Master's Thesis
석사 학위논문

A Simple and Accurate Camera-sensor Calibration for Surgical Endoscopes and Microscopes

Seongpung Lee (이 성 풍 李 成 豊)

Department of Robotics Engineering
로봇공학전공

DGIST

2015

Master's Thesis
석사 학위논문

A Simple and Accurate Camera-sensor Calibration for Surgical Endoscopes and Microscopes

Seongpung Lee (이 성 풍 李 成 豊)

Department of Robotics Engineering

로봇공학전공

DGIST

2015

A Simple and Accurate Camera-sensor Calibration for Surgical Endoscopes and Microscopes

Advisor : Professor Jaesung Hong

Co-advisor : Professor Sanghyun Joung

by

Seongpung Lee
Department of Robotics Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Robotics Engineering. The study was conducted in accordance with Code of Research Ethics[1]

. . 2015

Approved by

Professor Jaesung Hong (_____)

Professor Sanghyun Joung (_____)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

A Simple and Accurate Camera-sensor Calibration for Surgical Endoscopes and Microscopes

Seongpung Lee

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

. . 2015

Head of Committee Prof. Jaesung Hong (인)

Committee Member Prof. Sanghyun Joung (인)

Committee Member Prof. Pyunghun Chang (인)

MS/RT
201323008

이 성 풍. Seongpung Lee. A Simple and Accurate Camera-sensor Calibration for Endoscopes and Microscopes. Department of Robotics Engineering. 2015. 62p. Advisors Prof. Jaesung Hong, Prof. Co-Advisors Sanghyun Joung.

ABSTRACT

Augmented reality (AR) has become a key technology for surgical navigation system, which is the technology to overlay virtual objects on real one. In AR system, camera-sensor calibration is one of major factors to affect the accuracy. In order to perform camera-sensor calibration, it has been a common method to move a camera to establish and solve an **AX = XB** type. However, in the clinical environments, endoscopes and microscopes are commonly used and moving manually those cameras leads to inconvenience and inaccuracy due to heavy weight and large size. In addition, since optical tracking system has spatial error which increases with the distance, moving the camera which requires large motion to make necessary orientations between a calibration pattern and the camera makes the system affected by the spatial error. Therefore, we propose a method to solve the camera-sensor matrix. It is not mathematically strong to the noise, but possible to reduce the effect of the spatial error of optical tracking system effectively, and to provide users more convenient method. The proposed method can be easily performed by mounting an additional marker on the calibration pattern, which produces **AX = BYC** type formula. Through experiments, we compared the **AX = BYC** solution with the **AX = XB** solution in terms of the accuracy. As a result, we found the proposed method is more convenient, accurate, and stable than the conventional method.

Keywords: Camera Calibration, Hand-eye Calibration, Surgical Endoscopes, Surgical Microscopes.

Contents

List of contents

List of tables

List of figures

Ⅰ. INTRODUCTION

1.1 Introduction to surgical navigation system

Nowadays, the advanced computer graphic, electrical, mechanic, and medical imaging technologies as well as increasing needs of surgeons affect making surgical navigation system more accurate and precise than past one. Surgical navigation system is to let surgeons know accurately where surgical instruments are in patient's body in real time by using CT/MR images taken before surgery. It makes surgeons possible to operate the challenging targets which could not be hardly accessed without helps of surgical navigation system due to lack of information. In addition, it is possible to be safer and less invasive surgery, and to reduce time to operate, which can lead to reducing blood loss and minimizing trauma of patients [1-5]. In addition, aforementioned advantages of surgical navigation system result in reducing hospitalization which means it can decrease hospital costs. Therefore, surgical navigation system has emerged as one of the most reliable technologies and can be found in various surgical fields such as ENT, neuro, and orthopedic surgery. That is, surgical navigation system became to a part in surgery today.

External trackers are usually used to track surgical instruments in surgical navigation system. The kinds of the external trackers used in surgical navigation system are generally optical tracking system and electromagnetic tracking system. The basic concept of optical tracking system is simply to emit infrared signals and to receive the infrared signals reflected by markers. That is, optical tracking system emits infrared signals and the infrared signals are reflected by markers mounted on a patient or surgical instruments. Finally, optical tracking system receives the reflected infrared signals and calculates the pose (position and orientation) of markers relative to optical tracking system. Therefore, it is very important to keep the line of sight between optical tracking

system and markers attached on the object to be measured. If the line of sight is lost, optical tracking system cannot find the location of markers. This is an issue still mentioned in relative literature although optical tracking system has high accuracy. In comparison with optical tracking system, electromagnetic tracking system uses electromagnet to track the pose of sensor which works as markers in optical tracking system. Therefore, it doesn't need to keep the line of sight. It is very advantageous for the surgeries since there are many things such as surgeons, assistants, and various large devices which have potential to keep the line of sight off in the operating room. However, due to its property, it is very sensitive to metallic objects. In addition, its accuracy is less than optical tracking system [6].

The key technology of surgical navigation system is the registration between patient coordinate system and image coordinate system [7, 8]. The representative methods are as following: paired point registration (PPR) called "marker-based registration" and surface registration called "marker-less registration" [8, 9]. PPR is the method to register patient and image coordinate by using several paired points relative to each coordinate system. Generally, the number of points used in registration procedure is more than 3, which has to be linearly independent to each other to ensure the unique solution of rotation in three dimension [9]. Surface registration uses surface data or point cloud from each coordinate system. In this method, iterative closest point (ICP) algorithm is usually used to achieve the optimal solution. Since ICP is based on numerical analysis, the local minima problem are still issued. Therefore, in order to improve the accuracy of registration and to ensure reliable result, hybrid registration method, which the surface registration is performed with the initial matrix obtained from paired point registration is usually used [10]. Once the registration is performed, arbitrary point relative to patient coordinate system can be transformed to image coordinate system. Therefore, the location of surgical instruments with markers can be displayed on CT/MR image.

A drawback of the conventional surgical navigation system is to make surgeons to change of their sight which may lost the relationship between the guidance data to the operative field [11]. Endoscopes and microscopes are generally used for examining the inside of patient precisely. If surgeons use the conventional surgical navigation system, they have two displays. One is for surgical navigation system, which is to show the virtual world. The other is from clinical cameras, which is to show the real world. The solution to eliminate this problem is to apply augmented reality (AR) technology into the conventional surgical navigation system.

1.2 Augmented reality based surgical navigation using external tracker

As mentioned before, the method to solve the problem caused from the conventional surgical navigation system is to use AR technology. AR technology is the technology to overlay virtual objects on real one. That is, the aforementioned problem is solved by superimposing virtual objects represented in virtual space on the endoscopic image or the microscopic image. Therefore, One of the important requirements to build AR system is the information about the location the virtual objects overlays. However, in practice, the virtual objects are fixed and the image plane of a camera, which will be described later, is located in the backside of the virtual objects by using the location information. There is no image plane in the virtual space; Therefore, a virtual image plane is constructed by considering field of view (FOV) of the camera. After that, the image taken by the camera is covered on the virtual image plane. If the location of the virtual image plane are known from Fig. 1.1, the virtual image plane can be located in the backside of the virtual objects. Then, the virtual objects seem like to be overlaid on the real one.

From the conventional surgical navigation system, we already know the relationship from patient to image coordinate system. However, in AR system, to locate accurately the virtual image plane in the backside of the

virtual objects, following two conditions have to be fulfilled: first, arbitrary point relative to camera coordinate system can be transformed to image coordinate system since the virtual objects are formed relative to image coordinate system and the real image plane are formed relative to camera coordinate system. Second, the location of the real image plane relative to the camera coordinate system, p_C in Fig. 1.1, has to be solved. First condition can be fulfilled by finding the relationship between camera and patient coordinate system. The technology to find that relationship is called camera-sensor calibration we will deal with in this paper. Second condition is fulfilled by finding intrinsic parameters, which can be obtained from camera calibration we will explain later. By making the conditions fulfilled, the virtual image plane can be accurately located in the backside of the virtual objects. That is, the accuracy of AR is significantly depending on the result of camera-sensor calibration.

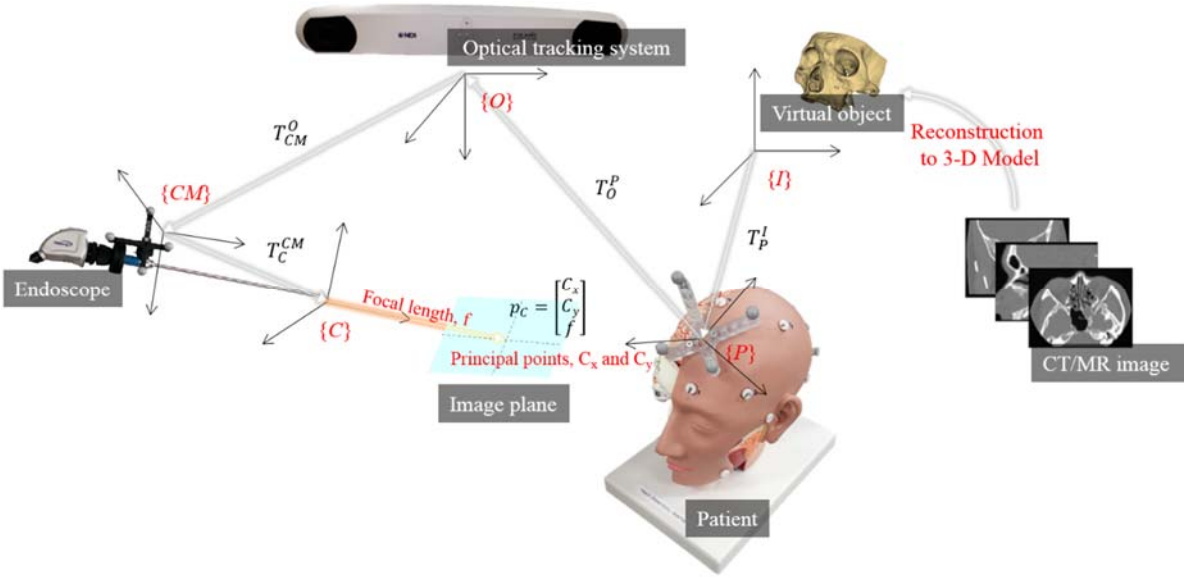


Figure 1.1 Concept of AR system. By using those relationships, the virtual image plane can be located in the backside of the 3-D models reconstructed from CT/MR image. It makes the virtual objects seem like to be overlaid on the real one.

1.3 Camera-sensor calibration in the robotic application

Camera-sensor calibration is well known as hand-eye calibration in the robotic application, **AX = XB**

problem, and is the method to solve the relationship between end-effector of a robot and the coordinate system of the camera lens as shown in Fig. 1.2 (a). Many methods to solve camera-sensor calibration have been researched by [12-17]. Those are largely separated by two methods. One is to solve rotation part and translation part separately. The other is to solve rotation part and translation part simultaneously. Axis-angle and quaternion are well known as the separated method [12-14]. One of drawbacks of those methods is that the error caused by rotation part affects the result of translation part. Therefore, in order to eliminate the propagation error, dual quaternion and Kronecker product were developed by [15, 16].

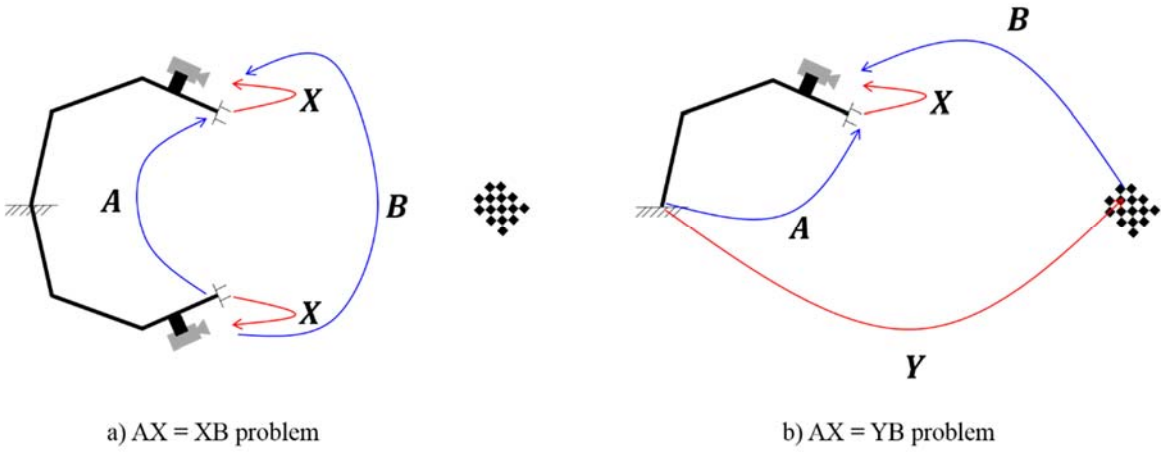


Figure 1.2 Problems to solve hand-eye matrix. “hand” means a robot arm holding a camera and “eye” means the center of camera lens in the robotic application. In order to transform a point relative to the center of camera lens to the robot base, hand-eye matrix has to be solved.

Fig. 1.2 (b) describes **AX = YB** problem. Methods to solve **AX = YB** problem are basically the same as methods to solve **AX = XB** problem [18-20]; however, **AX = YB** problem provides us the additional information, **Y** which is used for the evaluation of accuracy of **X** and solve **X** without the re-calibration. In order to solve **AX = XB** or **AX = YB**, a camera mounted rigidly on a robot arm must be moved since **Y** has to be constant to avoid camera-sensor calibration to be failed.

Thanks to the robot, camera-sensor calibration can be automatically performed by setting trajectory well.

1.4 Problem of camera-sensor calibration in the clinical application

In the clinical application such as AR based surgical navigation system, endoscopes or microscopes are commonly used for the surgery. Those cameras have heavy weight and large size. In addition, in the operating room, it is hard to set up the robot system for performing camera-sensor calibration automatically due to limitation of the size of the operating room and its cost. It means surgeons have to move manually the camera to perform camera-sensor calibration for building AR system as shown in Fig. 1.3. The problems such as the size and the weight of the clinical camera, and moving manually the camera lead to following two problems in performing camera-sensor calibration: inconvenience of camera-sensor calibration process, large range of motion of the camera.

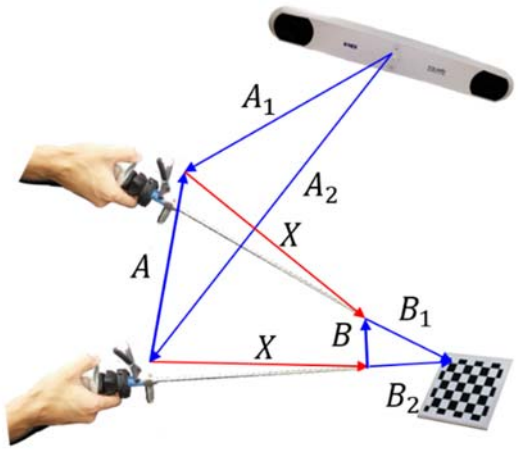


Figure 1.3 Camera-sensor calibration in the clinical application. Compring camera-sensor calibration in the robotic application, optical tracking system and passive markers mounted rigidly on a camera are corresponding to a robot base and a robot arm holding a camera, respectively.

First problem caused by its weight and size is the difficulty from moving the camera in performing camera-sensor calibration. Especially, when there are no the experts to perform camera-sensor calibration by using those cameras in the surgical team, it is more difficult work than former case. This problem is clearly obvious in practice without any proof. Due to second problem we mentioned, which is that moving the camera causes large range of motion, it leads to increasing the affection of the spatial error of optical tracking system, which is

described by Fig. 1.4 referred from [21]. In addition, Tsai and Lens analyzed five observations to improve the result of camera-calibration [12]. According to their observations, small translation in camera motion and large orientation of camera motion between stations leads to improve the accuracy. These trues implies that if the range of motion become smaller, higher accuracy is ensured.

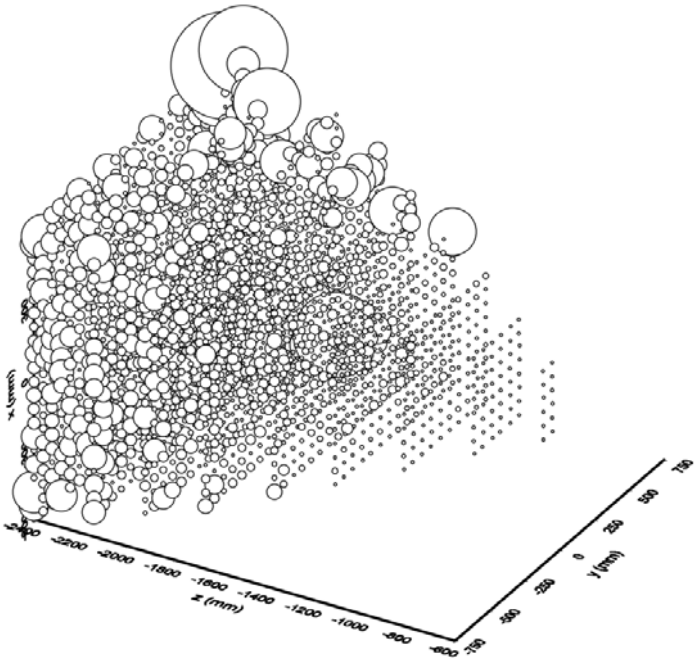


Figure 1.4 Spatial error distribution of optical tracking system. Circles mean the magnitude of the error. The spatial error increases when object we measure is far from optical tracking system according to z-axis and center of optical tracker in xy plane [6, 21].

