbf{R} = I + (2/\pi^2)[\mathbf{r}]^2$. If $\log \mathbf{R}$ is written as $\log \mathbf{R} = [\omega]\theta$, where $\omega \in \mathbb{R}^3$ and $\|\omega\| = 1$, then it has the following physical meaning: the unit vector ω represents the axis of a rotation and θ is the angle of the rotation.

3 Problem Statement

Let us first assume that a set of 3-D facial features including eye corner features can be obtained at every time step by using a certain 3-D face tracking algorithm with a monocular color camera. As shown in Fig. 2(a), $\{C\}$ and $\{E\}$ denote the co-ordinate frames fixed to the monocular color camera and the eye, respectively. From 3-D facial features at time step i, we can easily extract the 3-D eye position $\xi_i \in \mathbb{R}^3$, which represents the vector from the origin of $\{C\}$ to the origin of $\{E\}$ expressed in $\{C\}$. When ξ_i is not accurate enough, we need to adjust ξ_i to the correct value as a postprocessing procedure.

A reasonable approach for correcting ξ_i may involve finding a certain registration function $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that maps the inaccurate ξ_i into a more accurate value called the ground truth. To approximate f, we should first find the unknown ground truth of the 3-D eye position. Let $\mathbf{g}_i \in \mathbb{R}^3$ denote the unknown ground truth of the 3-D eye position corresponding to ξ_i.

To obtain $\mathbf{g}_i$, one may consider some additional devices like an optical mocap system or a checkerboard [see Figs. 2(b) and 2(c)]. However, in these cases, an important issue related to the unknown co-ordinate transformations may arise. To be more specific, let us first consider an optical mocap system consisting of multiple networked optical IR cameras and reflective markers, which is commonly believed to provide highly accurate poses of a set of markers relative to the optical camera system. As shown in Fig. 2(b), several reflective markers are assumed to be put around the eye. In this situation, the optical mocap system can provide the pose of $\{E\}$ denoted by $\mathbf{B}_i = \begin{bmatrix} \mathbf{R}_{B_i} & \mathbf{p}_{B_i} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3)$ relative to the co-ordinate frame of the optical mocap system $\{M\}$. Let $\mathbf{Y} \in \mathrm{SE}(3)$ denote the unknown rigid body transformation of the frame $\{C\}$ relative to the frame $\{M\}$ and $\mathbf{e}_4 = (0,0,0,1)^{\mathrm{T}}$. The representation of $\mathbf{g}$ in homogeneous co-ordinates can be denoted as $\breve{\mathbf{g}}_i = (\mathbf{g}_i^{\mathrm{T}}, 1)^{\mathrm{T}} \in \mathbb{R}^4$ by appending a 1 to $\mathbf{g}$. If $\mathbf{Y}$ can be determined by some methods in advance, we can calculate $\breve{\mathbf{g}}_i$ from

$$\breve{\mathbf{g}}_i = \mathbf{Y}\mathbf{B}_i\mathbf{e}_4, \tag{2}$$

where the vector $\mathbf{Y}\mathbf{B}_i\mathbf{e}_4$ is the translational part of the pose $\mathbf{Y}\mathbf{B}_i$.

Instead of using additional precise devices like an optical mocap system in Fig. 2(b), one may consider attaching a checkerboard on a head as shown in Fig. 2(c). In a similar fashion to the case in Fig. 2(b), we can obtain $\breve{\mathbf{g}}_i$ in Fig. 2(c) as

$$\breve{\mathbf{g}}_i = \mathbf{A}_i\mathbf{X}\mathbf{e}_4. \tag{3}$$

4 Method

In this section, we first present a method for obtaining the ground truth $\mathbf{g}_i$ of 3-D eye position in Fig. 2(a) by determining the unknown constant pose $\mathbf{Y}$ (or $\mathbf{X}$) as shown in Figs. 2(b) or 2(c). Then, from given pairs of 3-D points $(\xi_i, \mathbf{g}_i)$, $(i = 1, \ldots, N)$, where N denotes the number of pose measurements, we will approximate the registration function f to the first order by solving a least-square optimization problem.

4.1 *Ground Truths of Eye Positions*

To determine $\mathbf{Y}$ (or $\mathbf{X}$) in [Figs. 2(b) or 2(c)], we set up a system as depicted in [Figs. 1(a) or 1(b)] by adding additional apparatuses to the original system. Let us first consider the system in Fig. 2(b) that uses a precise measurement device like an optical mocap system to obtain an accurate eye position. To obtain $\mathbf{g}_i$ from Eq. (2), we should determine $\mathbf{Y}$ beforehand. The idea of attaching a checkerboard rigidly to the head as shown in Fig. 1(a) can provide a useful condition, which can give a clue to determine $\mathbf{Y}$ by collecting additional pose measurement $\mathbf{A}_i$. Here, $\mathbf{A}_i$ shown in Fig. 1(a) represents the pose of the checkerboard frame $\{P\}$ relative to the camera frame $\{C\}$ at time step i and it can be easily obtained by using a standard camera calibration method.[22]

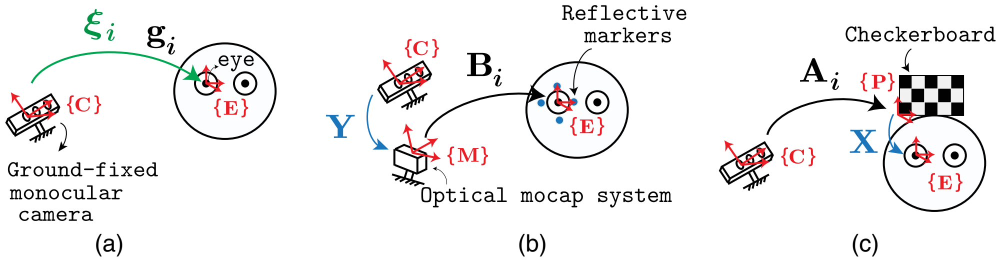

Fig. 2 Unknown co-ordinate transformations $\mathbf{X}$, $\mathbf{Y} \in \mathrm{SE}(3)$ that are constant. (a) Unknown ground truth $\mathbf{g}_i$, (b) optical mocap system, and (c) checkerboard pattern.

The pose of the eye co-ordinate frame $\{E\}$ defined by the reflective markers in Fig. 1(a) relative to the camera frame $\{C\}$ can expressed as $\mathbf{A}_i\mathbf{X}$ or equally $\mathbf{YB}_i$. Note that $\mathbf{A}_i\mathbf{X}$ represents the equivalent rigid body transformation to $\mathbf{YB}_i$.

Let us now consider another system as shown in Fig. 2(c), where there is unknown rigid body transformation $\mathbf{X}$ between the checkerboard frame $\{P\}$ and the eye co-ordinate frame $\{E\}$. To obtain $\mathbf{g}_i$ from Eq. (3), $\mathbf{X}$ should be identified in advance. To determine $\mathbf{X}$ in Fig. 2(c), we can construct the system as depicted in Fig. 1(b). In contrast to the system illustrated in Fig. 1(a), where a person is the subject of the experiments, the system in Fig. 1(b) employs a mockup face, of which an eye is replaced by a webcam. As depicted in Fig. 1(b), two distinct checkerboards are rigidly attached to a camera on the ground and a mockup face, respectively. Using a standard camera calibration method, we can obtain pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$, $i = 1, \ldots, N$.