1.5 The proposed method

In this paper, we propose simple and accurate camera-sensor calibration method. It is not mathematically robust method from the spatial error of optical tracking system. The previous studies are relative to eliminating the propagation error by solving simultaneously rotation part and translation part, and using strong representations of rotation such as axis-angle, quaternion, dual quaternion, and screw to express more compact equation. The other researches to reduce the error of camera-sensor calibration are [22-24], which are relative to data selection. The principle of [22, 24] is based on rules inferred from Tsai and Lens [12]. After poses are continuously obtained, reasonable poses which obey the rules are selected by their algorithm. And in [23], they tried to

applying inter-quartile range (IQR) to exclude outlier data. Though data selection algorithm, the performance of camera-sensor calibration methods is slightly improved. However, the methods are also based on moving the camera. The spatial error of optical tracking system is obviously reduced if the distance to optical tracking system is close, and the range of motion is small as shown in Fig 1.4.

In the medical application, the location of markers mounted rigidly on endoscopes and microscopes is restricted by its structure since markers cannot be mounted on the tip of the cylinder of endoscopes which is a part inserted in patient body for the observation. Therefore, the distance between markers and the center of camera lens cannot decrease. It leads to the large range of motion of the camera when we move the camera to acquire the various poses of the camera. In order to reduce the affection of the spatial error of optical tracking system and to ensure the convenient process to perform camera-sensor calibration, we need to move a calibration pattern instead of the camera. Our method is very simply achieved by mounting an additional passive marker on the calibration pattern to be tracked by optical tracking system. The location of markers to be mounted is free; therefore, it can be very close to the coordinate system of the calibration pattern. It means the range of motion of the calibration pattern in performing camera-sensor calibration could be smaller than moving the camera. In addition, since the size and the weight of the calibration pattern is lighter and smaller than that of the camera, surgeons can process the camera-sensor calibration easier than moving the camera. By using the calibration pattern, we can also comply with the rules due to its small range of motion during keeping the large angle between poses. Though comparing the proposed method with the conventional method, we found our method provides more convenient and comparable accuracy than the conventional method which must move the camera.

Ⅱ. METHODS

2.1 Camera-sensor calibration

2.1.1 Camera calibration

As mentioned before, the subject dealing with in this paper is camera-sensor calibration. This must be solved for building AR using external tracker such as optical tracking system in medical application and affects mainly the accuracy of AR based surgical navigation system. The accurate information about **A** and **B** of **AX = XB** problem is required for camera-sensor calibration. **A** is directly acquired from optical tracking system as raw data, but in order to acquire **B**, camera calibration is essentially required. That is, camera calibration must be performed before performing camera-sensor calibration.

Camera calibration is the process to obtain the intrinsic parameters and an extrinsic parameter. Before explaining the intrinsic parameters and the extrinsic parameter, the mathematical model of cameras has to be firstly explained. In expressing general cameras mathematically, the model, typically used, is the pin-hole camera model. Pin-hole camera model describes the projection from a point in 3-D space to a point in 2-D. Through pin-hole camera model, Eq. 1 can be obtained.

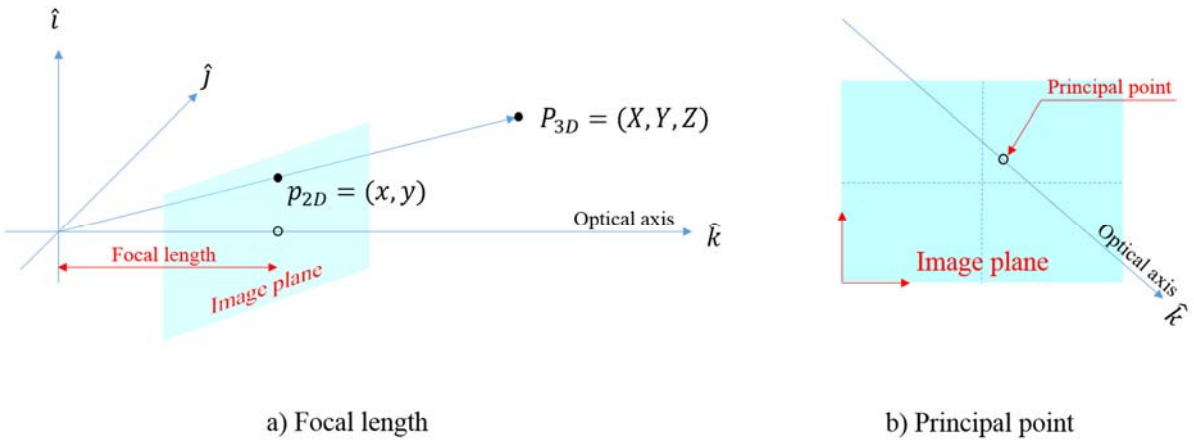
$$\begin{bmatrix} x_{img} \\ y_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

Simply, Eq. 1 is written as Eq. 2.

$$\begin{bmatrix} \mathbf{p_{2D}} \\ 1 \end{bmatrix} = \mathbf{M}[\mathbf{I_{3\times3}}|\mathbf{0_{3\times1}}]\mathbf{W}\begin{bmatrix} \mathbf{P_{3D}} \\ 1 \end{bmatrix} \quad (2)$$

f and c in Eq. 1 mean focal length which describes the distance between the center of lens of a camera and an image sensor such as CMOS and CCD called the image plane as shown Fig. 2.1 (a), and principal points which

describes the place the optical axis intersects the image plane as shown Fig. 2.1 (b), respectively. Note that the principal points are not the same as the center of the image plane.



a) Focal length b) Principal point

Figure 2.1 Focal length and principal point of a camera. Focal length is defined as the distance between the center of lens of a camera and the image sensor of a camera. Principal point is defined as the point on the image plane intersected with the optical axis.

Those parameters are defined as the intrinsic parameters and it can be expressed by the matrix form such as **M** in Eq. 2 In addition, **W** in Eq. 2 describes the relationship between world coordinate system and camera coordinate system as shown in Fig. 2.2, which is called the extrinsic parameter.

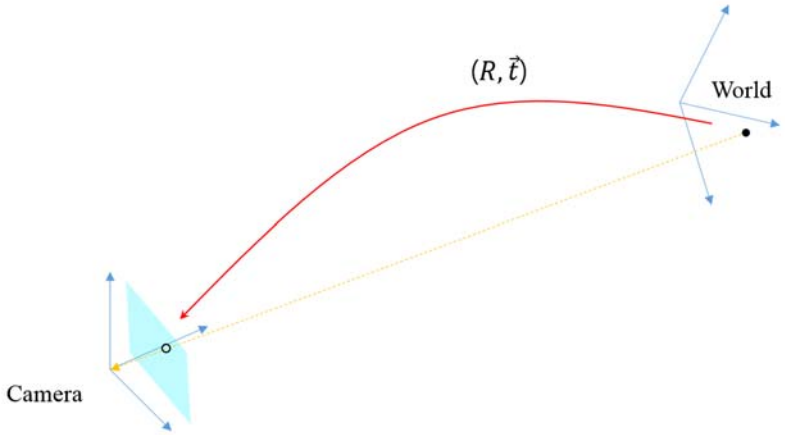


Figure 2.2 Extrinsic parameter. Extrinsic parameter is defined as the relationship between the coordinate system of a camera and that of world.

That is, if the intrinsic parameters and the extrinsic parameter are known, a point relative to world coordinate system can project to a point relative to the coordinate system of the image plane. This means that if intrinsic parameters and at least four point pairs between 2-D and 3-D are known, extrinsic parameter can be solved.

The distortion of a camera has to be considered when camera calibration is performed since in real, there is no camera lens system such as the pin-hole camera model. There are two distortion factors, which are the radial distortion and the tangential distortion. The radial distortion is caused by the shape of camera lens. This shape makes the optical line curved. Therefore, if the optical line is far and far to the center of camera lens, the radial distortion is more serious. Following two equations describes the radical distortion.

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (4)$$

where $x_{corrected}$ and $y_{corrected}$ are compensated position, x and y are the original position obtained from the image processing to find corner points, k_1, k_2, and k_3 are coefficients for expressing the radial distortion, and r is the radius of camera lens. The other factor to make points curved is the tangential distortion. This is caused by inaccurately aligned lens to CCD or CMOS chip during the manufacturing process. This distortion is expressed by following two equations.

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)] \quad (5)$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \quad (6)$$

where p_1 and p_2 are coefficients for expressing the tangential distortion. Therefore, through Eq. 3-6, the curved points by the radical and the tangential distortions are compensated. In this paper, we used Zhang's algorithm for performing camera calibration which is very famous algorithm in literature and very easy to use and ensure high accuracy [25].

2.1.2 **AX = XB** problem

The conventional method to solve camera-sensor calibration is to solve **AX = XB** problem. In the case of no movement, the relationship between optical tracking system and the calibration pattern has to be known to solve the relationship between the lens of a camera and the markers mounted on the camera. However, as shown in Fig. 2.3, the relationship is eliminated by moving the camera once.

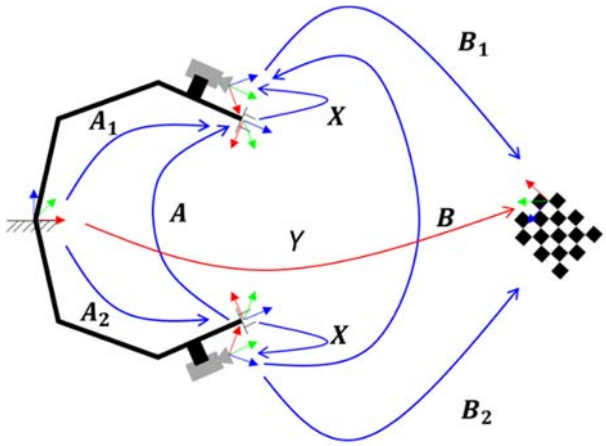


Figure 2.3 Conventional **AX = XB** problem. The relationship between the robot base and the coordinate system of the calibration pattern is elimiated by moving the camera once.

If the relationship between the lens of the camera and the markers is defined as **X** and between optical tracking system and the calibration pattern is as **Y**, we obtain Eq. 7.

$$\mathbf{A_i X = Y B_i^{-1}} \tag{7}$$

If the camera is moved to the different location, which must have non-parallel axis with previous locations, we can build two equations such as Eq. 8 and 9.

$$\mathbf{A_1 X = Y B_1^{-1}} \tag{8}$$

$$\mathbf{A_2 X = Y B_2^{-1}} \tag{9}$$

Eq. 9 can be written as Eq. 10 by multiplying both sides with $\mathbf{B_2}$.

$$\mathbf{Y = A_2 X B_2} \tag{10}$$

By replacing **Y** in Eq. 8 with Eq. 10,

$$\mathbf{A_1X = A_2XB_2B_1^{-1}} \tag{11}$$

By multiplying both sides with $\mathbf{A_2^{-1}}$,

$$\mathbf{A_2^{-1}A_1X = XB_2B_1^{-1}} \tag{12}$$

$$\mathbf{AX = XB} \tag{13}$$

where **A** is $\mathbf{A_2^{-1}A_1}$ and **B** is $\mathbf{B_2B_1^{-1}}$.

As mentioned before, **AX** = **YB** problem is directly solved by [18-20]. The advantage obtained from this equation is to obtain **Y** matrix which is possible to solve **X** without re-calibration if there is physically no change of the relationship between the lens of the camera and the markers, **X**. We know intuitively **Y** has to be constant in **AX** = **YB** problem obtained from Fig. 1.2 (b). Therefore, those kinds of **AX** = **XB** and **AX** = **YB** problems are solved by moving only the camera.

Since **A**, **B**, and **X** are homogeneous transformation matrix, they have 4-by-4 dimension, and contain rotation part and translation part. Therefore, we can represent **AX** = **XB** problem as following:

$$\begin{bmatrix} \mathbf{R_A} & \vec{\mathbf{t}}_\mathbf{A} \\ \mathbf{0_{1\times3}} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R_X} & \vec{\mathbf{t}}_\mathbf{X} \\ \mathbf{0_{1\times3}} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R_X} & \vec{\mathbf{t}}_\mathbf{X} \\ \mathbf{0_{1\times3}} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R_B} & \vec{\mathbf{t}}_\mathbf{B} \\ \mathbf{0_{1\times3}} & 1 \end{bmatrix} \tag{14}$$

In order to solve **AX** = **XB** problem, Eq. 14 has to be separated into rotation part and translation part such as Eq. 15 and 16.

$$\mathbf{R_AR_X} = \mathbf{R_XR_B} \tag{15}$$

$$\mathbf{R_A}\vec{\mathbf{t}}_\mathbf{X} + \vec{\mathbf{t}}_\mathbf{A} = \mathbf{R_X}\vec{\mathbf{t}}_\mathbf{B} + \vec{\mathbf{t}}_\mathbf{X} \tag{16}$$

From Eq. 15 and 16, rotation part is solved by using only one pose while translation part has to have at least two poses since it has 6 unknown parameters, which are rotation and translation of **X**.

X is simply solved by solving first Eq. 15, and then Eq. 16 by using the solution of Eq. 15. However, the error

from Eq. 15 affects translation part. This propagation error can be eliminated by solving rotation part and translation part simultaneously. The typical methods to solve rotation part and translation part at the same time are Kronecker product and dual quaternion we will explain.

2.1.2.1 Solution using Kronecker product

Kronecker product is also known as tensor product and is firstly used in performing camera-sensor calibration by Andreff with vectorization [16]. Kronecker product is defined as following: if **A** and **B** are m-by-n, p-by-q matrix respectively, the result of Kronecker product, $\mathbf{A} \otimes \mathbf{B}$ is mp-by-nq block matrix as Eq. 17.

$$\mathbf{A_{m\times n}} \otimes \mathbf{B_{p\times q}} = \begin{bmatrix} a_{11}\mathbf{B_{p\times q}} & \cdots & a_{1n}\mathbf{B_{p\times q}} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B_{p\times q}} & \cdots & a_{mn}\mathbf{B_{p\times \boldsymbol{q}}} \end{bmatrix} \tag{17}$$

In addition, vectorization is defined as following: if **A** is m-by-n matrix, its vectorization, denoted by $\mathrm{vec}(\mathbf{A})$ is mn-by-1 column vector as Eq. 18.

$$\mathrm{vec}(\mathbf{A_{m\times n}}) = [a_{11},\ \cdots,\ a_{m1},\ a_{12},\ \cdots,\ a_{m2},\ \cdots,\ a_{1n},\ \cdots,\ a_{mn}]^T \tag{18}$$

Vectorization is compatible with Kronecker product; therefore, it is frequently used together with the Kronecker product to express matrix multiplication as a linear transformation on matrices. To apply Kronecker product and vectorization to **AX** = **XB** problem, Eq. 15 and 16 have to be first arranged as Eq. 19 and 20, respectively.

$$\mathbf{R_A R_X} - \mathbf{R_X R_B} = 0 \tag{19}$$

$$(\mathbf{R_A} - \mathbf{I})\vec{\mathbf{t}}_\mathbf{X} + \mathbf{R_X}\vec{\mathbf{t}}_\mathbf{B} = -\vec{\mathbf{t}}_\mathbf{A} \tag{20}$$

By using aforementioned definitions (Kronecker product and vectorization), Eq. 19 and 20 are rewritten as Eq. 21 and 22, respectively.

$$\left(\mathbf{R_A} \otimes \mathbf{I_{3\times 3}} - \mathbf{I_{3\times 3}} \otimes \mathbf{R_B}^{\mathrm{T}}\right)\mathrm{vec}(\mathbf{R_X}) = 0 \tag{21}$$

$$(\mathbf{R_A} - \mathbf{I})\overrightarrow{\mathbf{t_X}} + (\mathbf{I_{3\times 3}} \otimes \mathbf{t_B}^{\mathrm{T}})\mathrm{vec}(\mathbf{R_X}) = -\overrightarrow{\mathbf{t_A}} \tag{22}$$

Eq. 21 and 22 yield following least square problem:

$$\begin{bmatrix} \mathbf{R_A} \otimes \mathbf{I_{3\times3}} - \mathbf{I_{3\times3}} \otimes \mathbf{R_B}^{\mathrm{T}} & \mathbf{0_{9\times3}} \\ \mathbf{I_{3\times3}} \otimes \mathbf{t_B}^{\mathrm{T}} & \mathbf{R_A} - \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathrm{vec}(\mathbf{R_X}) \\ \overrightarrow{\mathbf{t_X}} \end{bmatrix} = \begin{bmatrix} \mathbf{0_{9\times1}} \\ -\overrightarrow{\mathbf{t_A}} \end{bmatrix} \tag{23}$$

Simply, Eq. 23 is represented by Eq. 24.

$$\mathbf{Q_i v_i} = \mathbf{p_i} \tag{24}$$

$\mathbf{Q_i}$ is 12-by-12 matrix, $\mathbf{v_i}$ and $\mathbf{p_i}$ are 12-by-1 vector which has the information of **X**. if the pose datasets are more than 2, Eq. 24 is represented by Eq. 25.

$$\mathbf{Mv} = \mathbf{N} \tag{25}$$

where **M** and **N** are $[\mathbf{Q_1} \quad \cdots \quad \mathbf{Q_n}]^T$, $[\mathbf{p_1} \quad \cdots \quad \mathbf{p_n}]^T$, respectively.