Given noisy pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$, the problem of determining $\mathbf{X}$ and $\mathbf{Y}$ in Fig. 1(a) can be formulated as minimizing the cost function J as

$$J(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{X} - \mathbf{YB}_i\|^2, \tag{4}$$

where several choices of matrix norm $\| \bullet \|_2$ are available. We define $\| \bullet \|_2$ as $\|\mathbf{P}-\mathbf{Q}\|^2 = \|\mathbf{R}_P - \mathbf{R}_Q\|_F^2 + \zeta\|\mathbf{p}_P - \mathbf{p}_Q\|^2$ where $\mathbf{R}_P$, $\mathbf{R}_Q \in \mathrm{SO}(3)$ and $\mathbf{p}_P$, $\mathbf{p}_Q \in \mathbb{R}^3$ represent the rotations and the translations of $\mathbf{P} = \begin{bmatrix} \mathbf{R}_P & \mathbf{p}_P \\ \mathbf{0} & 1 \end{bmatrix}$, $\mathbf{Q} = \begin{bmatrix} \mathbf{R}_Q & \mathbf{p}_Q \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3)$, respectively. Here, $\| \bullet \|_F$ denotes the Frobenius norm and the positive scalar $\zeta \in \mathbb{R}$ is a weighting factor for the translation. As a mathematical tool for solving $\mathbf{X}$ and $\mathbf{Y}$ that minimizes the cost function J in Eq. (4), we can employ a recent geometric optimization algorithm for the hand-eye and robot-world calibration,[20] which is summarized in Appendix A.

By using pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$ and the transformation $\mathbf{X}$ (or $\mathbf{Y}$), we can obtain the ground truth $\mathbf{g}_i$ of 3-D eye position in Fig. 1 from [Eq. (3) or Eq. (2)].

4.2 *Affine Registration for Fitting Two Sets of 3-D Points Data*

In the previous section, we have presented how to obtain $\mathbf{g}_i$ by constructing the system as shown in Fig. 1 and using the algorithm given in Appendix A. We now present a method for approximating a registration function f that maps ξ_i into $\mathbf{g}_i$. Given pairs of 3-D points $(\mathbf{g}_i, \xi_i)$, $(i = 1 \ldots N)$, the registration function $f\colon \mathbb{R}^3 \rightarrow \mathbb{R}^3$ can be written as $\mathbf{g}_i \approx f(\xi_i) = f(\mathbf{c}) + \nabla f(\mathbf{c})(\xi_i - \mathbf{c}) + \cdots$, where $\mathbf{c} \in \mathbb{R}^3$ is unknown. Here, $f(\xi_i)$ can be approximated by $f(\xi_i) \approx h(\xi_i) := f(\mathbf{c}) + \nabla f(\mathbf{c})(\xi_i - \mathbf{c})$. The function h is the first-order approximation of f at a point $\mathbf{c}$ and can be rewritten as an affine function form

$$h(\xi_i) = \mathbf{S}\xi_i + \mathbf{v}, \tag{5}$$

where $\mathbf{S} := \nabla f(\mathbf{c})$ is the 3×3 matrix and $\mathbf{v} := f(\mathbf{c}) - \nabla f(\mathbf{c})\mathbf{c}$ is the 3×1 column vector.

Given pairs of corresponding data $(\mathbf{g}_i, \xi_i)$, $(i = 1, \ldots, N)$, we can formulate the optimization problem by constructing the cost function J_0 as

$$\text{minimize } J_0(\mathbf{S}, \mathbf{v}) = \sum_{i=1}^{N} \|\mathbf{g}_i - (\mathbf{S}\xi_i + \mathbf{v})\|^2, \tag{6}$$

where $\mathbf{S}$ and $\mathbf{v}$ are the optimization variables. We now use the first-order necessary conditions to compute the optimal $\mathbf{S}$ and $\mathbf{v}$. From $\frac{\partial J_0(\mathbf{S},\mathbf{v})}{\partial \mathbf{v}} = \mathbf{0}$, we can obtain

$$\mathbf{v} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{g}_i - \mathbf{S}\xi_i). \tag{7}$$

Substituting Eqs. (7) into (6) yields

$$J_0(\mathbf{S}) = \sum_{i=1}^{N} \|\tilde{\mathbf{g}}_i - \mathbf{S}\tilde{\xi}_i\|^2, \tag{8}$$

where $\tilde{\xi}_i := \xi_i - \frac{1}{N}\sum_{i=1}^{N} \xi_i$ and $\tilde{\mathbf{g}}_i := \mathbf{g}_i - \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}_i$. From $\frac{\partial J_0(\mathbf{S})}{\partial \mathbf{S}} = \mathbf{0}$ in Eq. (8), we have

$$\mathbf{S} = \left(\sum_{i=1}^{N} \tilde{\mathbf{g}}_i\tilde{\xi}_i^{\mathrm{T}}\right)\left(\sum_{i=1}^{N} \tilde{\xi}_i\tilde{\xi}_i^{\mathrm{T}}\right)^{-1}. \tag{9}$$

Using Eq. (9), we can obtain the optimal $\mathbf{S}$. The optimal $\mathbf{v}$ can now be computed by substituting $\mathbf{S}$ into Eq. (7). Thus far, we have obtained the optimal parameters $\mathbf{S}$ and $\mathbf{v}$ in Eq. (5) offline. For real-time applications, we can correct ξ_i by using Eq. (5).

5 Experimental Results Using Real Data

Although our method considers a monocular color camera, it can be used to improve the accuracy of 3-D eye positions estimated by any type of 3-D face pose trackers using the red, green, blue and depth (RGB-D) camera system, of which extrinsic calibration parameters between an RGB camera and an IR camera are not sufficiently accurate. To verify this, a set of real data is obtained by using a 3-D face tracker (Intel Realsense SDK) with the RGB-D camera, of which extrinsic calibration parameters are just used in a factory-calibrated setting. Since the factory-calibrated extrinsic parameters of the RGB-D camera are usually inaccurate, obviously the accuracy of 3-D eye positions estimated by 3-D face tracker using this camera system without customization will be inaccurate. The other set of real data will be collected by commercial face tracking software called FaceAPI™ using a monocular color camera.

In our experiments, we first set up the system as shown in Fig. 1(b). A webcam (Logitech C600) is installed on the right eye of a mockup face and a 7 × 9 checkerboard is rigidly attached to this mockup, of which each checker square is of dimensions 40 mm × 40 mm. A 7 × 9 checkerboard pattern image is displayed on the screen of the display device (Baytech Yamakasi QH2711 Black Label DP; maximum display pixel resolution is 2560 × 1440). The size of each checker on the screen is $36 \times 36\ \mathrm{mm}^2$.