As mentioned before, at least 2 poses are needed for solving **X**. Eq. 25 is solved by using pseudo inverse as Eq. 26.

$$\mathbf{v} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{N} \tag{26}$$

Although **v** has the information of **X**, it is not matrix form; therefore, we need to reconstruct **v** vector into matrix form. This method doesn't have any constraints for rotation part while calculating Eq. 26. Therefore, the rotation part doesn't keep the orthogonal property due to the noise after calculating Eq. 26. The post-processing to make the rotation part to be orthogonal matrix is needed. It is performed by using singular value decomposition (SVD), or Rodrigues function in practice. However, the translation part has still the noise, but there is no way to eliminate properly it. This problem leads to the additional error of **X**. However, the advantage of this method is that small range of angle can be expressed. In the small change of angle, the other rotation representation such as axis-angle and quaternion cannot express small change of angle since the computer do round-processing about small angle. Although this is obviously an advantage for Kronecker product method, it is not good for camera-sensor calibration since small angle is not suitable for ensuring high accuracy of camera-sensor calibration [12].

2.1.2.2 Solution using dual quaternion

Before introducing dual quaternion, we briefly introduce quaternion. The quaternion were first introduced by Hamilton. It is one of the representations for rotation widely used in literature relative to robotics and computer graphics today. The definition of quaternion is as pairs $(q_0, \vec{\mathbf{q}})$, where q_0 is scalar, $\vec{\mathbf{q}}$ is vector which is 3-by-1.

This is written as Eq. 27.

$$q = q_0 + q_i\hat{\mathbf{i}} + q_j\hat{\mathbf{j}} + q_k\hat{\mathbf{k}} \tag{27}$$

or

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_i \\ q_j \\ q_k \end{bmatrix} \tag{28}$$

where q_0 is scalar part of quaternion, $q_i, q_j,$ and q_k are vector part of quaternion, we define them as $\vec{\mathbf{q}}$. Sometimes, we can change the stacking order by stacking firstly not scalar element, but vector elements. However, it doesn't affect the physical meanings. Once we have rotation matrix, 3-by-3 from homogeneous matrix as Eq. 29, the rotation matrix can be converted to quaternion by using Eq. 30-33.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{29}$$

$$q_0 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \tag{30}$$

$$q_i = \frac{r_{32} - r_{23}}{4q_0} \tag{31}$$

$$q_j = \frac{r_{13} - r_{31}}{q_0} \tag{32}$$

$$q_k = \frac{r_{21} - r_{12}}{q_0} \tag{33}$$

The concept of quaternion we mentioned above can be also used for dual quaternion. Dual quaternion is the

concept expanded from quaternion, which is used to represent rigid body displacements. That is, by using dual quaternion, rotation and translation of rigid body can be expressed while quaternion represents only rotation part. This expression encodes the screw motion theory which all rigid body motions can be expressed by using the unique axis (screw axis), the point directing the unique axis from the origin which is orthogonal to the screw axis, rotation angle, and pitch as shown in Fig. 2.4.

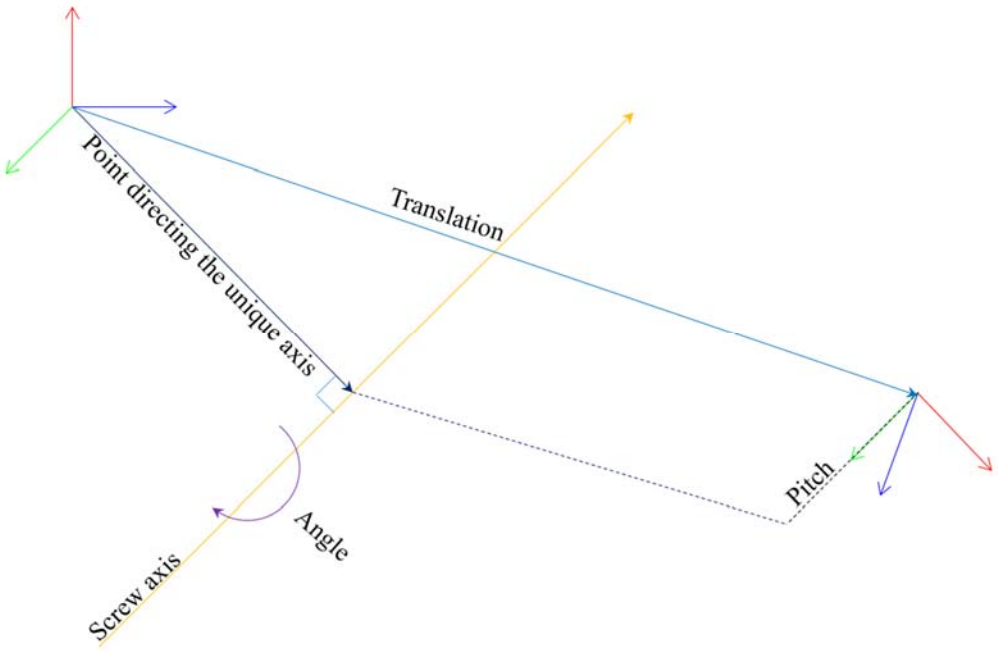


Figure 2.4 Screw theory. All rigid body motions can be expressed by using screw axis, point directing the screw axis, angle, and pitch (6 DOF).

A dual quaternion, $\breve{\mathbf{q}}$ is an ordered pair of quaternion like Eq. 34, which consists of 8 elements.

$$\breve{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q}' \tag{34}$$

where $\mathbf{q}$ is quaternion defined as Eq. 27 or 28, $\mathbf{q}'$ is also quaternion defined as Eq. 35, and ε is a dual unit in which $\varepsilon^2 = 0$. However, $\mathbf{q}$ is unit quaternion which means the norm value is one while $\mathbf{q}'$ is not unit quaternion.

$$\mathbf{q}' = -\frac{\vec{\mathbf{q}}^{\mathrm{T}}\vec{\mathbf{t}}}{2} + \frac{q_0\vec{\mathbf{t}}}{2}\vec{\mathbf{t}} \times \vec{\mathbf{q}} \tag{35}$$

where q_0 is a scalar of unit quaternion, $\vec{\mathbf{q}}$ is a vector part of the unit quaternion, and $\vec{\mathbf{t}}$ is translation in homogeneous transformation matrix. Therefore, if the homogenous transformation matrix is known, it can be converted

from 4-by-4 matrix form to dual quaternion form by using Eq. 30-35. In order to solve **AX** = **XB** problem by using dual quaternion, we have to first convert transformation matrix to dual quaternion like Eq. 36.

$$\breve{\mathbf{q}}_{\mathbf{A}}\breve{\mathbf{q}}_{\mathbf{X}} = \breve{\mathbf{q}}_{\mathbf{X}}\breve{\mathbf{q}}_{\mathbf{B}} \tag{36}$$

Then,

$$\breve{\mathbf{q}}_{\mathbf{A}}\breve{\mathbf{q}}_{\mathbf{X}} - \breve{\mathbf{q}}_{\mathbf{X}}\breve{\mathbf{q}}_{\mathbf{B}} = 0 \tag{37}$$

By using the definition of dual quaternion,

$$(\mathbf{q}_{\mathbf{A}} + \varepsilon\mathbf{q}'_{\mathbf{A}})(\mathbf{q}_{\mathbf{X}} + \varepsilon\mathbf{q}'_{\mathbf{X}}) - (\mathbf{q}_{\mathbf{X}} + \varepsilon\mathbf{q}'_{\mathbf{X}})(\mathbf{q}_{\mathbf{B}} + \varepsilon\mathbf{q}'_{\mathbf{B}}) = 0 \tag{38}$$

If developing Eq. 34, we can separate it into Eq. 39, and 40, which are rotation part and translation part, respectively.

$$\mathbf{q}_{\mathbf{A}}\mathbf{q}_{\mathbf{X}} - \mathbf{q}_{\mathbf{X}}\mathbf{q}_{\mathbf{B}} = 0 \tag{39}$$

$$\mathbf{q}_{\mathbf{A}}\mathbf{q}'_{\mathbf{X}} + \mathbf{q}'_{\mathbf{A}}\mathbf{q}_{\mathbf{X}} - \mathbf{q}_{\mathbf{X}}\mathbf{q}'_{\mathbf{B}} - \mathbf{q}'_{\mathbf{X}}\mathbf{q}_{\mathbf{B}} = 0 \tag{40}$$

Eq. 39 and 40 can be expressed by using matrix form, then we have,

$$\widetilde{\mathbf{q}}_{\mathbf{A}}\mathbf{q}_{\mathbf{X}} - \underline{\mathbf{q}}_{\mathbf{B}}\mathbf{q}_{\mathbf{X}} = 0 \tag{41}$$

$$\widetilde{\mathbf{q}}_{\mathbf{A}}\mathbf{q}'_{\mathbf{X}} + \widetilde{\mathbf{q}}'_{\mathbf{A}}\mathbf{q}_{\mathbf{X}} - \underline{\mathbf{q}}'_{\mathbf{B}}\mathbf{q}_{\mathbf{X}} - \underline{\mathbf{q}}_{\mathbf{B}}\mathbf{q}'_{\mathbf{X}} = 0 \tag{42}$$

where $\widetilde{\mathbf{q}}_{\mathbf{A}}$ and $\widetilde{\mathbf{q}}'_{\mathbf{A}} = \begin{bmatrix} q_0 & -\vec{\mathbf{q}}^{\mathrm{T}} \\ \vec{\mathbf{q}} & q_0\mathbf{I} + \mathrm{skew}(\vec{\mathbf{q}}) \end{bmatrix}$, $\underline{\mathbf{q}}_{\mathbf{B}}$ and $\underline{\mathbf{q}}'_{\mathbf{B}} = \begin{bmatrix} q_0 & -\vec{\mathbf{q}}^{\mathrm{T}} \\ \vec{\mathbf{q}} & q_0\mathbf{I} - \mathrm{skew}(\vec{\mathbf{q}}) \end{bmatrix}$.

where $\mathrm{skew}(\vec{\mathbf{q}}) = \begin{bmatrix} 0 & -q_k & q_j \\ q_k & 0 & -q_i \\ -q_j & q_i & 0 \end{bmatrix}$ called “anti-symmetric matrix” or “skew-symmetric matrix”.

If the first order to stack the elements is not scalar value of quaternion but vector,

$\widetilde{\mathbf{q}}_{\mathbf{A}}$ and $\widetilde{\mathbf{q}}'_{\mathbf{A}} = \begin{bmatrix} q_0\mathbf{I} + \mathrm{skew}(\vec{\mathbf{q}}) & \vec{\mathbf{q}} \\ -\vec{\mathbf{q}}^{\mathrm{T}} & q_0 \end{bmatrix}$, $\underline{\mathbf{q}}_{\mathbf{B}}$ and $\underline{\mathbf{q}}'_{\mathbf{B}} = \begin{bmatrix} q_0\mathbf{I} - \mathrm{skew}(\vec{\mathbf{q}}) & \vec{\mathbf{q}} \\ -\vec{\mathbf{q}}^{\mathrm{T}} & q_0 \end{bmatrix}$.

Eq. 41 and 42 yield least square problem as following:

$$\begin{bmatrix} \widetilde{\mathbf{q}}_{\mathbf{A}} - \underline{\mathbf{q}}_{\mathbf{B}} & \mathbf{0}_{4\times4} \\ \widetilde{\mathbf{q}}'_{\mathbf{A}} - \underline{\mathbf{q}}'_{\mathbf{B}} & \widetilde{\mathbf{q}}_{\mathbf{A}} - \underline{\mathbf{q}}_{\mathbf{B}} \end{bmatrix} \begin{bmatrix} \mathbf{q}_{\mathbf{X}} \\ \mathbf{q}'_{\mathbf{X}} \end{bmatrix} = 0 \tag{43}$$

Supposed that we have n-poses. Then,

$$\mathbf{T} = [\mathbf{S_1}, \mathbf{S_2}, \cdots, \mathbf{S_n}]^{\mathbf{T}} \tag{44}$$

where $\mathbf{S_i}$ is $\begin{bmatrix} \widetilde{\mathbf{q}}_{\mathbf{A_i}} - \underline{\mathbf{q}}_{\mathbf{B_i}} & \mathbf{0}_{4\times4} \\ \widetilde{\mathbf{q}}'_{\mathbf{A_i}} - \underline{\mathbf{q}}'_{\mathbf{B_i}} & \widetilde{\mathbf{q}}_{\mathbf{A_i}} - \underline{\mathbf{q}}_{\mathbf{B_i}} \end{bmatrix}$.

By taking SVD to Eq. 44, we can obtain **V** matrix consisting of right singular vectors. After taking 7th and 8th column vectors in **V** matrix, proper lambda values to obtain final solution which are a linear combination of 7th and 8th column vectors are found by using discriminant [15]. Comparing equation we present here with origin equation, the only difference is to use scalar part of quaternion. In [15], since **A** and **B** are similar matrix, their angle and pitch are the same as each other when the camera is moved; therefore, they removed the scalar part. In noise free case, they proved that eliminating scalar part don't affect the results. However, in practice, due to the noise, it is not equal to each other. Therefore, we have to consider the scalar part of quaternion. Once $\breve{\mathbf{q}}_{\mathbf{X}}$ is determined from linear combination explained from [15], it is reconstructed to 3-by-3 rotation matrix and 3-by-1 translation by using Eq. 45 and 46, respectively.

$$\mathbf{R_{3\times3}} = \begin{bmatrix} 1-2q_j^2-2q_k^2 & 2(q_iq_j-q_kq_0) & 2(q_iq_k+q_jq_0) \\ 2(q_iq_j+q_kq_0) & 1-2q_i^2-2q_k^2 & 2(q_jq_k-q_iq_0) \\ 2(q_iq_k-q_jq_0) & 2(q_jq_k+q_iq_0) & 1-2q_i^2-2q_j^2 \end{bmatrix} \tag{45}$$

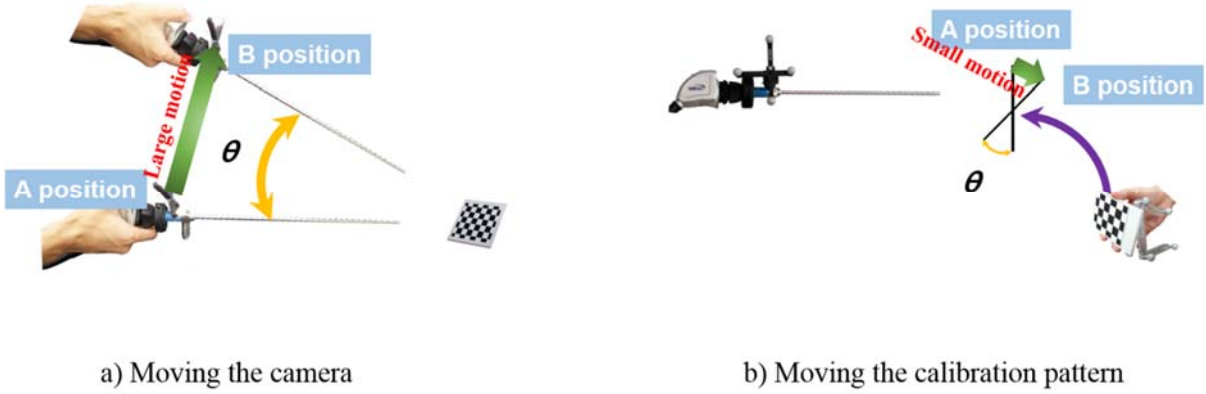
$$\vec{\mathbf{t}}_{\mathbf{3\times1}} = 2\mathbf{q}'\overline{\mathbf{q}} \tag{46}$$

where $\overline{\mathbf{q}}$ is the conjugate of $\mathbf{q}$ defined as Eq. 47.

$$\overline{\mathbf{q}} = \begin{bmatrix} q_0 \\ -q_i \\ -q_j \\ -q_k \end{bmatrix} \tag{47}$$

2.1.3 **AX** = **BYC** problem

As mentioned before, one of the important things to solve **AX** = **XB** problem is to keep the relationship between optical tracking system and the calibration pattern, **Y** to be constant. It means we have to move only the camera to obtain the correct solution; otherwise, the solution is not converged. Moving the camera is much affected from the spatial error in medical application such as AR. Therefore, in order to reduce the spatial error of optical tracking system, moving the calibration pattern could be better choice than moving the camera since it ensures the convenient manufacture and the small range of motion in performing camera-sensor calibration. As shown in Fig. 2.5, comparing moving the camera, moving the calibration pattern requires the small motion to make the orientation of the camera.



a) Moving the camera b) Moving the calibration pattern

Figure 2.5 Comparison between the range of motion of each method. When making the same orientation in each movement, moving the calibration pattern (b) requires smaller range of motion than moving the camera (a).