5.1 *First Experiment: 3-D Face Tracker Using a Factory-Calibrated RGB-D Camera*

As the first experiment, we mount the RGB-D camera (Intel RealSense™ F200) onto the display device. In this experiment, the extrinsic calibration parameters of the RGB-D

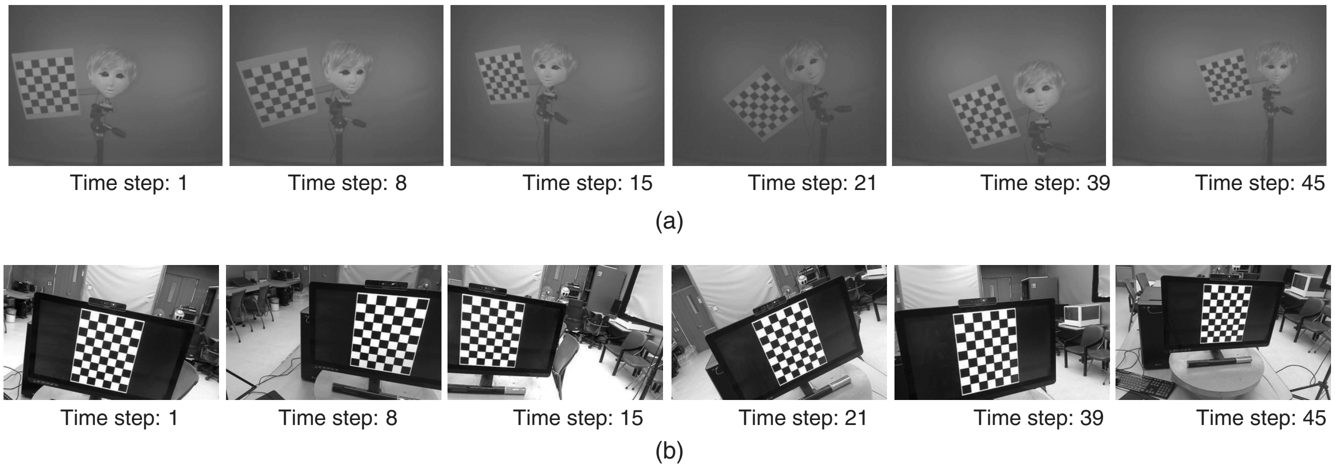

Fig. 3 Synchronized images captured by the webcam and RGB-D camera at time step $i = 1, 8, 15, 21, 39$, and 45. Images captured by the (a) infrared camera of the RGB-D camera on the display screen, which correspond to $\mathbf{A}_i$ and (b) the webcam on the mockup face, which correspond to $\mathbf{B}_i$.

camera are just used in a factory-calibrated setting. The Intel RealSense™ SDK can provide 3-D positions of facial feature points in metric units with respect to its IR camera. By averaging the positions of four symmetric corner feature points around the right eye, we can obtain the estimate ξ_i $(i = 1, \ldots, 50)$ of 3-D eye positions from the eye tracker that is provided by the Intel RealSense™ SDK.

While keeping a mockup face stationary, we can obtain the images of the checkerboard on the display screen by using the webcam on the right eye of the mockup face. At the same time, the RGB-D camera on the display device captures the image of the checkerboard attached to the mockup face (see Fig. 3). We repeat this procedure 50 times with different poses of the mockup face. Using a standard camera calibration method,[22] we can obtain pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$ as well as ξ_i, $(i = 1, \ldots, 50)$. Figure 4 shows the trajectory of $((\mathbf{A}_i, \mathbf{B}_i))$; $(i = 1, \ldots, 50))$.

Given the pose data $\mathbf{A}_i = \begin{bmatrix} \mathbf{R}_{A_i} & \mathbf{p}_{A_i} \\ \mathbf{0} & 1 \end{bmatrix}$, $\mathbf{B}_i = \begin{bmatrix} \mathbf{R}_{B_i} & \mathbf{p}_{B_i} \\ \mathbf{0} & 1 \end{bmatrix}$, $(i = 1, \ldots, 50)$, we can determine unknown constant poses $\mathbf{X} = \begin{bmatrix} \mathbf{R}_X & \mathbf{p}_X \\ \mathbf{0} & 1 \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} \mathbf{R}_Y & \mathbf{p}_Y \\ \mathbf{0} & 1 \end{bmatrix}$ by using the hand-eye and robot-world calibration algorithm,[20] which is briefly explained in Appendix A. Since the weighting factor ζ in Eq. (4) is a designer's tuning parameter, we can set $\zeta = 1$. In our experiments, the checkerboard frame $\{Q\}$ in Fig. 1(b) can be considered as the screen co-ordinate frame (see Fig. 3). In this respect, we remark that $\mathbf{Y}$ represents a rigid body transformation of the display screen co-ordinate frame $\{Q\}$ relative to the camera co-ordinate frame $\{C\}$.

Let us denote the rotation error $s_{\text{rot}}^{(i)}$ and the translation error $s_{\text{tran}}^{(i)}$ $(i = 1, \ldots, 50)$ of the hand-eye and robot-world calibration at time step i as

$$s_{\text{rot}}^{(i)} := \| \log[\mathbf{R}_{A_i} \mathbf{R}_X (\mathbf{R}_Y \mathbf{R}_{B_i})^{-1}] \|, \tag{10}$$

$$s_{\text{tran}}^{(i)} := \| \mathbf{R}_{A_i} \mathbf{p}_X + \mathbf{p}_{A_i} - \mathbf{R}_Y \mathbf{p}_{B_i} - \mathbf{p}_Y \|. \tag{11}$$

The explicit equation about the matrix logarithm $\log(\cdot)$ on SO(3) for computing $s_{\text{rot}}^{(i)}$ in Eq. (10) is given by Eq. (1).

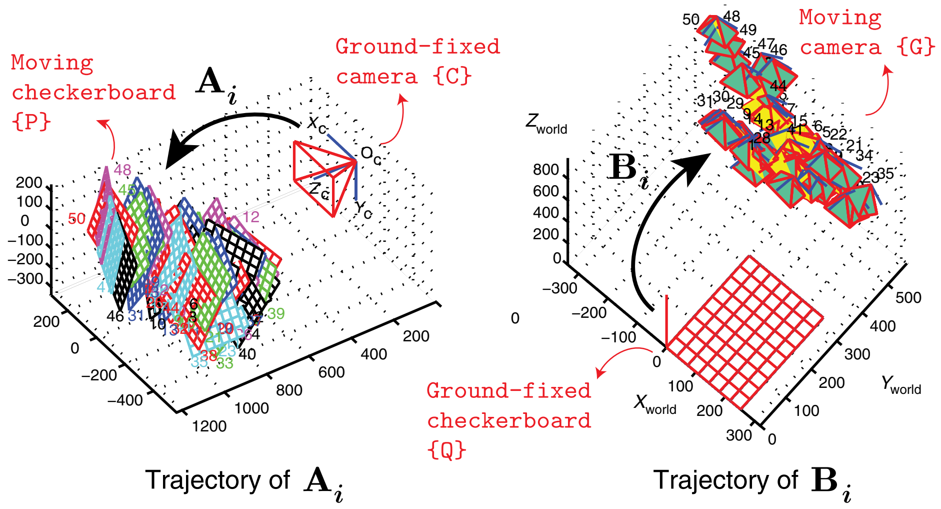

Fig. 4 Trajectories of pose data for hand-eye and robot-world calibration.