Eventually, it results in higher accuracy than moving the camera. In order to move the calibration pattern instead of the camera, an additional marker has to be mounted on the calibration pattern. The location of marker to be mounted on the calibration pattern is not restricted by structures since the calibration pattern is not used on the operation while the location of marker mounted on the clinical camera has to be seriously considered since it is used on the operation. By mounting the additional marker on the calibration pattern at the proper location, the unknown parameter increases from one to two, which are the camera-sensor relationship and the relationship between the coordinate system of the calibration pattern and the marker mounted on the calibration pattern. We

will call those relationships to **X** and **Y**, respectively in this paper. Therefore, we can build **AX** = **BYC** problem as shown in Fig. 2.6. The effects of mounting marker on the calibration pattern are to keep **Y** to be constant, and to be tracked from optical tracking system. In **AX** = **XB** or **AX** = **YB** problem, **Y** is variable if we move the calibration pattern.

The notations of **AX** = **BYC** is slightly different from the conventional method, **AX** = **XB**. **A** represents the transformation matrix from optical tracking system to the passive marker mounted on the camera, **B** represents that from optical tracking system to the passive marker mounted on the calibration pattern, and **C** represents that from the calibration pattern to the camera lens. **X** is the same as **X** of the **AX** = **XB** problem, **Y** is the relationship between the passive marker mounted on the calibration pattern and the calibration pattern.

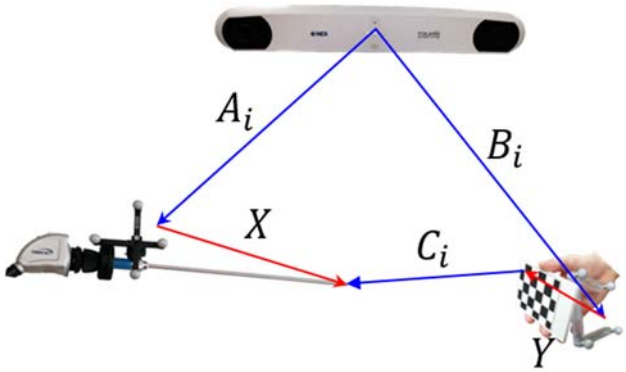


Figure 2.6 AX = BYC problem. By mounting an additional markers on a calibration pattern, the calibration pattern can be moved. Although unknown parameters increases from one to two, the effect of the spacial error from optical tracking system could be reduced.

2.1.3.1 Procedures for solution of **AX** = **BYC**

Once **AX** = **BYC** problem is built from Fig. 2.6, there are three procedures to solve it. Two of three procedures use **Y**, rest procedure don't use **Y**. Simply, first procedure is to solve **X** by using **Y** firstly calculated from **BY** = **YC** problem. Second procedure is to solve **X** and **Y** at the same time. Third procedure is to solve only **X** by making **DX** = **XF** equation.

First procedure for solving AX = BYC problem: First step of procedure 1 is to acquire n-poses by moving the

calibration pattern as shown in Fig. 2.7. Second step is to make **BY** = **YC** equation by using $\mathbf{B_i}, \mathbf{C_i}$ as shown in Fig. 2.8. After building **BY** = **YC** equation, **Y** can be solved by using Kronecker product or dual quaternion method. In this time, we used Kronecker product method to solve **AX = BYC** in all procedures to avoid unbiased result.

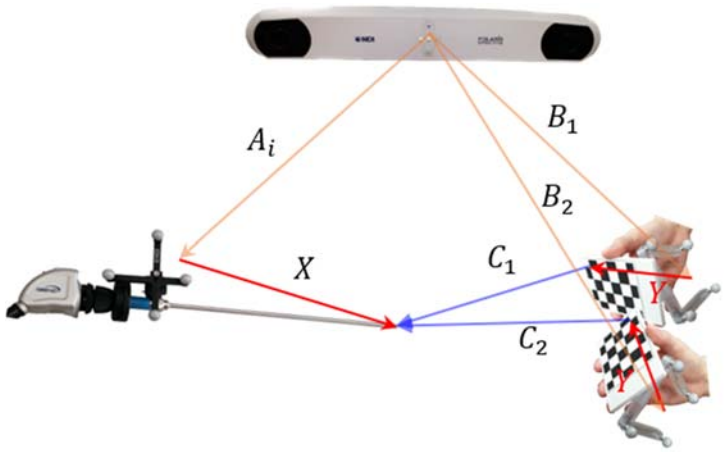


Figure 2.7 First step of procedure 1. From 1st step, n-poses are acquired. The calibration pattern was moved in all procedures.

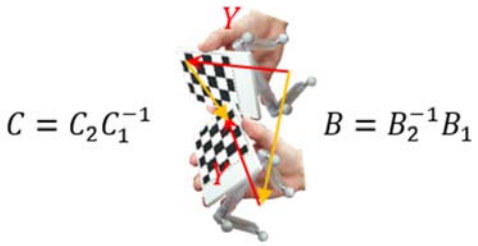


Figure 2.8 Second step of procedure 1. 2nd step yields **BY** = **YC** equation.

Third step is to solve **X** by using **Y** solved in step 2 as shown in Fig. 2.9. In order to reduce the effect of the noise from optical tracking system, the average method such as Eq. 48 is used.

$$\mathbf{X_{final}} = \frac{1}{n}\sum_{i}^{n} \mathbf{A_i^{-1} B_i Y_i C_i} \tag{48}$$

From the observation of Eq. 48, procedure 1 has two shortages to make it more sensitive to the noise from optical tracking system. One is many multiplication times. That is, procedure 1 is not compact. The other is that procedure 1 is depending on the result of **Y** solved in step 2. Therefore, the error of **Y** affects the result of **X**.

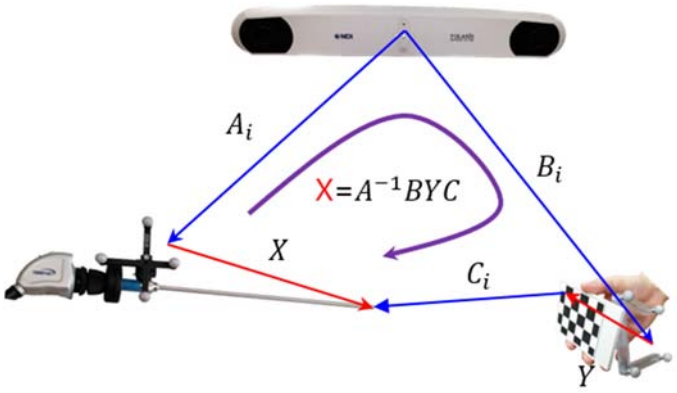


Figure 2.9 Third step of procedure 1. 3rd step is to solve **X** by using **Y** calculated in step 2.

Second procedure for solving AX = BYC problem: This procedure is to solve **X** and **Y** simultaneously. Since **X** and **Y** are solved at the same time, **X** is not depending on **Y**. However, in least square problem, the existence of **Y** affects the result of **X** during solving problem. In order to achieve the solution by using procedure 2, we have to follow next 3 steps. First step of procedure 2 is to acquire n-poses in the same manner with procedure 1. Second step is to make **DX** = **YC** equation which is the same form as **AX** = **YB** problem as shown in Fig. 1.2 (b). Fig. 2.10 shows second step of procedure 2. Although this equation is used for solving **AX** = **BYC** problem, the reason why this equation are distinguished from our equation is that the moving target is the camera in **AX** = **YB** problem while in **AX** = **BYC** problem, the moving target is the calibration pattern. Final step is to calculate **DX** = **YC** equation by using the method we will explain next sub-chapter more details. The method also uses Kronecker product, or dual quaternion as **AX** = **XB** problem.

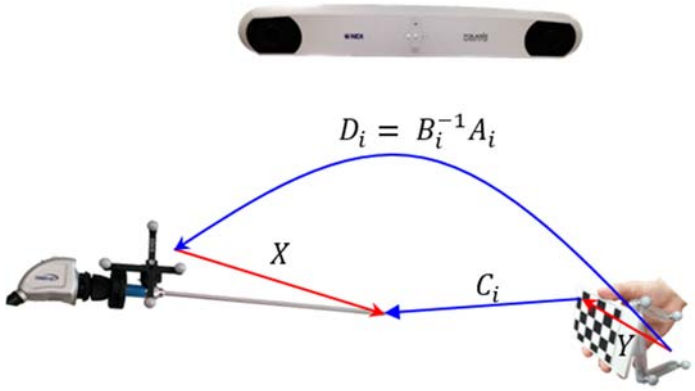


Figure 2.10 Second step of procedure 2. 2nd step is to build **DX = YC** equation by multiplying both sides of **AX = BYC** problem with the inverse of **B**.

Third procedure for solving AX = BYC problem: Third procedure is not relative to **Y** as mentioned before. First step is to acquire n-poses, second step is to make **D** and **F** as shown in Fig. 2.11.

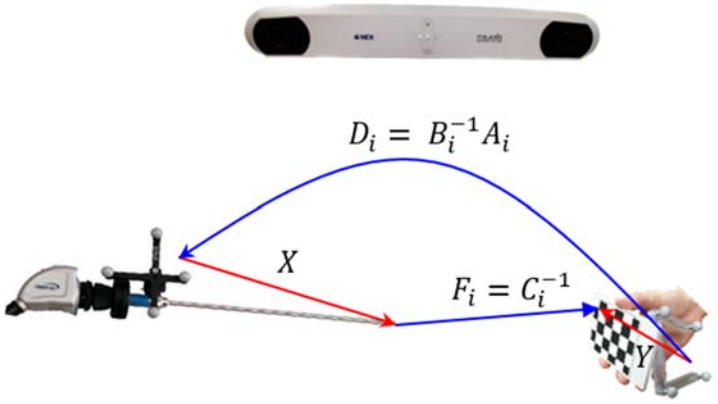


Figure 2.11 Second step of procedure 3. 2nd step is to make **D** and **F**. Note that **F** is not the same direction as **C** in procedure 2. Its direction is the inverse of **C**.

Comparing the case in Fig. 2.10 with the case in Fig. 2.3 which is the normal case of **AX** = **XB** in the robotic application, **D** and **F** in procedure 3 is corresponding to **A** and **B** in the normal case, respectively. In addition, **Y** in procedure 3 is corresponding to the relationship between optical tracking system and the coordinate system of the calibration pattern, which is constant. Therefore, this is the same as **AX** = **XB** derived from the case of moving the camera.

By using Eq. 7-13, it can be mathematically proved that procedure 3 is the same as the conventional **AX** = **XB** problem except that moving target is not the camera but the calibration pattern. From procedure 3 as shown in Fig. 2.12, following two equations can be derived.

$$\mathbf{XF_1} = \mathbf{D_1^{-1}Y} \tag{49}$$

$$\mathbf{XF_2} = \mathbf{D_2^{-1}Y} \tag{50}$$

Those equations can be written as Eq. 51 by replacing **Y** of Eq. 50 with $\mathbf{D_1XF_1}$ obtained from Eq. 49

$$\mathbf{XF_2F_1^{-1}} = \mathbf{D_2^{-1}D_1X} \tag{51}$$

Therefore,

$$\mathbf{AX} = \mathbf{XB} \tag{52}$$

where $\mathbf{A} = \mathbf{D_2^{-1}D_1}$, $\mathbf{B} = \mathbf{F_2F_1^{-1}}$ which are the same as multiplication order derived from Eq. 7-13.

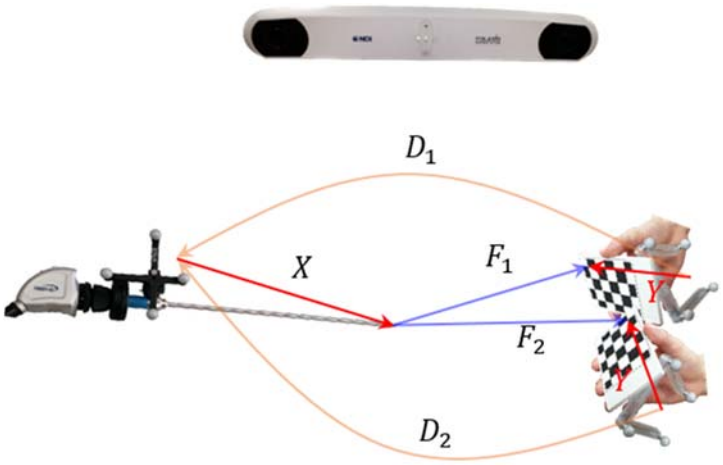


Figure 2.12 Third step of procedure 3. By inversing **C**, **AX** = **BYC** problem can be re-built to **DX** = **XF** problem without **Y**; Therefore, it can be easily solved as the conventional **AX** = **XB** problem.

Third step is to build **DX** = **XF** equation and solve it by using Kronecker product. There is no way to know if the target of movement is the camera or the calibration pattern until user gives an additional data to the computer. The only information the computer can know is to use the information of **A**, **B** obtained from optical tracking system and the camera. Therefore, if the direction and the order to multiply two poses for making **D** and **F** is fulfilled as the conventional **AX** = **XB** method as shown in Fig. 2.3, then, the computer recognizes the moving target is not the calibration pattern but the camera although the calibration pattern is moved.

2.1.3.2 Comparison between procedures

Through pervious sub-chapter, three procedures to solve **AX** = **BYC** problem were presented. In this sub-chapter, we are going to compare procedures by performing synthetic experiments in terms of the accuracy. In order to perform synthetic experiments, we developed the simulation tool based MATLAB (R2013a, MathWorks, USA) as shown in Fig. 2.13. In the simulation tool, it is possible to control the number of stations (poses), the noise level to be added into poses, and to select procedures.

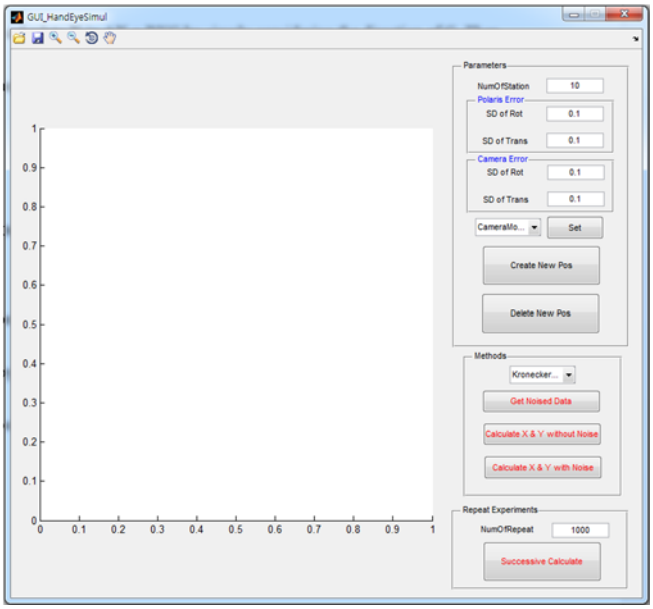


Figure 2.13 Simulation tool for camera-sensor calibration.