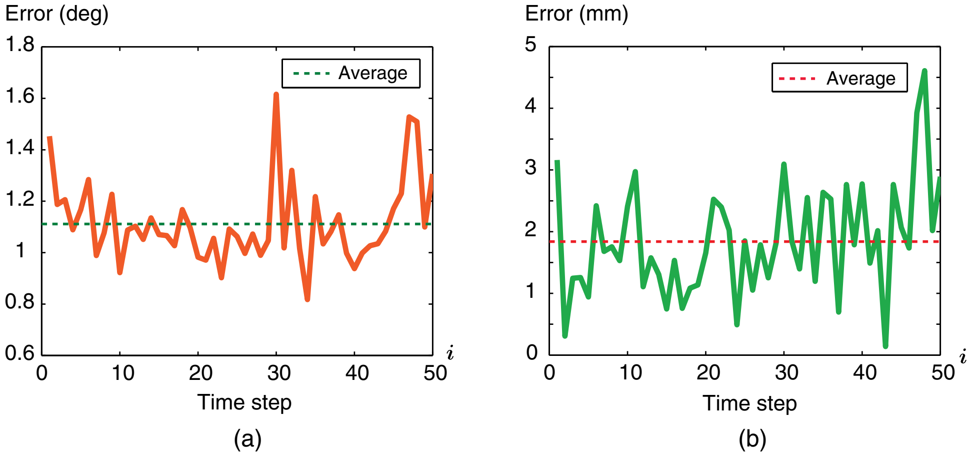

Fig. 5 The errors of the hand-eye and robot-world calibration. Errors (a) $S_{\mathrm{rot}}^{(i)}$ and (b) $S_{\mathrm{tran}}^{(i)}$.

Figure 5 shows the errors, $s_{\mathrm{rot}}^{(i)}$ and $s_{\mathrm{tran}}^{(i)}$, $(i = 1, \ldots, 50)$ representing the errors of the estimated rigid body transformations $(\mathbf{X}, \mathbf{Y})$ in terms of the rotational and translational parts. Let us define the averages of errors $s_{\mathrm{rot}}^{(i)}$ and $s_{\mathrm{tran}}^{(i)}$, $(i = 1, \ldots, 50)$ as follows: $\bar{s}_{\mathrm{rot}} := (1/50)\sum_{i=1}^{50} s_{\mathrm{rot}}^{(i)}$ and $\bar{s}_{\mathrm{tran}} := (1/50)\sum_{i=1}^{50} s_{\mathrm{tran}}^{(i)}$. From Fig. 5, we can find that $\bar{s}_{\mathrm{rot}} = 1.11$ deg and $\bar{s}_{\mathrm{tran}} = 1.85$ mm.

Although two unknown poses $\mathbf{X}$ and $\mathbf{Y}$ have been determined, only one pose $\mathbf{X}$ (or $\mathbf{Y}$) is enough to compute $\mathbf{g}_i$ [see Eqs. (3) or (2)]. In our experiments, Eq. (3) is used for computing $\mathbf{g}_i$. We now construct pairs of 3-D points $(\mathbf{g}_i, \xi_i)$, $(i = 1, \ldots, 50)$, from which the optimal parameters, $\mathbf{S}$ and $\mathbf{v}$ can be calculated by using Eqs. (7) and (9). Let us define $\mathbf{u}_i = (u_{i,x}, u_{i,y}, u_{i,y})^{\mathrm{T}} \in \mathbb{R}^3$ and $\mathbf{w}_i = (w_{i,x}, w_{i,y}, w_{i,y})^{\mathrm{T}} \in \mathbb{R}^3$ as

$$\mathbf{u}_i := \xi_i - \mathbf{g}_i, \tag{12}$$

$$\mathbf{w}_i := h(\xi_i) - \mathbf{g}_i = \mathbf{S}\xi_i + \mathbf{v} - \mathbf{g}_i, \tag{13}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the error vectors of ξ_i and $h(\xi_i)$, respectively. Note that $\mathbf{u}_i$ and $\mathbf{w}_i$, respectively, represent the errors of eye positions before and after compensation.

Figure 6 shows the experimental results of componentwise errors in the directions of the x, y, and z axes. In Fig. 7, the componentwise errors of ξ_i and $h(\xi_i)$ are given with respect to $g_{i,z} \in \mathbb{R}$ that is the third component of $\mathbf{g}_i = (g_{i,x}, g_{i,y}, g_{i,z})^{\mathrm{T}}$. From Fig. 7(c), one can notice that the error in z-axis, $u_{i,z}$ increases rapidly as $g_{i,z}$ becomes large. As shown in Figs. 6 and 7, we can obtain much smaller errors after applying the registration function h. Table 1 summarizes the results of this experiment.

5.2 *Second Experiment: 3-D Face Tracker Using a Monocular Color Camera*

To validate the effectiveness of our method, of which the main task is to determine two parameters $\mathbf{S}$ and $\mathbf{v}$ in Eq. (5), we conduct another experiment using a commercial 3-D face tracking software, FaceAPI™ with a monocular webcam (Logitech C905). In scientific literature, the FaceAPI™ is often used for obtaining reliable head poses.[9,23,24]

By following the same procedure as the first experiment, we can collect data $\{(\mathbf{A}_i, \mathbf{B}_i, \xi_i) | i = 1, \ldots, 87\}$ and then determine $(\mathbf{X}, \mathbf{Y})$ by using a hand-eye and robot-world calibration method. From Eqs. (12) and (13), we can compute the errors, $\mathbf{u}_i$ and $\mathbf{w}_i$ as shown in Figs. 8 and 9. We could find that the errors of 3-D eye position estimates in the

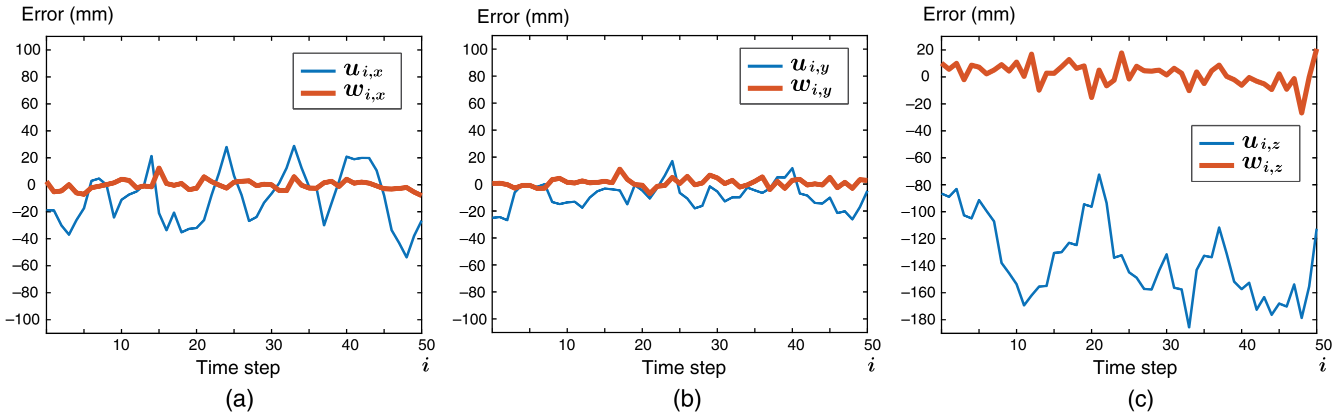

Fig. 6 Eye position errors when using a 3-D face tracker with a factory-calibrated RGB-D camera. Errors at (a)–(c) x-, y-, and z-axes.