In addition, this program provides the function to compare procedures with the iteration as times user set. By clicking the 'Successive Calculate' button, the result of synthetic experiments are shown by the graph. We supposed that the spatial error of optical tracking system and that of the camera have the normal distribution (Gaussian normal distribution) and the magnitude of their error increases with the distance in this simulation. Actually, the distribution of optical tracker follows Maxwell distribution. However, since the goal of experiments are to show which procedure is less sensitive to the noise, aforementioned assumption is not critical problem. For example, if assuming that arbitrary pose of n-poses obtained from the program are as Eq. 53, then,

$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{53}$$

The rotation matrix, **R** in **T** can be represented by 3-by-1 vector, $\mathbf{R_{XYZ}}$ by using Rodrigues formula defined as Eq. 54-56.

$$\theta = \text{acos}\left(\frac{\text{tr}(\mathbf{R}) - 1}{2}\right) \tag{54}$$

$$\vec{\mathbf{k}} = \frac{1}{2\sin(\theta)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \tag{55}$$

$$\mathbf{R_{XYZ}} = \theta\vec{\mathbf{k}} \tag{56}$$

The standard deviation to generate the noise is determined by following equation if the standard deviation is corresponding to 0.1% of the rotation.

$$\begin{bmatrix} r_{sd\_X} \\ r_{sd\_Y} \\ r_{sd\_Z} \end{bmatrix} = \frac{1}{10}\begin{bmatrix} r_X \\ r_Y \\ r_Z \end{bmatrix} \tag{57}$$

Then, the random noise are generated by using standard deviation determined from Eq. 57.

$$r_{noise\_X} = r_{sd\_X} * \text{randn}(1) \tag{58}$$

$$r_{noise\_Y} = r_{sd\_Y} * \text{randn}(1) \tag{59}$$

$$r_{noise\_Z} = r_{sd\_Z} * \text{randn}(1) \tag{60}$$

where randn is the random function provided from MATLAB, which returns a random value drawn from the normal distribution. For example, for generating a value, v from a normal distribution with mean and standard deviation, we have to type MATLAB code as following: $v = \text{mean} + \text{standard deviation} * \text{randn}(1)$. Therefore, the final 3-by-1 rotation vector with the noise is as Eq. 61.

$$\mathbf{R_{sd\_XYZ}} = [r_X + r_{noise\_X} \quad r_Y + r_{noise\_Y} \quad r_Z + r_{noise\_Z}]^T \tag{61}$$

Simply, we can reconstruct 3-by-1 vector to 3-by-3 matrix by using Eq. 62-64.

$$\vec{\mathbf{k}} = \frac{\mathbf{R_{sd\_XYZ}}}{\theta} \tag{62}$$

$$\theta = \text{norm}(\mathbf{R_{sd\_XYZ}}) \tag{63}$$

$$\mathbf{R_{sd}} = \mathbf{I_{3\times3}}\cos(\theta) + \text{skew}(\vec{\mathbf{k}})\sin(\theta) + (1 - \cos(\theta))(\vec{\mathbf{k}}\vec{\mathbf{k}}^T) \tag{64}$$

where, norm is the magnitude of a vector.

In common with the method above, the translation with noise can be created as Eq. 65.

$$\vec{\mathbf{t}}_{\mathbf{sd}} = [t_X + t_{noise\_X} \quad t_Y + t_{noise\_Y} \quad t_Z + t_{noise\_Z}]^T \tag{65}$$

We performed synthetic experiment with 1000 iterations to compare procedures each noise level (0.1 – 1.0

with 0.1 increment). The results of synthetic experiment is shown in Fig. 2.14-15. In the rotation part, the vertical and the horizontal axis mean the magnitude of the error represented by radian unit and the noise level (0.1 – 1.0), respectively. For example, in noise level 1, we gave standard deviation corresponding to 0.1% of orientation and translation to the simulator for the poses with the noise. Procedure 2 is least sensitive to the noise. As the rotation result, translation part also shows the similar results. In the translation part, vertical axis means the magnitude of error represented by millimeter unit. Considering the number of multiplication, these results can be easily expected. The more the number of multiplication is, the more the sensitivity of noise increases. In addition, comparing between procedure 2 and 3, procedure 3 has more multiplication time than procedure 2 since $\mathbf{D_i}$ is from $\mathbf{A_i}$ and $\mathbf{B_i}$ (2 multiplication time) and **D** and **F** are from two poses (2 multiplication for each matrix). That is, for making **DX** = **XF** in procedure 3, four multiplication time are required. However, for making **DX** = **YC** in procedure 2, two multiplication time are only required, which is from making **D** matrix. Next section, we are going to explain how to solve the final step in procedure 2 more details.

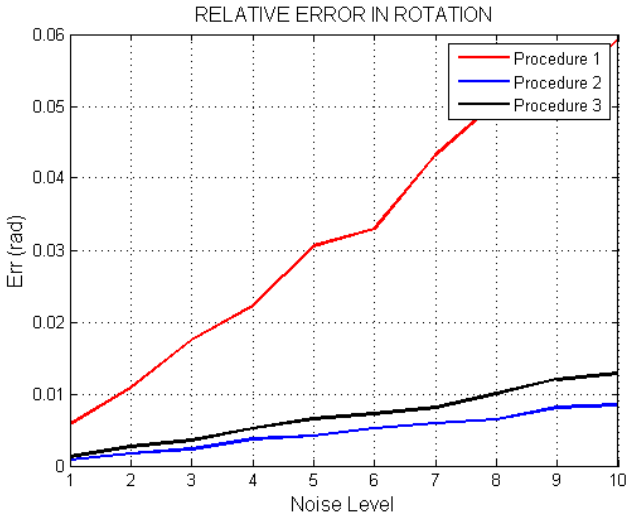


Figure 2.14 Rotation error of procedures. Procedure 2 is least sensitive to the noise since procedure 2 has least muliplication time in building least square problem.

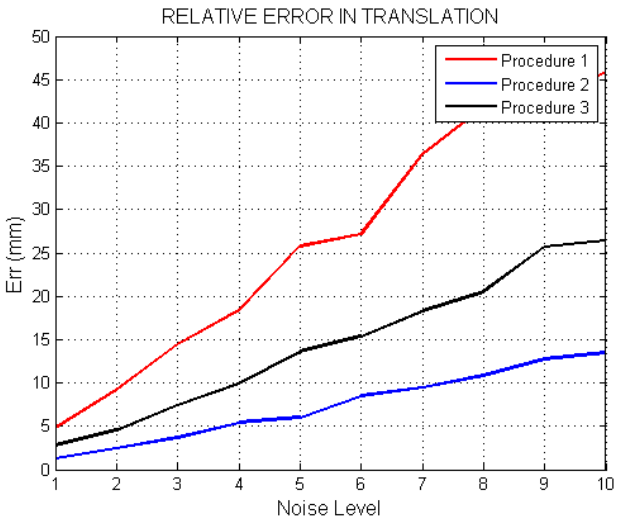


Figure 2.15 Translation error of procedures. As a result of rotation part, in the result of translation, procedure 2 is also least sensitive to the noise.

2.1.3.3 Solution using Kronecker product

In the previous section, three kinds of procedures to solve **AX** = **BYC** problem are compared. The strongest procedure from the noise is second procedure since it is mathematically compact (procedure 2 has least multiplication time). In this section, we will explain briefly how step 3 in procedure 2 is solved more details. First, Kronecker product can be used for solving **DX** = **YC** equation obtained from step 2 in procedure 2. Kronecker product explained in section 2.1.2.1 is relative to **AX** = **XB** problem which has only one unknown parameter, **X**. The method using Kronecker product to solve **DX** = **YC** equation begins from separating rotation part and translation part as the method to solve **AX** = **XB** problem.

$$\mathbf{R_D R_X} = \mathbf{R_Y R_C} \tag{66}$$

$$\mathbf{R_D}\vec{\mathbf{t}}_\mathbf{X} + \vec{\mathbf{t}}_\mathbf{D} = \mathbf{R_Y}\vec{\mathbf{t}}_\mathbf{C} + \vec{\mathbf{t}}_\mathbf{Y} \tag{67}$$

Then,

$$\mathbf{R_D R_X} - \mathbf{R_Y R_C} = 0 \tag{68}$$

$$-\mathbf{R_Y}\vec{\mathbf{t}}_\mathbf{C} + \mathbf{R_D}\vec{\mathbf{t}}_\mathbf{X} - \vec{\mathbf{t}}_\mathbf{Y} = -\vec{\mathbf{t}}_\mathbf{D} \tag{69}$$

By using Kronecker product and vectorization described in section 2.1.2.1, Eq. 68 and 69 can be written as Eq.

70 and 71, respectively.

$$[\mathbf{R_D} \otimes \mathbf{I_{3\times3}}][\mathrm{vec}(\mathbf{R_X})] - [\mathbf{I_{3\times3}} \otimes \mathbf{R_C^T}][\mathrm{vec}(\mathbf{R_Y})] = 0 \tag{70}$$

$$-[\mathbf{I_{3\times3}} \otimes \vec{\mathbf{t}}_\mathbf{C}^{\mathrm{T}}][\mathrm{vec}(\mathbf{R_Y})] + \mathbf{R_D}\vec{\mathbf{t}}_\mathbf{X} - \mathbf{I_{3\times3}}\vec{\mathbf{t}}_\mathbf{Y} = -\vec{\mathbf{t}}_\mathbf{D} \tag{71}$$

Therefore,

$$\begin{bmatrix} \mathbf{R_D} \otimes \mathbf{I_{3\times3}} & -\mathbf{I_{3\times3}} \otimes \mathbf{R_C^T} & \mathbf{0_{9\times3}} & \mathbf{0_{9\times3}} \\ \mathbf{0_{3\times9}} & -\mathbf{I_{3\times3}} \otimes \mathbf{t_C^T} & \mathbf{R_D} & -\mathbf{I_{3\times3}} \end{bmatrix} \begin{bmatrix} \mathrm{vec}(\mathbf{R_X}) \\ \mathrm{vec}(\mathbf{R_Y}) \\ \vec{\mathbf{t}}_\mathbf{X} \\ \vec{\mathbf{t}}_\mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{0_{9\times1}} \\ -\vec{\mathbf{t}}_\mathbf{D} \end{bmatrix} \tag{72}$$

This equation is **Qv** = **p**. the dimension of **Q**, **v**, and **p** has $12 \times 24, 24 \times 1$, and 12×1, respectively. **Q** and **p** are the notations for 2 poses. If the number of poses is larger than 2, Eq. 72 can be written as Eq. 73.

$$\mathbf{Mv} = \mathbf{N} \tag{73}$$

Next step to solve Eq. 73 is the same as the step described in section 2.1.2.1.

2.1.3.4 Solution using dual quaternion

The definition of dual quaternion was described in section 2.1.2.2. In common with solving **AX** = **XB** problem by using dual quaternion, first step to solve **DX** = **YC** problem by using dual quaternion is to convert the homogeneous transformation matrix to dual quaternion. Though first step, we can obtain Eq. 74.

$$\breve{\mathbf{q}}_\mathbf{D}\breve{\mathbf{q}}_\mathbf{X} = \breve{\mathbf{q}}_\mathbf{Y}\breve{\mathbf{q}}_\mathbf{C} \tag{74}$$

By using the definition of dual quaternion, Eq. 74 can be written as Eq. 75.

$$(\mathbf{q_D} + \varepsilon\mathbf{q'_D})(\mathbf{q_X} + \varepsilon\mathbf{q'_X}) - (\mathbf{q_Y} + \varepsilon\mathbf{q'_Y})(\mathbf{q_C} + \varepsilon\mathbf{q'_C}) = 0 \tag{75}$$

Then, Eq. 75 can be separated to rotation part and translation part by developing and grouping terms.

$$\mathbf{q_D}\mathbf{q_X} - \mathbf{q_Y}\mathbf{q_C} = 0 \tag{76}$$

$$\mathbf{q_D}\mathbf{q'_X} + \mathbf{q'_D}\mathbf{q_X} - \mathbf{q_Y}\mathbf{q'_C} - \mathbf{q'_Y}\mathbf{q_C} = 0 \tag{77}$$

If Eq. 76 and 77 are represented by the matrix form, we can change the multiplication order as Eq. 78 and 79,

respectively.

$$\tilde{\mathbf{q}}_{\mathbf{D}}\mathbf{q}_{\mathbf{X}} - \underline{\mathbf{q}}_{\mathbf{C}}\mathbf{q}_{\mathbf{Y}} = 0 \tag{78}$$

$$\tilde{\mathbf{q}}_{\mathbf{D}}\mathbf{q}'_{\mathbf{X}} + \tilde{\mathbf{q}}'_{\mathbf{D}}\mathbf{q}_{\mathbf{X}} - \underline{\mathbf{q}}'_{\mathbf{C}}\mathbf{q}_{\mathbf{Y}} - \underline{\mathbf{q}}_{\mathbf{C}}\mathbf{q}'_{\mathbf{Y}} = 0 \tag{79}$$

The array of elements in matrix is described in section 2.1.2.2.

Therefore,

$$\begin{bmatrix} \tilde{\mathbf{q}}_{\mathbf{D}} & \mathbf{0}_{4\times4} & -\underline{\mathbf{q}}_{\mathbf{C}} & \mathbf{0}_{4\times4} \\ \tilde{\mathbf{q}}'_{\mathbf{D}} & \tilde{\mathbf{q}}_{\mathbf{D}} & -\underline{\mathbf{q}}'_{\mathbf{C}} & -\underline{\mathbf{q}}_{\mathbf{C}} \end{bmatrix} \begin{bmatrix} \mathbf{q}_{\mathbf{X}} \\ \mathbf{q}'_{\mathbf{X}} \\ \mathbf{q}_{\mathbf{Y}} \\ \mathbf{q}'_{\mathbf{Y}} \end{bmatrix} = [\mathbf{0}_{8\times1}] \tag{80}$$

Next step to solve Eq. 80 is the same as the step described in section 2.1.2.2.

2.1.3.4.1 Problem of dual quaternion

The dual quaternion is the strong representation to express rigid motion in 3-D. However, in solving **DX** = **YC** equation, it has a problem which has to select the correct direction of rotation axis. When cosine rotation matrix is converted to axis-angle or quaternion representation, they have always two solutions, $< \vec{\mathbf{k}}, \theta >$ and $< -\vec{\mathbf{k}}, -\theta >$. Dual quaternion is based on quaternion; therefore, it has also pair of solution. These pairs have physically the same meaning each other although their sign are opposite to each other as shown in Fig. 2.15.

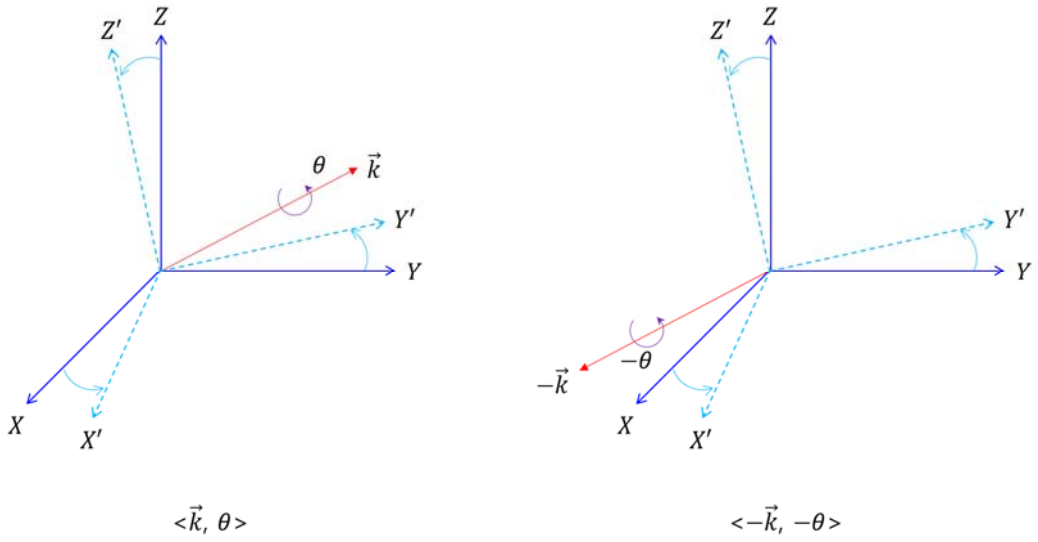


Figure 2.16 Pair of solution for quaternion and axis-angle representation. Although two solution have opposite sign to each other, their physical meaning is the same.

Therefore, in **DX** = **YC** equation, Eq. 78 and 79 may be not fulfilled according to the sign of quaternion. In order to select the correct sign of pair, we used **X** and **Y** obtained from Kronecker product method. Since the conversion such as cosine matrix to axis-angle or quaternion is not used in Kronecker product method, this method can work very well in **DX** = **YC** equation. Although the solution is used for selecting the correct sign of pair, they don't affect directly the result of dual quaternion. After unknown parameters are known, Eq. 78 and 79 can provide the solutions. In ideal case, these solutions are zero; however, due to the noise from optical tracking system and the camera, these values are not zero (almost zero). In the case of the incorrect sign of pair, these solutions are not almost zero but big values. Following equations express when the sign is opposite to Eq. 78 and 79:

$$\tilde{\mathbf{q}}_{\mathbf{D}}\mathbf{q}_{\mathbf{X}} + \underline{\mathbf{q}}_{\mathbf{C}}\mathbf{q}_{\mathbf{Y}} = 0 \tag{81}$$

$$\tilde{\mathbf{q}}_{\mathbf{D}}\mathbf{q}'_{\mathbf{X}} + \tilde{\mathbf{q}}'_{\mathbf{D}}\mathbf{q}_{\mathbf{X}} + \underline{\mathbf{q}}'_{\mathbf{C}}\mathbf{q}_{\mathbf{Y}} + \underline{\mathbf{q}}_{\mathbf{C}}\mathbf{q}'_{\mathbf{Y}} = 0 \tag{82}$$

The algorithm we made is to select the correct sign from Eq. 78, 79, 81, and 82. If the solutions from Eq. 78, 79, 81, and 82 are E1, E2, E3, and E4, respectively, the algorithm compares between E1 and E2 to know which one is lower. Once lower value of E1 and E2 is selected, corresponding equations are used for solving **AX** = **BYC** problem.