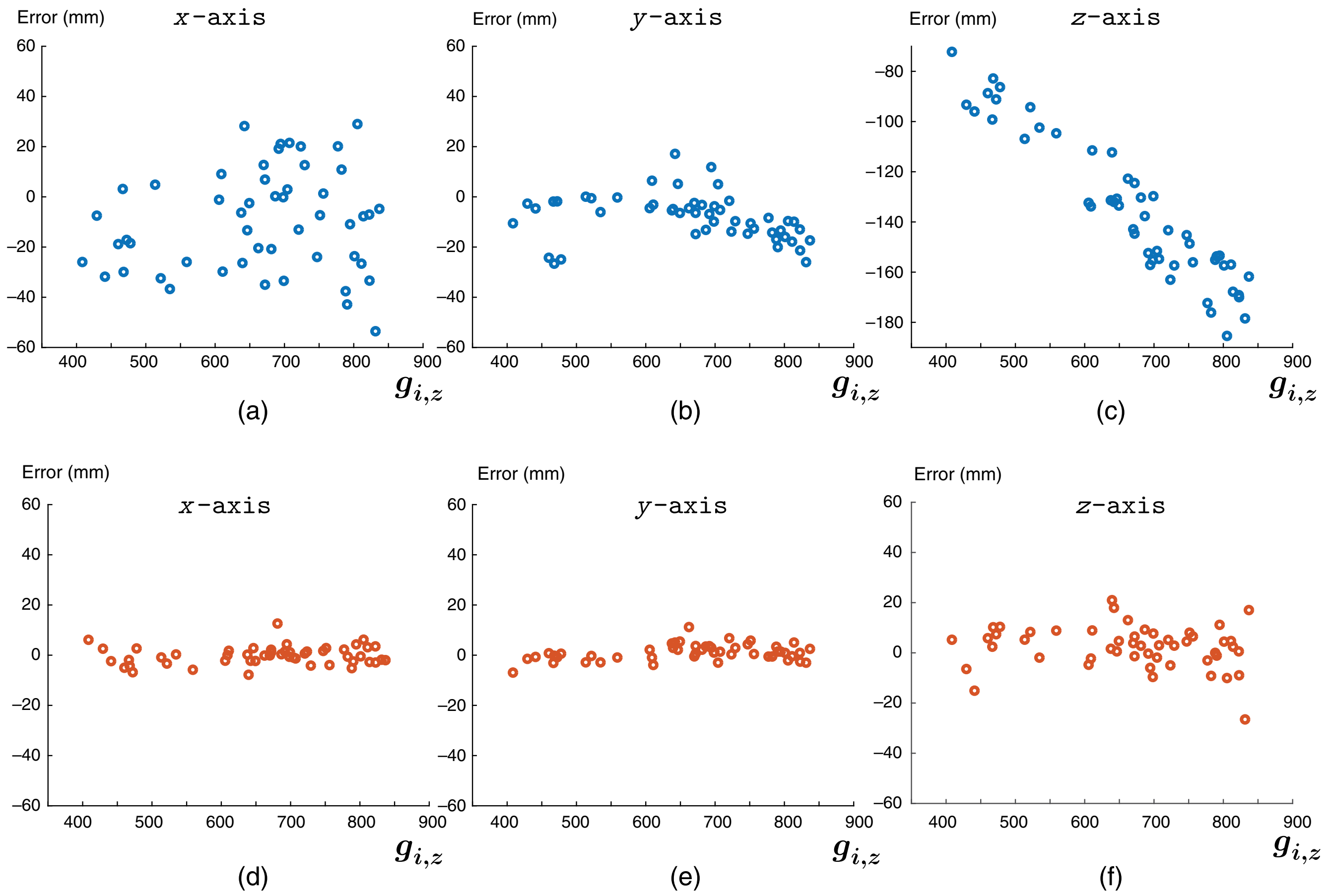

Fig. 7 Eye position errors with respect to $g_{i,z}$ when using a 3-D face tracker with a factory-calibrated RGB-D camera. Errors of (a)–(c) $\mathbf{u}_{i,x}$, $\mathbf{u}_{i,y}$, $\mathbf{u}_{i,z}$ and (d)–(e) $\mathbf{w}_{i,x}$, $\mathbf{w}_{i,y}$, $\mathbf{w}_{i,z}$.

Table 1 When using a 3-D face tracker with a factory-calibrated RGB-D camera: componentwise error (mm), $(i = 1, \ldots, 50)$.

	Before compensation			After compensation		
	$\mathbf{u}_i := \xi_i - \mathbf{g}_i$			$\mathbf{w}_i := h(\xi_i) - \mathbf{g}_i$		
	$u_{i,x}$	$u_{i,y}$	$u_{i,z}$	$w_{i,x}$	$w_{i,y}$	$w_{i,z}$
Average	−10.2	−8.4	−135.7	−0.3	0.9	2.2
Standard deviation	20.3	9.2	28.8	3.6	3.2	8.4

direction of z-axis are larger than those in the directions of the x- and y-axes from Figs. 8(c), 9(c), and 9(f). Table 2 demonstrates that our method can considerably improve the accuracy of the 3-D eye position tracker with a monocular color camera.

5.3 *Third Experiment: Applying the Proposed Method to Autostereoscopic Display Systems*

We perform real experiments by applying the proposed method for improving eye position trackers to the recently

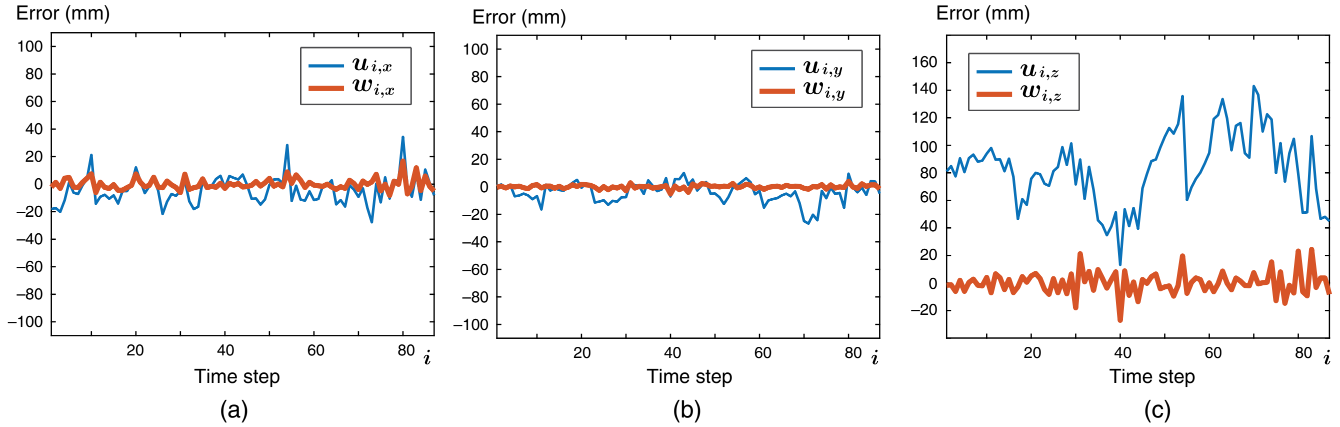

Fig. 8 Eye position errors when using a 3-D face tracker, FaceAPI™ with monocular color camera. Errors at (a)–(c) x-, y-, and z-axes.