2.2 Evaluation of the proposed method

Until now, we presented three procedures to solve **AX** = **BYC** problem as well as the methods such as dual quaternion and Kronecker product to get a final solution. Also, we described the methods to solve **AX** = **XB** problem, the conventional method. In this section, we are going to explain how to evaluate the proposed method in terms of the accuracy. To show our method is more accurate and stable than the conventional method in any environment in performing camera-sensor calibration, camera-sensor calibration was performed in three layouts as shown in Fig. 2.17. The layouts were selected considering the spatial error of optical tracking system.

Layout 1 is the case less error is expected since both the camera and the calibration pattern can be close to the optical tracking system. As mentioned before, the optical tracking system has the spatial error which increases with the distance. Therefore, the layout 1 has least spatial error of layouts. We cannot expect exactly which layout 2 or layout 3 is most error case since two layout is very similar case each other but those layouts are just different location of the camera or the calibration pattern each other. However, the layout 2 and 3 are the case more error is expected than others since one of the camera and the calibration pattern can be close to the optical tracking system, but the other cannot be close to the optical tracking system and is always far from the optical tracking system.

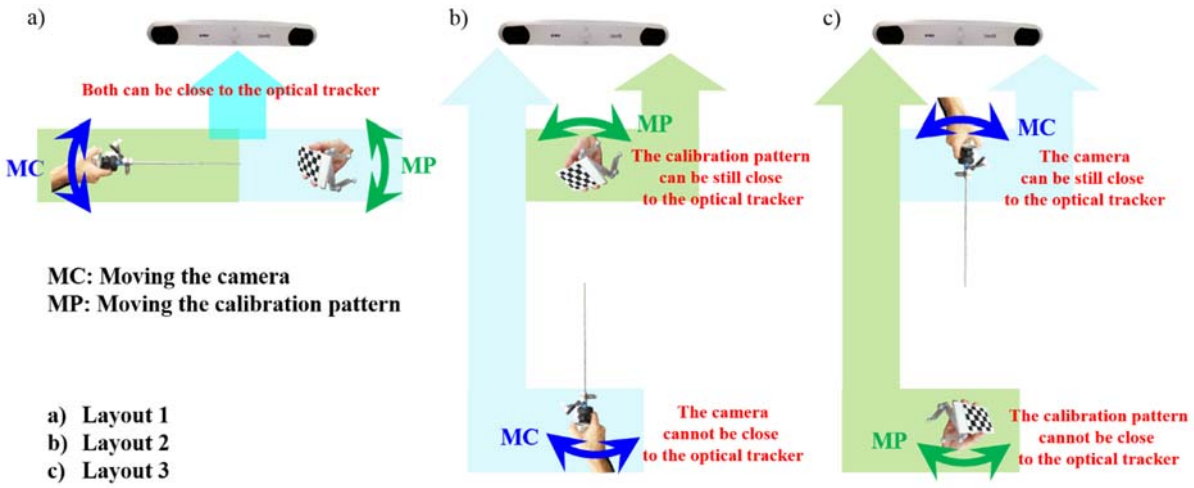


Figure 2.17 Three layouts for comparing both method in terms of accuracy. Layout 1 is for the least error case of three layout.

2.2.1 Experimental setup

Experiments to evaluate the proposed method were conducted with an optical tracker system (Polaris Spectra, NDI, Canada) to track the position and the orientation of the passive markers mounted on the endoscopes and the calibration pattern. In addition, we used an endoscopic system (1188HD, STRYKER, USA) to capture the calibration pattern image, and the calibration pattern with dimensions of 5×4 which has 15×15 mm in each square. The image resolutions and the frame rate we used were 720×576 pixels and 30 frame per second (fps),

respectively. In order to acquire image and pose data from optical tracking system and the camera, the data acquisition program we made was used with aforementioned systems. We will explain the function of the program later. In the case of the conventional method, the number of experiments was 30, which means that we had 30 times to perform camera-sensor calibration, and in each experiment, the pose information of the camera and the calibration pattern was acquired to 30 locations with the different orientation and position. That is, we used 30 poses per a camera-sensor calibration. Experimental setup for the proposed method was the same as that of the conventional method except that the calibration pattern was moved instead of the camera. We performed those experiments in each layout we explained above. The overall process of our experiments is shown in Fig. 2.18.

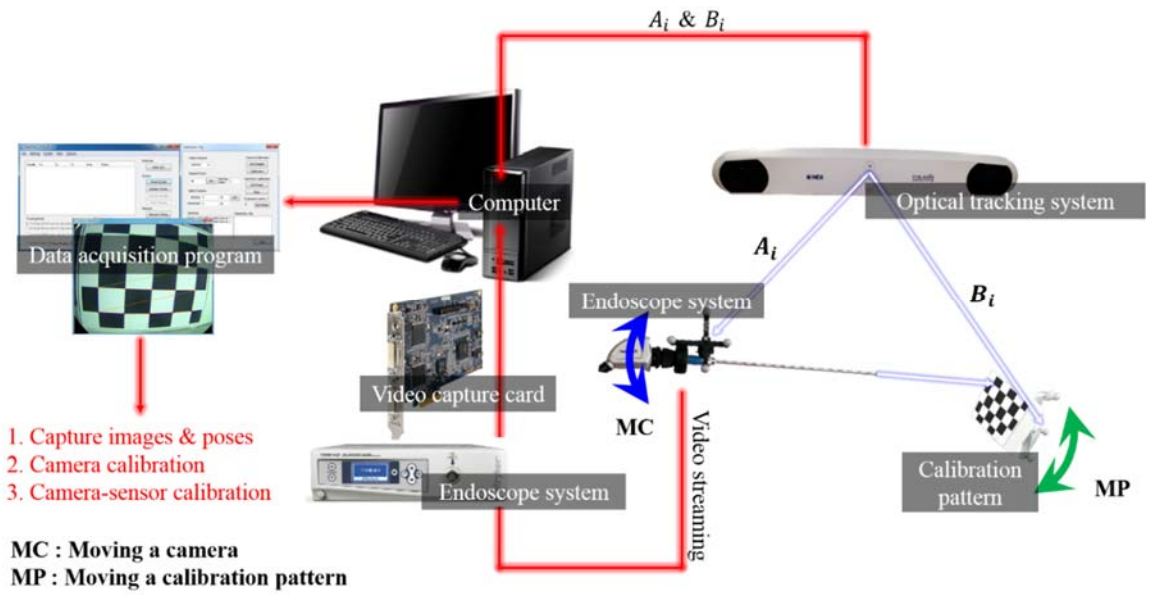


Figure 2.18 Experimental setup and overall process. From optical tracking system, $\mathbf{A_i}$ and $\mathbf{B_i}$ can be directly obtained. Endoscope system captures the image of the calibration pattern which is to be $\mathbf{C_i}$ after camera calibration. By specific event, those information are stored into memory at the same time.

The location of markers for the endoscope or the calibration pattern was selected considering the line of sight of optical tracking system. Therefore, the location of markers mounted is different according to layouts. The tool error which is results of the least squares minimization between the marker geometry in the tool definition file and the tool's measured marker position was also collected to perform camera-sensor calibration without the high error pose.

2.2.2 Data acquisition program

Data acquisition program based on the sample program distributed by NDI (Sample program is only designed for showing tracking information without saving tracking information or performing calibration) was developed by us with additional useful functions, and is based on C++ language. In addition, MFC was used for graphic user interface (GUI). Fig. 2.18 shows user interface (UI) of our program to acquire pose data from optical tracking system and endoscopes and to perform camera calibration. It is possible to set ROM file which can be made from 6-D architecture program distributed by NDI and includes the geometry of the tool's markers, the tool's manufacturing data, information on marker and face normal, face definitions, and the parameters used to track tools [27].

Through calibration menu, the number of corner points in the calibration pattern, the size of each square, and the repeat count can be set through edit boxes. Those parameters are required to perform camera calibration previously. In addition, this program provides the function to select the resolution and the frame rate for video streaming. Once we start this program by clicking the 'Start Tracking' button, this program shows the information about the poses that the optical tracker tracks, and the tool error through list box in real time. Through the 'Get Poses' button, we can start camera-sensor calibration step. Once camera-sensor calibration step runs, the image, the pose and the error of pose are captured and stored by specific event (In this time, we used 'enter' key to give the event.) at the same time in the path user set. Images, poses, and errors were saved as bitmap, text, and text format, respectively. Our program check automatically that the optical tracking system tracks markers mounted on the camera and the calibration pattern, and the corner points are correctly found at the same time. Therefore, if the markers aren't tracked or the corner points aren't found simultaneously, the program doesn't store that pose and that image although users give the event for data acquisition. After finishing data acquisition by moving target (one is to move the calibration pattern, the other is to move the camera), next steps are to extract extrinsic parameter from images followed by camera-sensor calibration.

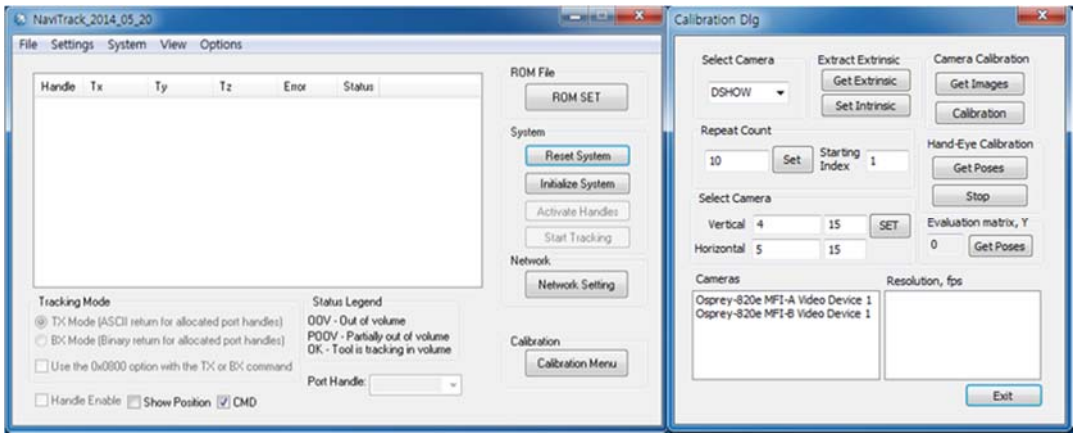


Figure 2.19 Data acquisition program.

Before extracting extrinsic parameter from images, intrinsic parameters of the camera are required. In this time, we used the intrinsic parameters solved for calculating re-projection error, which will be described in next sub-section. Note that we used the same intrinsic parameters only for one layout. If layout is changed, the intrinsic parameters are also changed. Once the intrinsic parameters are solved through camera calibration, the parameters can be loaded to the program through the 'Set Intrinsic' button. Then, **C** of **AX** = **BYC** problem, which is also **B** of **AX** = **XB** problem, can be calculated and saved as text format through the 'Get Extrinsic' Button. In addition, image points, corner points are also saved as text format. The points are used for calculating re-projection error as a reference value. Next, in order to perform camera-sensor calibration, we used MATLAB to apply the algorithm to solve the **AX** = **BYC** problem or the **AX** = **XB** problem.

2.2.3 Re-projection error

Once the calibration step in the layout 1 was finished, that in the layout 2 and 3 was performed as the same manner above. After finishing all calibration steps of each layout, comparison steps were performed. In order to compare each method in terms of the accuracy, the re-projection error was calculated in MATLAB. The re-projection point can be solved by using Eq. 83.

$$\begin{bmatrix}\mathbf{p}'\\1\end{bmatrix} = \begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}[\mathbf{I_{3\times3}}|\mathbf{0_{3\times1}}][\mathbf{X}^{-1} \quad \mathbf{A}^{-1} \quad \mathbf{B} \quad \mathbf{Y}]\begin{bmatrix}\mathbf{P}\\1\end{bmatrix} \tag{83}$$

where $\mathbf{p}'$ is a 2-D point in image coordinate, f_x, f_y are focal length, c_x, c_y are principal points, **P** is a 3-D corner point of the calibration pattern, **A**, **B** are pose data obtained from optical tracking system, and **X**, **Y** are the solutions of **AX** = **BYC** problem. The difference between the re-projection point and the image point obtained from image processing is ideally zero. However, it is not zero since there are many factors to make the difference between the re-projection point and the image point such as incorrect image point, camera calibration, camera-sensor calibration. Eq. 83 represents only a point, but in our experiments, there are 20 points on an image since the number of corner we used is 5-by-4. Therefore, considering the number of images and corner points, we can obtain representative error of camera-sensor calibration such as Eq. 84.

$$\epsilon = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|\mathbf{p}_{ij} - \mathbf{p}'_{ij}\right\| \text{ (pixels)} \tag{84}$$

where m is the number of images we used in each experiment, n is the number of corner points of the calibration pattern, and $\mathbf{p}$ and $\mathbf{p}'$ are the corner points of the calibration pattern obtained from image processing of m-th picture and the re-projection points obtained from Eq. 83, respectively.

Re-projection error is originally used for estimating camera calibration. Therefore, instead of **A**, **B**, **X**, and **Y**, only extrinsic parameter, **C** is used. Once **X** and **Y** are solved, we can express Eq. 83 by using two ways. One is Eq. 83, the other is to use only **C** instead of **A**, **B**, **X**, and **Y**. In this paper, those methods are called 'indirect re-projection' and 'direct re-projection', respectively.

In this time, newly 50 images were used for calculating indirect re-projection error for each layout and result of camera calibration of each layout had less than 0.4 pixels in terms of the direct re-projection error. In addition, the results of camera calibration using the images also used for extracting the extrinsic parameters. The reason why newly pose datasets were used instead of existing pose datasets used for camera-sensor calibration is that

existing pose datasets tend to reduce indirect re-projection error. Since they don't affect the camera-sensor calibration, the re-projection error calculated from the pose datasets provides us more reasonable results.

The problem we faced when the re-projection error is used for the evaluation is that **Y** has to be used in evaluating **X** from **AX** = **XB** problem. In **AX** = **XB** problem, there is no **Y** information; Therefore, we cannot obtain re-projection point by using Eq. 83. In order to calculate re-projection point, **Y** information is solved by the different method from **AX** = **XB** and **AX** = **BYC**.

Y can be solved by making the coordinates of the calibration pattern relative to optical tracking system. In order to make accurately the coordinate system, three points from optical tracking system are needed; therefore, a probe as shown in Fig. 2.19 is used to extract points. In order to track the tip of probe exactly, the pivot calibration (it is also known as 'Pivoting') are required which is the process to determine the offset of probe. To perform pivot calibration, we used 6-D architecture program. Since the coordinate system of the calibration pattern relative to the camera is formed as Fig. 2.19 (b), the coordinate system of the calibration pattern relative to optical tracking system must be the same as aforementioned coordinate system.

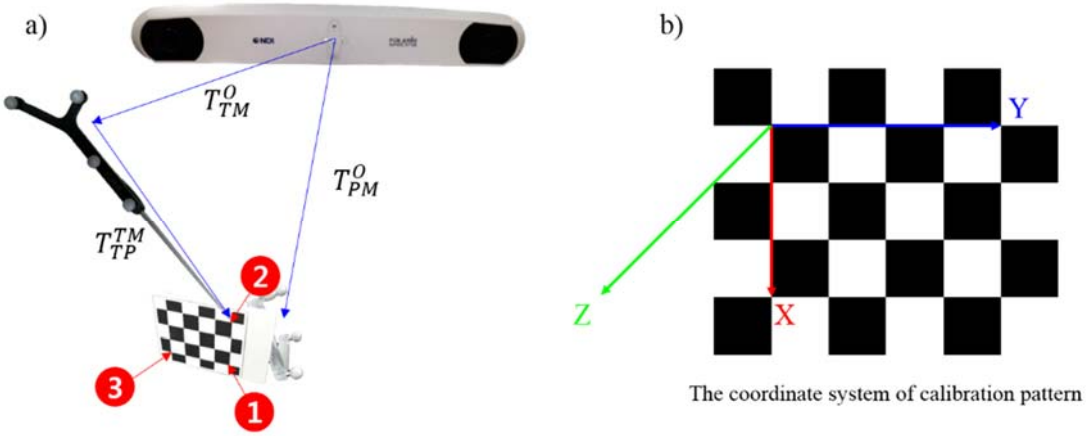


Figure 2.20 Process of making the coordinate system of calibration pattern. In order to make the coordinate system of the calibration pattern relative to optical tracking system, a probe is generally used.

The data acquisition program we made can also work for acquiring poses to make **Y** matrix. Through the 'Get Poses' button, the transformation matrix from optical tracking system to the tip of the probe can be obtained.

When the points are extracted, it is very important to keep the position of the tip of the probe on the corner point as shown in Fig. 2.19 (a) without any motion to ensure the accuracy of $\mathbf{Y}$. In order to make $\mathbf{Y}$, $\mathbf{T}_{\mathbf{PM}}^{\mathbf{O}}$ which is the transformation from optical tracking system to the markers mounted on the calibration pattern (PM) is required; therefore, the data acquisition program extracts simultaneously $\mathbf{T}_{\mathbf{PM}}^{\mathbf{O}}$ and $\mathbf{T}_{\mathbf{TP}}^{\mathbf{O}} = \mathbf{T}_{\mathbf{TM}}^{\mathbf{O}}\,\mathbf{T}_{\mathbf{TP}_*}^{\mathbf{TM}}$. TP* means the point where the tip of the probe is located. For example, TP1 means the point 1 relative to optical tracking system as shown in Fig. 2.19 (a). Once those matrices are obtained, three points are transferred from optical tracking system to PM by using Eq. 85-87.

$$\mathbf{T_1} = \mathbf{T}_{\mathbf{PM}}^{\mathbf{O}\,-1}\mathbf{T}_{\mathbf{TP1}}^{\mathbf{O}} \tag{85}$$

$$\mathbf{T_2} = \mathbf{T}_{\mathbf{PM}}^{\mathbf{O}\,-1}\mathbf{T}_{\mathbf{TP2}}^{\mathbf{O}} \tag{86}$$

$$\mathbf{T_3} = \mathbf{T}_{\mathbf{PM}}^{\mathbf{O}\,-1}\mathbf{T}_{\mathbf{TP3}}^{\mathbf{O}} \tag{87}$$

Therefore, three points, $\vec{\mathbf{P}}_{\mathbf{1}}, \vec{\mathbf{P}}_{\mathbf{2}}$, and $\vec{\mathbf{P}}_{\mathbf{3}}$ are extracted from three transformation matrices.

Then,

$$\vec{\mathbf{X}} = \vec{\mathbf{P}}_{\mathbf{2}} - \vec{\mathbf{P}}_{\mathbf{1}} \tag{88}$$

$$\vec{\mathbf{Y}} = \vec{\mathbf{P}}_{\mathbf{3}} - \vec{\mathbf{P}}_{\mathbf{1}} \tag{89}$$

$\vec{\mathbf{X}}$ and $\vec{\mathbf{Y}}$ are not normalized. In order to make unit vector, normalization step is required.

Then,

$$\hat{\mathbf{X}} = \frac{\vec{\mathbf{X}}}{\|\vec{\mathbf{X}}\|} \tag{90}$$

$$\hat{\mathbf{Y}} = \frac{\vec{\mathbf{Y}}}{\|\vec{\mathbf{Y}}\|} \tag{91}$$

Since $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ are orthogonal each other, $\hat{\mathbf{Z}}$ is calculated by using Eq. 92.

$$\hat{\mathbf{Z}} = \hat{\mathbf{X}} \times \hat{\mathbf{Y}} \tag{92}$$

Therefore, rotation matrix relative to PM is as Eq. 93.

$$\mathbf{R}_{\mathbf{I}}^{\mathbf{PM}} = [\hat{\mathbf{X}} \quad \hat{\mathbf{Y}} \quad \hat{\mathbf{Z}}] \tag{93}$$

Therefore,

$$\mathbf{Y} = \mathbf{T}_{\mathbf{I}}^{\mathbf{PM}} = \begin{bmatrix} \mathbf{R}_{\mathbf{I}}^{\mathbf{PM}} & \vec{\mathbf{P}}_{\mathbf{1}} \\ \mathbf{0}_{\mathbf{1}\times\mathbf{3}} & 1 \end{bmatrix} \tag{94}$$

Once **Y** are created, **X** from **AX** = **BYC** and **AX** = **XB** is evaluated by using the re-projection error.

Note that this method can also calculate **X** or **Y**, but this is less reliable, less accurate and more time consuming work than performing camera-sensor calibration. The reason is as following: the time and the error of pivot calibration, manually extracted points by using the probe, and non-based variable poses.

2.2.4 Survey of the motion range

As mentioned before, the range of motion affects the accuracy of camera-sensor calibration since optical tracking system has the spatial error which increases with the distance to optical tracking system. Therefore, through this survey, the difference between the range of motion of the conventional method and that of the proposed method is showed. The objective we surveyed is the position of markers mounted on the calibration pattern in the case of the proposed method and that of markers mounted on the camera in case of the conventional method. This information can be slightly different from original point since the spatial error and its oscillation which optical tracking system has itself. However, this variance is very smaller than the range of motion. Therefore, in researching the range of motion, the factors (the spatial error and the oscillation) can be ignored. In order to show the range of motion as the graph, the points used in camera-sensor calibration were extracted from $\mathbf{A_i}$ or $\mathbf{B_i}$ matrix. For example, in the case of moving the calibration pattern, the point from $\mathbf{B_i}$ matrix are extracted and in the case of moving the camera, the point from $\mathbf{A_i}$ matrix are extracted. All points used in 30 camera-sensor calibrations were used in the case of each method and each layout.

Ⅲ. RESULTS

3.1 Results

Dual quaternion (DQ) and Kronecker product (KP) methods were used for solving **AX** = **XB** problem and **AX** = **BYC** problem. Here, we present the experimental results performed in three layouts. As a result, the graphs of re-projection error, the range of motion of the camera and the calibration pattern, the table to show the range of motion in value, and the table for the results of t-test are presented for each layout. In the graph of the re-projection error, we present two kinds of graphs to provide various view points for readers. One shows all results of camera-sensor calibration using DQ and KP method. The other shows the average and the standard deviation relative to all results described in former. In the following graphs, we denote moving the camera, the conventional method, by "MC", and moving the pattern, the proposed method, by "MP". Vertical axis is the re-projection error (pixel unit). Plus, we performed t-test to show statistical significance. Since the environment among all layouts was different, t-test was performed by using results of each layout with a one-sided significance level of 1%. We appended all results of camera-sensor calibration to APPENDIX A. In APPENDIX A, rotation results were represented by Rodrigues formula.

3.1.1 Results of Layout 1

As mentioned before, layout 1 is the case less error is expected than other cases. Fig. 3.1-3.4 show the re-projection error of the layout 1. Fig. 3.1 and 3.2 show all results of camera-sensor calibration we performed when we applied DQ and KP method, respectively. Fig 3.3 and 3.4 show the average and the standard deviation relative to all results when we applied DQ and KP method, respectively. Although all results of MP method weren't better than that of MC, all results of MP were comparable with that of MC. From the average result as shown in Fig. 3.3

and 3.4, MP method was more accurate than MC method about 2.52 pixels in DQ method and 6.16 pixels in KP method. That is, in DQ and KP methods, the performance of camera-sensor calibration was improved by about 16.52% ($p < 0.01$) and 33.79% ($p < 0.01$), respectively when the proposed method were used. Also, the proposed method was more stable than the conventional method, which means the all results were almost similar to each other as shown in Fig 3.1 and 3.2. The standard deviation describes how our method is more stable than conventional method. Fig. 3.5 and 3.6 show the range of motion of the camera and the calibration pattern, respectively. The proposed method had smaller range of motion than that of the conventional method as expected. Table 3.1 shows the range of motion of each method relative to each axis and total volume that is $X \times Y \times Z$. The volume of the proposed method was smaller than that of the conventional method about 55 times.

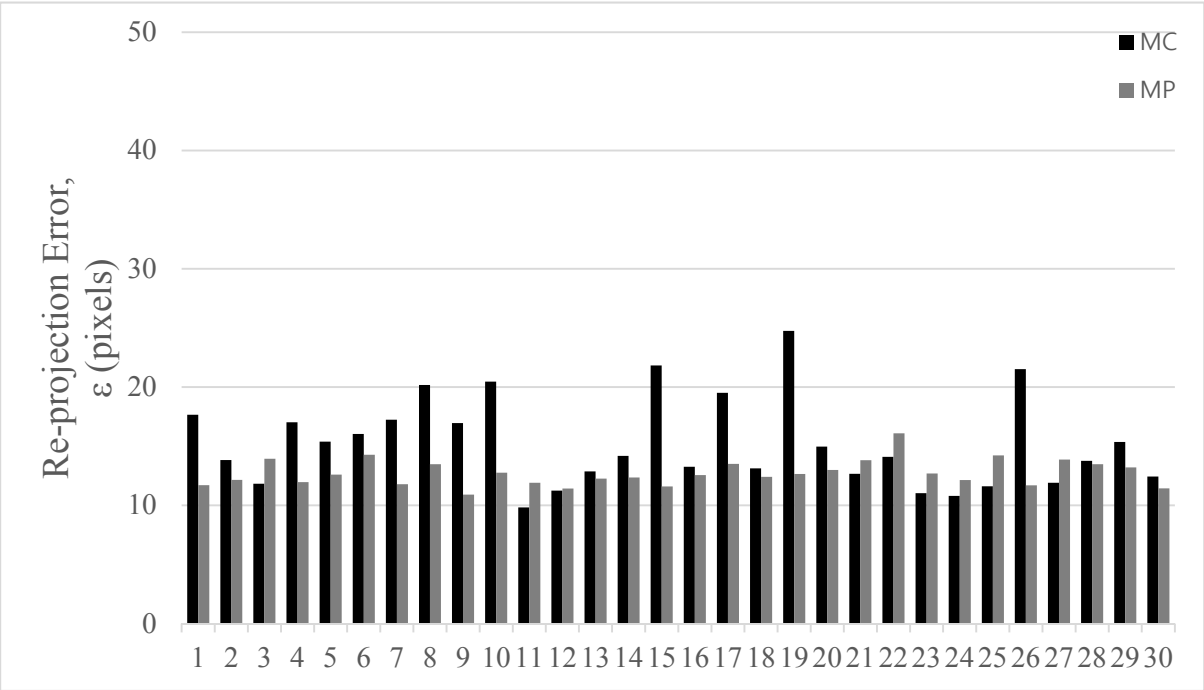


Figure 3.1 All results of camera-sensor calibration using DQ method in layout 1

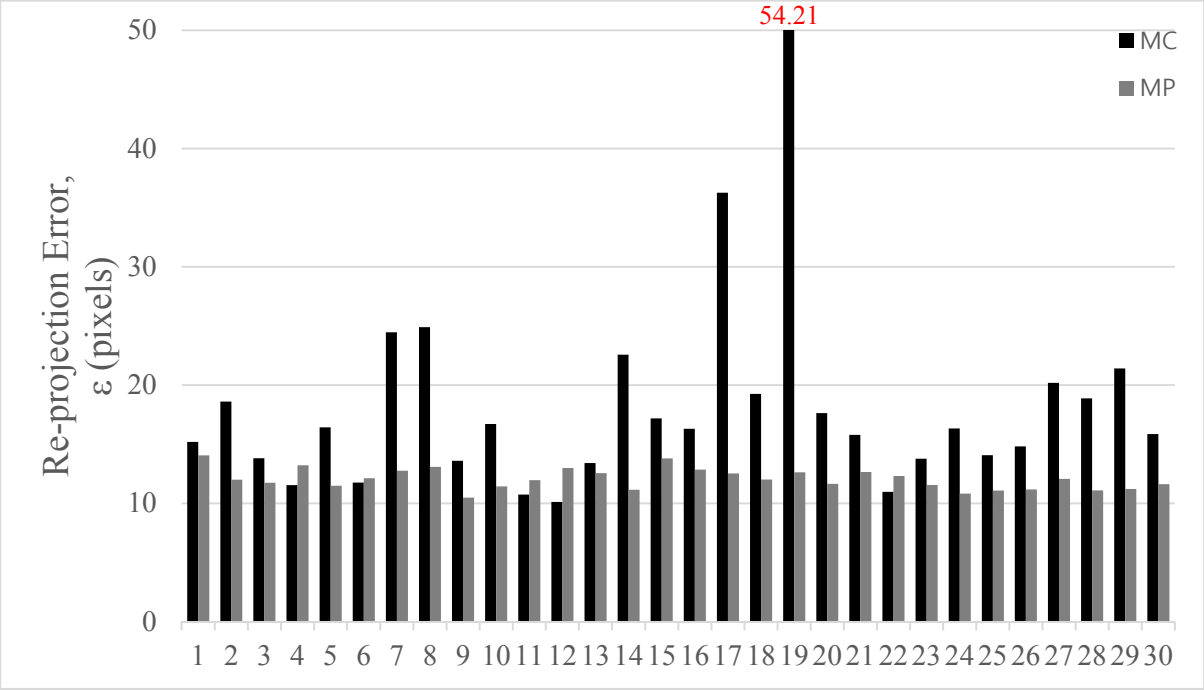


Figure 3.2 All results of camera-sensor calibration using KP method in layout 1

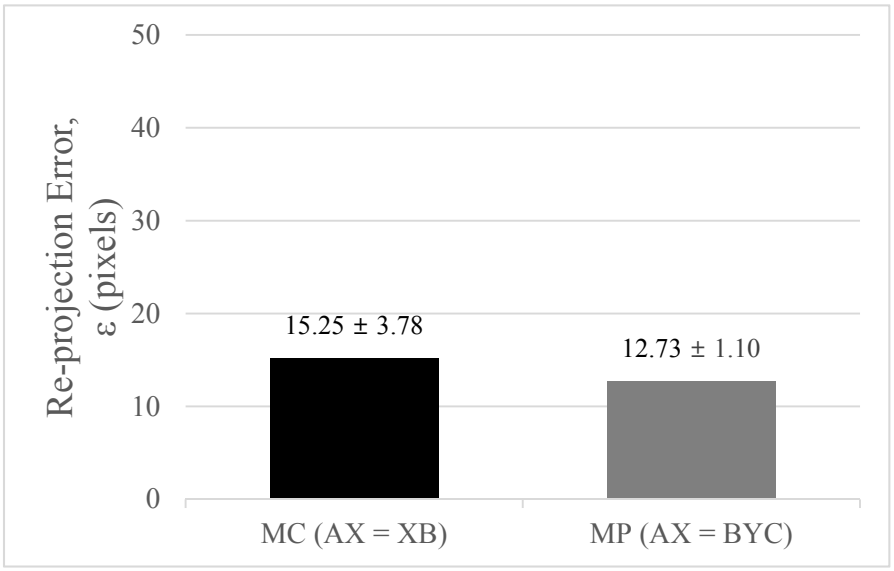


Figure 3.3 Average and standard deviation relative to results illustrated in Fig. 3.1

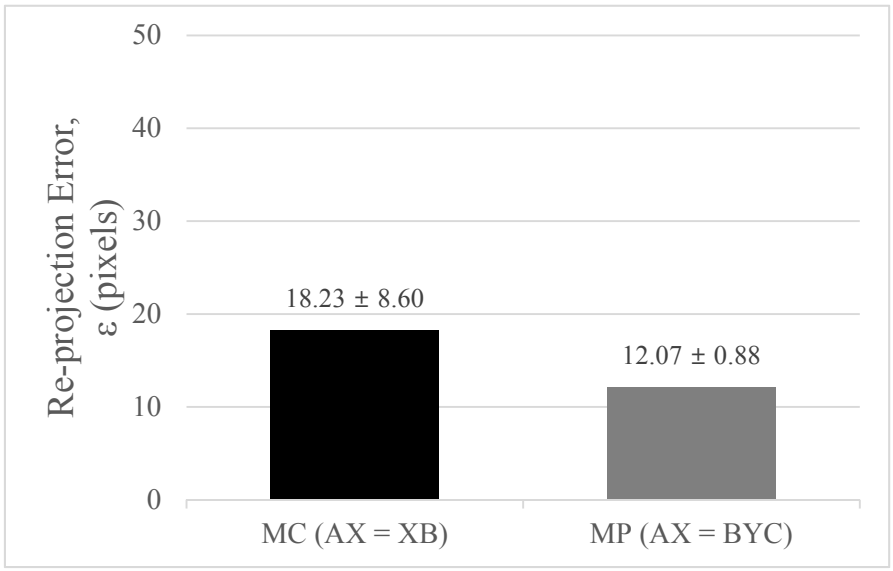


Figure 3.4 Average and standard deviation relative to results illustrated in Fig. 3.2

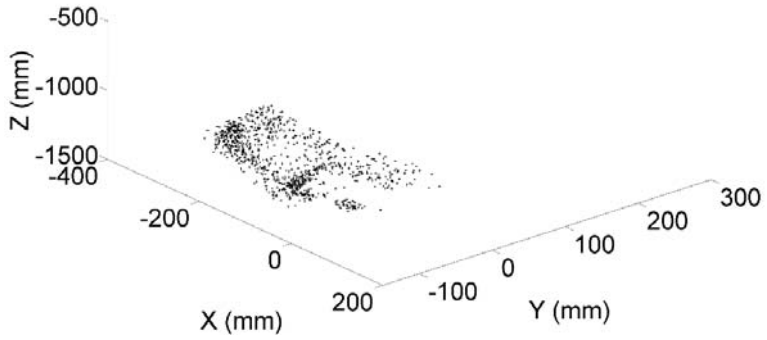


Figure 3.5 Range of motion of camera in layout 1

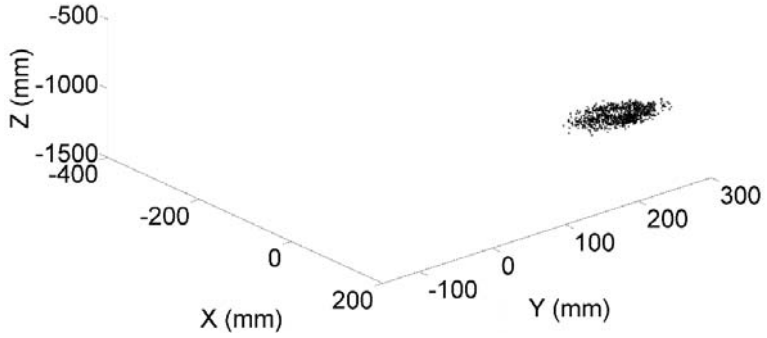


Figure 3.6 Range of motion of calibration pattern in layout 1

Table 3.1 Range of motion of each method in layout 1

Method	**AX** = **XB** method (Conventional method)	**AX** = **BYC** method (Proposed method)
X Range (mm)	-349 ~ 80	-122 ~ -39
Y Range (mm)	-64 ~ 113	269 ~ 406
Z Range (mm)	-1653 ~ -1028	-1360 ~ -1284
Total Volume (mm^3)	47458125	864196

Table 3.2 Results of t-test in layout 1

	P-value
Dual quaternion	0.0010
Kronecker product	0.0002

3.1.2 Results of Layout 2

Layout 2 is the case more error is expected than layout 1. Fig. 3.7-3.10 show the re-projection error of the layout 2. Fig 3.7 and 3.8 show all results of camera-sensor calibration when we applied DQ and KP method, respectively. Fig 3.9 and 3.10 show the average and the standard deviation relative to all results when we applied DQ and KP method, respectively. From the average result as shown in Fig. 3.9 and 3.10, in the case of applying KP method, MP method was still more accurate than MC method about 2.65 pixels. However, in the case of applying DQ method, MC method was slightly more accurate than MP method about 1.13 pixels but comparable accurate. Although the average results of MC was slightly better than MP, the standard deviation was still worse than MP. Fig 3.11 and 3.12 show the range of motion of each method. As expected, MP method had still smaller range of motion than MC method. Table 3.2 describes the range of motion relative to each axis and the volume numerically. The volume of the proposed method was smaller than that of the conventional method about 149 times. Although the performance of the proposed method using KP method was better than that of the conventional method in the case of only the average and the standard deviation, as shown in Table 3.4, the results were not statistically significant.