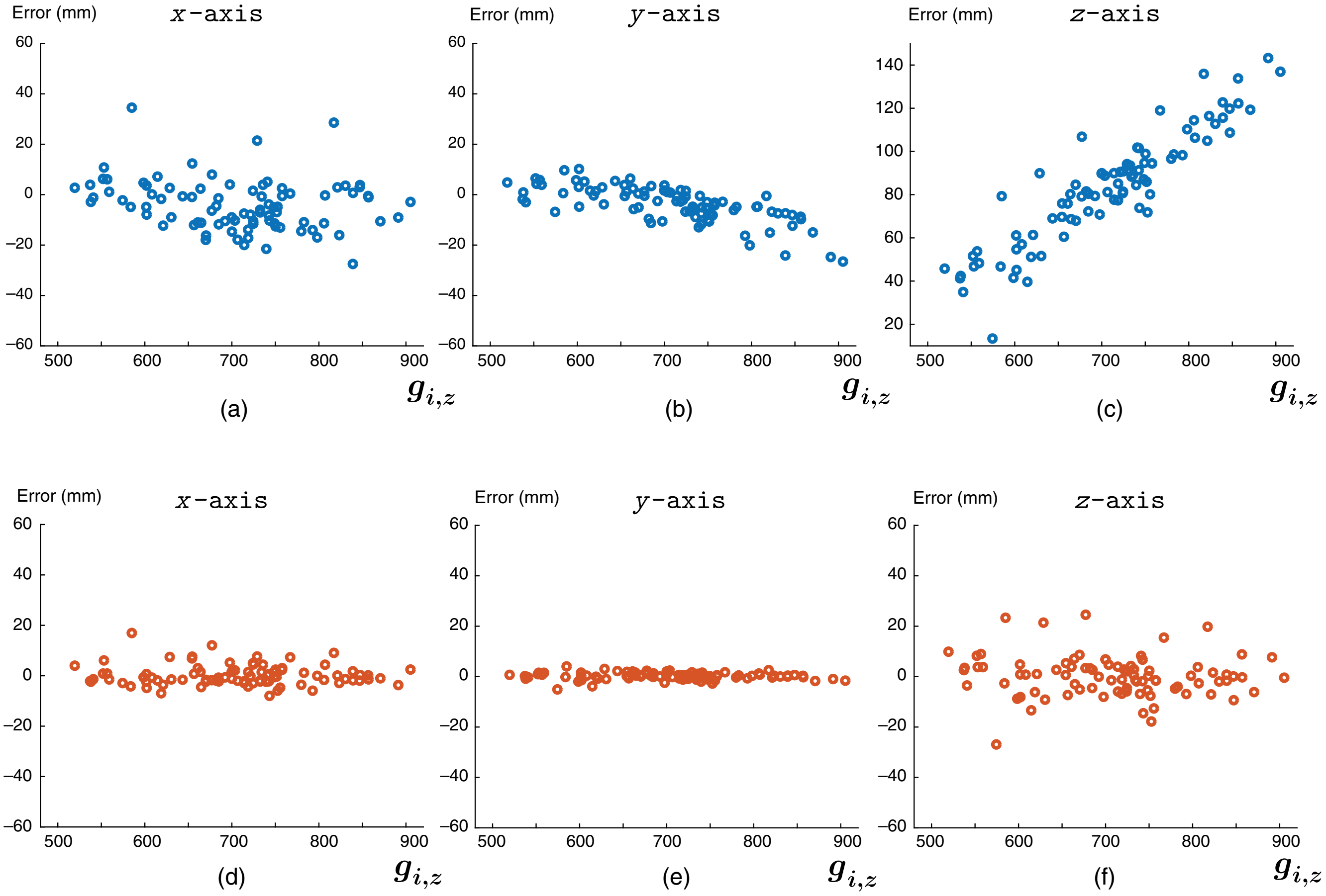

Fig. 9 Eye position errors with respect to the ground truth z values using a 3-D face tracker, FaceAPI™ with monocular color camera. Errors of (a)–(c) $\mathbf{u}_{i,x}$, $\mathbf{u}_{i,y}$, $\mathbf{u}_{i,z}$ and (d)–(e) $\mathbf{w}_{i,x}$, $\mathbf{w}_{i,y}$, $\mathbf{w}_{i,z}$.

developed autostereoscopic display system.[6] As explained in the paper,[6] the crosstalk can be dramatically reduced by combining the algorithm of view image defragmentation and an accurate eye position tracker.

In this experiment, we use the same 3-D display hardware system as described in the paper[6] and the FaceAPI™, for which the accuracy is already analyzed in Table 2. For readers' conveniences, we specify the parameters of the designed autostereoscopic 3-D display in Table 3. Figure 10 shows two different photographs taken by the camera on the mockup face as shown in Fig. 1(b) at the optimal viewing distance (OVD), i.e., 1 m between the display panel and the viewer. More specifically, Figs. 10(a) and 10(b) represent the example images before and after compensating the eye tracker of FaceAPI™ by using the proposed method, respectively. When compared with Fig. 10(a), one can find that Fig. 10(b) shows the enhanced visual quality, which results from the reduced crosstalk. This experiment demonstrates that the proposed method can work as a useful tool for reducing the crosstalk in autostereoscopic 3-D rendering.

Table 2 When using a 3-D face tracker, FaceAPI™ with monocular color camera: componentwise error (mm), $(i = 1, \ldots, 87)$.

	Before compensation			After compensation		
	$\mathbf{u}_i := \xi_i - \mathbf{g}_i$			$\mathbf{w}_i := h(\xi_i) - \mathbf{g}_i$		
	$u_{i,x}$	$u_{i,y}$	$u_{i,z}$	$w_{i,x}$	$w_{i,y}$	$w_{i,z}$
Average	−4.2	−3.9	83.2	0.1	−0.1	−0.2
Standard deviation	10.1	7.3	26.0	4.1	1.4	8.2

Table 3 Parameters of the 3-D display system (PB, parallax barrier; DP, display panel).

Design parameters	Values
Display size (diagonal)	30 in.
Subpixel size	83.5 μm × 250.5 μm
Total number of view images	12
Number of view images for defragmentation	3
Unit view resolution	640 × 533
Slanted angle of PB slit from vertical line	arctan (1/3)
OVD	1000 mm
Interval between neighboring views at OVD	16.25 mm
Gap between DP and PB	5.1385 mm
PB slit period	0.99688 mm
PB slit aperture width	74.766 μm

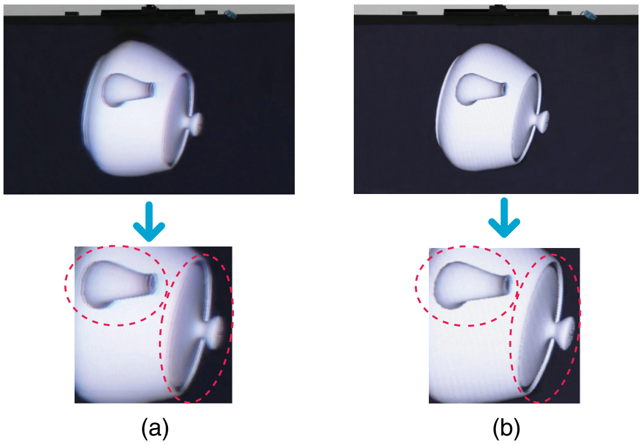

Fig. 10 Comparison of visual quality (a) before and (b) after compensating the eye position tracker by using the proposed method.

6 Conclusion

In this paper, we have presented a postprocessing method for improving the accuracy of 3-D eye position trackers with a monocular color camera by applying a registration function to the position estimates of eye trackers. The ground truths of 3-D eye positions have been obtained by constructing two types of systems consisting of an optical mocap system and checkerboards and then exploiting a hand-eye and robot-world calibration method. Experiments with real data demonstrate that our proposed method can considerably improve the accuracy of eye position trackers.

Appendix A: Algorithm for the Hand-Eye and Robot-World Calibration

This appendix summarizes a recent geometric optimization algorithm[20] for minimizing Eq. (4). A stochastic version of the algorithm that attempts to find the global minimizer is also presented in the paper.[20]

A.1 Determining an Initial Guess

First, let us denote $(\mathbf{R}_X, \mathbf{R}_Y) \in \mathrm{SO}(3) \times \mathrm{SO}(3)$ as the rotational parts of unknown $(\mathbf{X}, \mathbf{Y})$ satisfying Eq. (4) when N pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$, $(i = 1, \dots, N)$ are given. To obtain the initial guess $(\mathbf{R}_{X0}, \mathbf{R}_{Y0}) \in \mathrm{SO}(3) \times \mathrm{SO}(3)$ of unknown $(\mathbf{R}_X, \mathbf{R}_Y)$, it is sufficient to consider only the rotational parts $(\mathbf{R}_{A_i}, \mathbf{R}_{B_i}) \in \mathrm{SO}(3) \times \mathrm{SO}(3)$ of the pose data $(\mathbf{A}_i, \mathbf{B}_i)$.

By selecting any $k \in [1, N]$ at random, we can construct $(N-1)$ pairs of equations as follows: $\mathbf{R}_{A_k}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_k}$ and $\mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_i}$ $(i = 1, \dots, N$ and $i \neq k)$. By eliminating $\mathbf{R}_Y$ from the above equations, we can obtain $(N-1)$ equations: $\mathbf{R}_{A_k}^{\mathrm{T}}\mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_X\mathbf{R}_{B_k}^{\mathrm{T}}\mathbf{R}_{B_i}$. These equations can be rewritten as $\mathbf{R}_X\beta_{1i} = \alpha_{1i}$, where $[\alpha_{1i}] := \log(\mathbf{R}_{A_k}^{\mathrm{T}}\mathbf{R}_{A_i})$ and $[\beta_{1i}] := \log(\mathbf{R}_{B_1}^{\mathrm{T}}\mathbf{R}_{B_i})$. Recall that $[\mathbf{r}] \in \mathrm{so}(3)$ denotes the 3×3 skew-symmetric matrix representation of the form $[\mathbf{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$ for any $\mathbf{r} = (r_1, r_2, r_3)^{\mathrm{T}} \in \mathbb{R}^3$, which is explained in Sec. 3. Using the main result in the paper,[21] we can obtain $\mathbf{R}_{X0}$ as

$$\mathbf{R}_{X0} = (\mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1/2}\mathbf{U}^{\mathrm{T}}, \tag{14}$$

where $\mathbf{U} = \sum_{i=1}^{N-1} \beta_{1i}\alpha_{1i}^{\mathrm{T}}$. Let us define $\tilde{\beta}_{1i}$ and $\tilde{\alpha}_{1i}$ as $[\tilde{\alpha}_{1i}] := \log(\mathbf{R}_{A_k}\mathbf{R}_{A_i}^{\mathrm{T}})$ and $[\tilde{\beta}_{1i}] := \log(\mathbf{R}_{B_k}\mathbf{R}_{B_i}^{\mathrm{T}})$. In a similar way, we can obtain

$$\mathbf{R}_{Y0} = (\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1/2}\mathbf{V}^{\mathrm{T}}, \tag{15}$$

where $\mathbf{V} = \sum_{i=1}^{N-1} \tilde{\beta}_{1i}\tilde{\alpha}_{1i}^{\mathrm{T}}$.

A.2 Line Search

The objective function can be defined as $J(\mathbf{R}_X, \mathbf{R}_Y) = \frac{1}{2}\sum_{i=1}^{18} \lambda_i[\mathrm{tr}(\mathbf{P}_i\mathbf{R}_X) + \mathrm{tr}(\mathbf{Q}_i\mathbf{R}_Y)]^2 + \mathrm{tr}(\mathbf{P}_0\mathbf{R}_X) + \mathrm{tr}(\mathbf{Q}_0\mathbf{R}_Y) + c$, where $\mathbf{P}_i, \mathbf{Q}_i \in \mathbb{R}^{3\times 3}$, $\mathrm{tr}(\cdot)$ denotes the trace of matrix and λ_i, $c \in \mathbb{R}$ are given by the eigenvalue analysis of the original function [Eq. (4)] (see Appendix B of Ref. 20). We can expand $(\mathbf{R}_X, \mathbf{R}_Y)$ about $(\mathbf{R}_{X_k}, \mathbf{R}_{Y_k})$ via the matrix exponential as

$$\mathbf{R}_X = \mathbf{R}_{X_k}\left(\mathbf{I} + [\omega_{R_X}] + \frac{1}{2}[\omega_{R_X}]^2 + \dots\right),$$

$$\mathbf{R}_Y = \mathbf{R}_{Y_k}\left(\mathbf{I} + [\omega_{R_Y}] + \frac{1}{2}[\omega_{R_Y}]^2 + \dots\right).$$

The gradient and Hessian in analytic forms can be obtained by differentiating $J(\mathbf{R}_X, \mathbf{R}_Y)$ with respect to ω_{R_X} and ω_{R_Y}. The search direction for the geometric version of the steepest descent method is given by

$$\begin{bmatrix} \omega_{R_X} \\ \omega_{R_Y} \end{bmatrix} = -\nabla J, \tag{16}$$

while for Newton's method

$$\begin{bmatrix} \omega_{R_X} \\ \omega_{R_Y} \end{bmatrix} = -[\nabla^2 J]^{-1}\nabla J. \tag{17}$$

The detailed derivations of the above Eqs. (16) and (17) are given in Appendix C of Ref. 20.

A.3 Stepsize Estimate

From the result in Appendix D of the paper,[20] the analytic equation for a strictly descending stepsize estimate t^* is given by

$$t^* = -\frac{\phi'(0)}{c}, \tag{18}$$

where $\phi'(0) = \sum_{i=1}^{18} \lambda_i\mathrm{tr}(\mathbf{P}_i\mathbf{R}_{X_k} + \mathbf{Q}_i\mathbf{R}_{Y_k})\mathrm{tr}(\mathbf{P}_i\mathbf{R}_{X_k}[\omega_{R_X}] + \mathbf{Q}_i\mathbf{R}_{Y_k}[\omega_{R_Y}]) + \mathrm{tr}(\mathbf{P}_0\mathbf{R}_{X_k}[\omega_{R_X}] + \mathbf{Q}_0\mathbf{R}_{Y_k}[\omega_{R_Y}])$ and $c = |\lambda|_{\max}(\|[\omega_{R_X}]\|^2 + \|[\omega_{R_Y}]\|^2) + \sqrt{6}|\lambda|_{\max}\sqrt{\|[\omega_{R_X}]^2\|^2 + \|[\omega_{R_Y}]^2\|^2} + \sqrt{3}(\|\mathbf{P}_0\mathbf{R}_{X_k}[\omega_{R_X}]^2\| + \|\mathbf{Q}_0\mathbf{Y}_k[\omega_{R_Y}]^2\|)$. Using $(\omega_{R_X}, \omega_{R_Y})$ in [Eqs. (16) or (17)] and t^* in Eq. (18), $(\mathbf{R}_X, \mathbf{R}_Y)$ can be updated as

$$\mathbf{R}_{X_{k+1}} = \mathbf{R}_{X_k}e^{[\omega_{R_X}]t^*}, \tag{19}$$

$$\mathbf{R}_{Y_{k+1}} = \mathbf{R}_{Y_k}e^{[\omega_{R_Y}]t^*}. \tag{20}$$

Table 4 Algorithm for hand-eye and robot-world calibration.

```
1 Initialization
  Set $(\mathbf{R}_{X_0}, \mathbf{R}_{Y_0})$ using Eqs. (14) and (15).
  Set $k = 0$.
2 Set search direction
  Find $\omega_{R_X}$ and $\omega_{R_Y}$ using Eqs. (16) or (17)
3 Update
  Compute stepsize $t^*$ using Eq. (18).
  Find $(R_{X_{k+1}}, R_{Y_{k+1}})$ using Eqs. (19) and (20).
4. Check local convergence
  If local convergence criterion is satisfied
    break and return $(R_{X_{k+1}}, R_{Y_{k+1}})$
  Else
    $k \leftarrow k + 1$
    go to Step 2
```

If a certain local convergence criterion is satisfied, then $(\mathbf{R}_{X_{k+1}}, \mathbf{R}_{Y_{k+1}})$ is the optimal solution, which we want to find. Otherwise, we update the time step as $k \leftarrow k + 1$ and go back to the stage of the search direction. Table 4 summarizes the geometric optimization algorithm using a local search algorithm.

Acknowledgments

This work was supported by The Cross-Ministry Giga KOREA Project grant from the Ministry of Science, ICT, and Future Planning, Korea.

References

1. J.-Y. Son et al., "Three-dimensional imaging for creating real-world-like environments," *Proc. IEEE* **101**, 190–205 (2013).
2. D. Kim et al., "Effect of parallax distribution and crosstalk on visual comfort in parallax barrier autostereoscopic display," *Opt. Eng.* **54**(5), 053107 (2015).
3. D. Kim et al., "Parallax adjustment for visual comfort enhancement using the effect of parallax distribution and cross talk in parallax-barrier autostereoscopic three-dimensional display," *Opt. Eng.* **54**(12), 123104 (2015).
4. N. A. Dodgson, "Autostereoscopic 3D displays," *IEEE Computer* **38**, 31–36 (2005).
5. S.-K. Kim et al., "Parallax barrier engineering for image quality improvement in an autostereoscopic 3D display," *Opt. Express* **23**, 13230–13244 (2015).
6. S.-K. Kim et al., "Defragmented image based autostereoscopic 3D displays with dynamic eye tracking," *Opt. Commun.* **357**, 185–192 (2015).
7. A. Fassi et al., "Optical eye tracking system for noninvasive and automatic monitoring of eye position and movements in radiotherapy treatments of ocular tumors," *Appl. Opt.* **51**(13), 2441–2450 (2012).
8. Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *IEEE Trans. Biomed. Eng.* **54**(12), 2246–2260 (2007).
9. N. M. Bakker et al., "Accurate gaze direction measurements with free head movement for strabismus angle estimation," *IEEE Trans. Biomed. Eng.* **60**(11), 3028–3035 (2013).
10. H. C. Lee et al., "Gaze tracking system at a distance for controlling IPTV," *IEEE Trans. Consum. Electron.* **56**(4), 2577–2583 (2010).
11. Y. Ebisawa and K. Fukumoto, "Long–range gaze tracking system for large movements," *IEEE Trans. Biomed. Eng.* **60**(12), 3432–3440 (2013).
12. S. Song and M. Chandraker, "Robust scale estimation in real–time monocular SFM for autonomous driving," in *IEEE Conf. CVPR*, pp. 23–28 (2014).
13. P. J. Phillips et al., "Fast, reliable head tracking under varying illumination: an approach based on registration of texture–mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(4), 322–336 (2000).
14. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: an integrated system and evaluation for monitoring driver awareness," *IEEE Trans. Intell. Transp. Syst.* **11**(2), 300–311 (2010).
15. R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Trans. Image Process.* **21**(2), 802–815 (2012).
16. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 607–626 (2010).
17. D. W. Hansen and Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(3), 478–500 (2010).
18. H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form $AX = YB$," *IEEE Trans. Rob. Autom.* **10**, 549–554 (1994).
19. F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Trans. Rob. Autom.* **14**, 617–622 (1998).
20. J. Ha, D. Kang, and F. C. Park, "A stochastic global optimization algorithm for the two-frame sensor calibration problem," *IEEE Trans. Ind. Electron.* **63**, 2434–2446 (2016).
21. F. C. Park and B. J. Martin, "Robot sensor calibration: solving $AX = XB$ on the Euclidean group," *IEEE Trans. Rob. Autom.* **10**(5), 717–721 (1994).
22. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330–1334 (2000).
23. M. M. Hoque et al., "Effect of robot's gaze behaviors for attracting and controlling human attention," *Adv. Rob.* **27**(11), 813–829 (2013).
24. D. Das et al., "Supporting human–robot interaction based on the level of visual focus of attention," *IEEE Trans. Hum. Mach. Syst.* **45**(6), 664–675 (2015).

Donghoon Kang received his BS and MS degrees in mechanical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea, in 1997 and 1999, respectively. Currently, he is working toward the PhD degree at Seoul National University, Seoul, Republic of Korea. Since 2000, he has been at Korea Institute of Science and Technology (KIST), where he is currently a research scientist. His current research interests include signal processing and computer vision.

Jinwook Kim received his BS degree in mechanical design and production engineering from Seoul National University, Seoul, Republic of Korea, in 1995, and his MS and PhD degrees in mechanical and aerospace engineering from Seoul National University in 1997 and 2002, respectively. He is currently a principal research scientist at Korea Institute of Science and Technology (KIST). His current research interests include innovative human–computer interaction techniques, physics-based animation, and image-based modeling.

Sung-Kyu Kim received his BS, MS, and PhD degrees from the Quantum Optics Group of Physics, Korea University, Seoul, Republic of Korea, in 1989, 1991, and 2000, respectively. Then, he spent 2 years as an invited research scientist at TAO of Japan. In 2001, he was appointed as a research scientist at Korea Institute of Science and Technology (KIST). Currently, he is a principal research scientist at KIST. His current research interests include three-dimensional display systems.