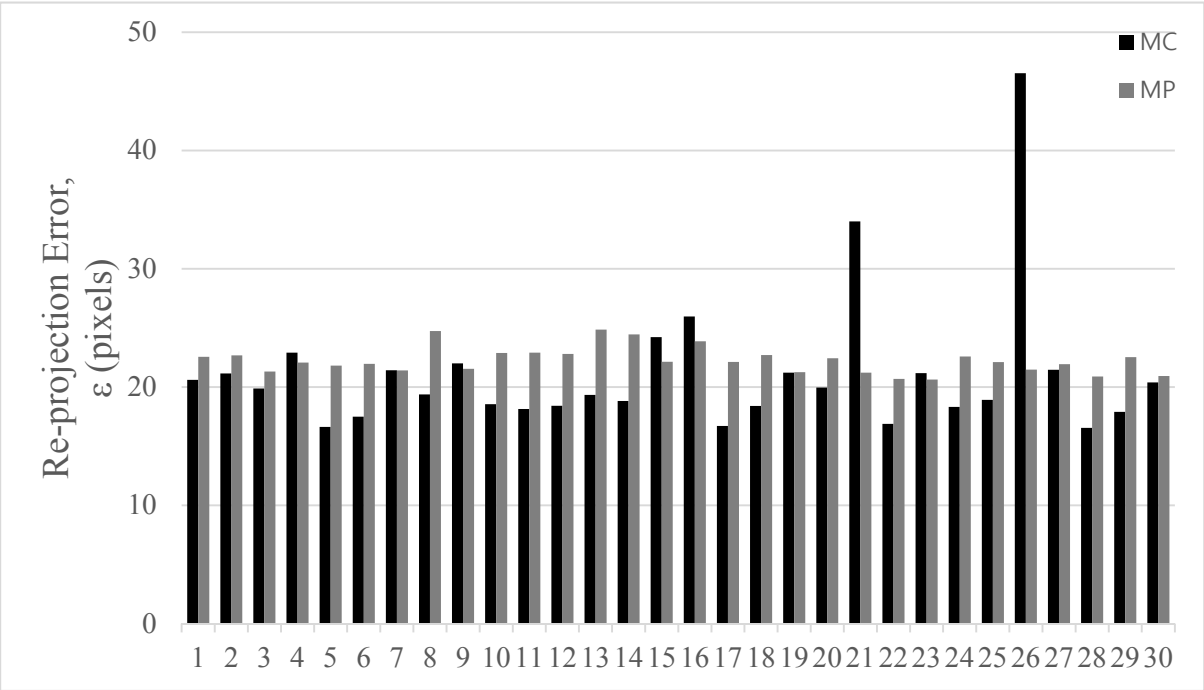


Figure 3.7 All results of camera-sensor calibration using DQ method in layout 2

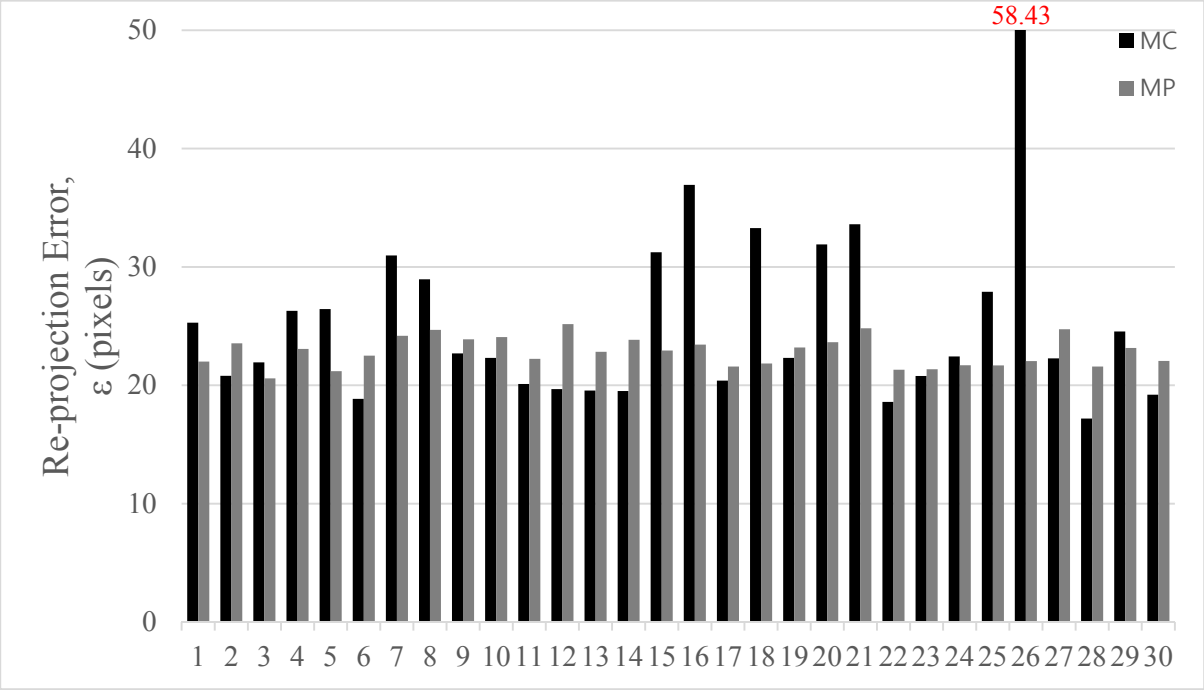


Figure 3.8 All results of camera-sensor calibration using KP method in layout 2

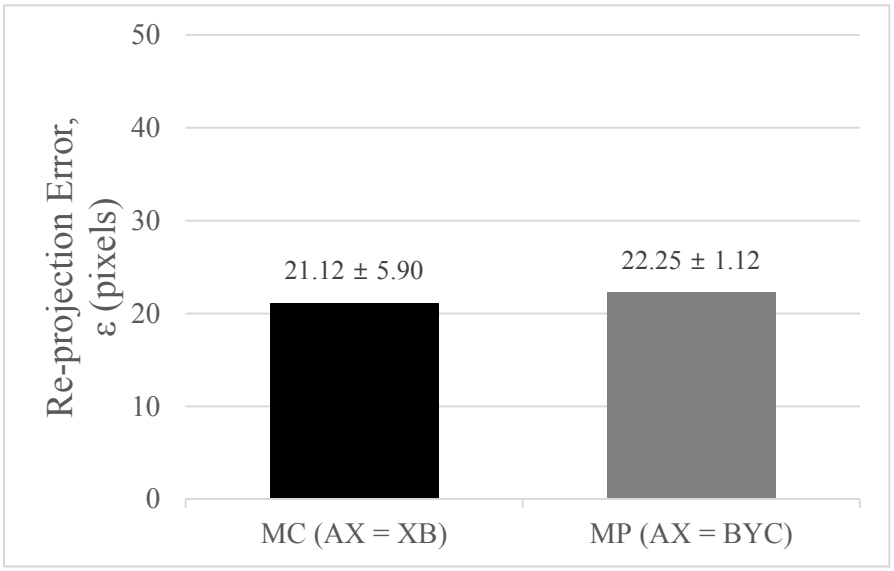


Figure 3.9 Average and standard deviation relative to results illustrated in Fig. 3.7

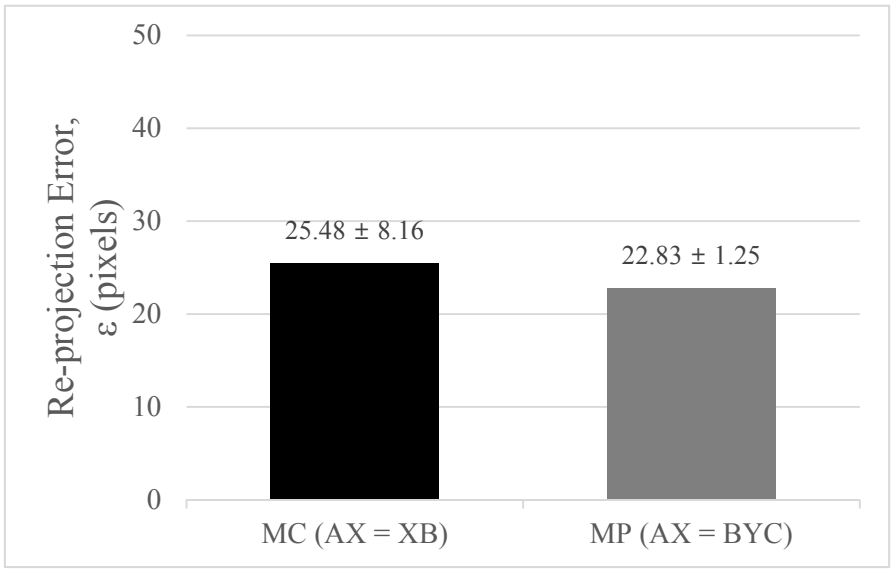


Figure 3.10 Average and standard deviation relative to results illustrated in Fig. 3.8

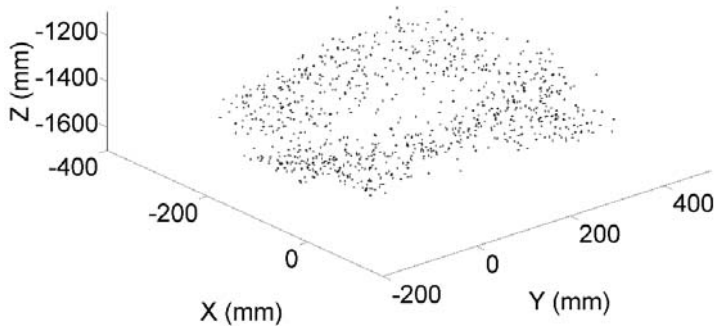


Figure 3.11 Range of motion of camera in layout 2

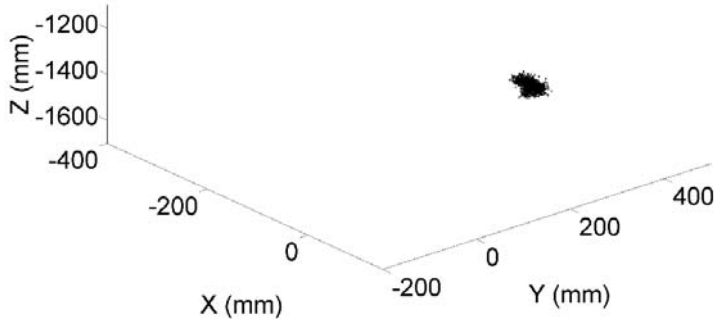


Figure 3.12 Range of motion of calibration pattern in layout 2

Table 3.3 Range of motion of each method in layout 2

Method	**AX** = **XB** method (Conventional method)	**AX** = **BYC** method (Proposed method)
X Range (mm)	-342 ~ 54	86 ~ 144
Y Range (mm)	-161 ~ 467	113 ~ 178
Z Range (mm)	-1656 ~ -1361	-1235 ~ -1104
Total Volume (mm^3)	73362960	493870

Table 3.4 Results of t-test in layout 2

	P-value
Dual quaternion	0.1610
Kronecker product	0.0427

3.1.3 Results of Layout 3

Layout 3 is also the case more error is expected than layout 1 such as layout 2. Fig. 3.13-16 show the result of layout 3. From the average result, the accuracy of the proposed method increased about 0.39 pixels in DQ method and 8.66 pixel in KP method. That is, the performance of camera-sensor calibration was improved by about 2.14% in DQ method and 30.82% ($p < 0.01$) in KP method when the proposed method were used. In addition, without reference to any method such as DQ and KP, the proposed method provided still stable results as the previous results. Fig 3.17 and 3.18 show the range of motion of each method. As expected, MP method had still smaller range of motion than MC method such as previous results. Table 3.3 describes the range of motion relative to each axis and the volume numerically. The volume of the proposed method was smaller than that of the conventional method about 14 times. Although the performance of the proposed method using DQ method was better than that of the conventional method in the case of only the average and the standard deviation, as shown in Table 3.6, the results were not statistically significant but significant in KP method.

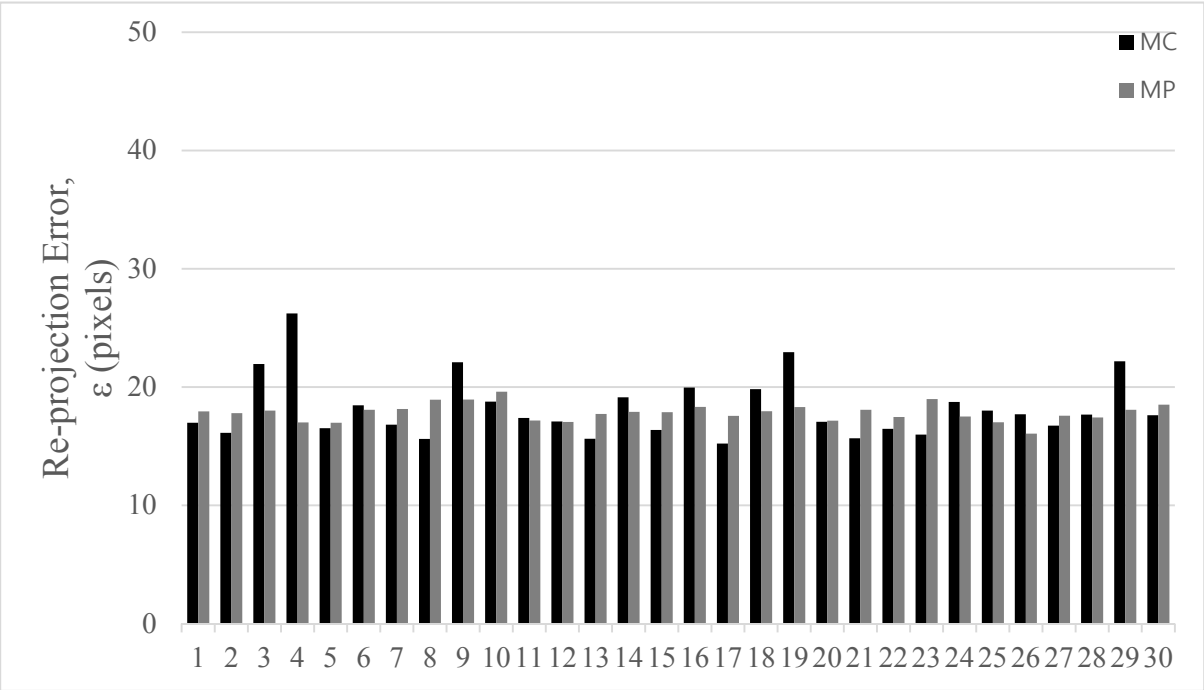


Figure 3.13 All results of camera-sensor calibration using DQ method in layout 3

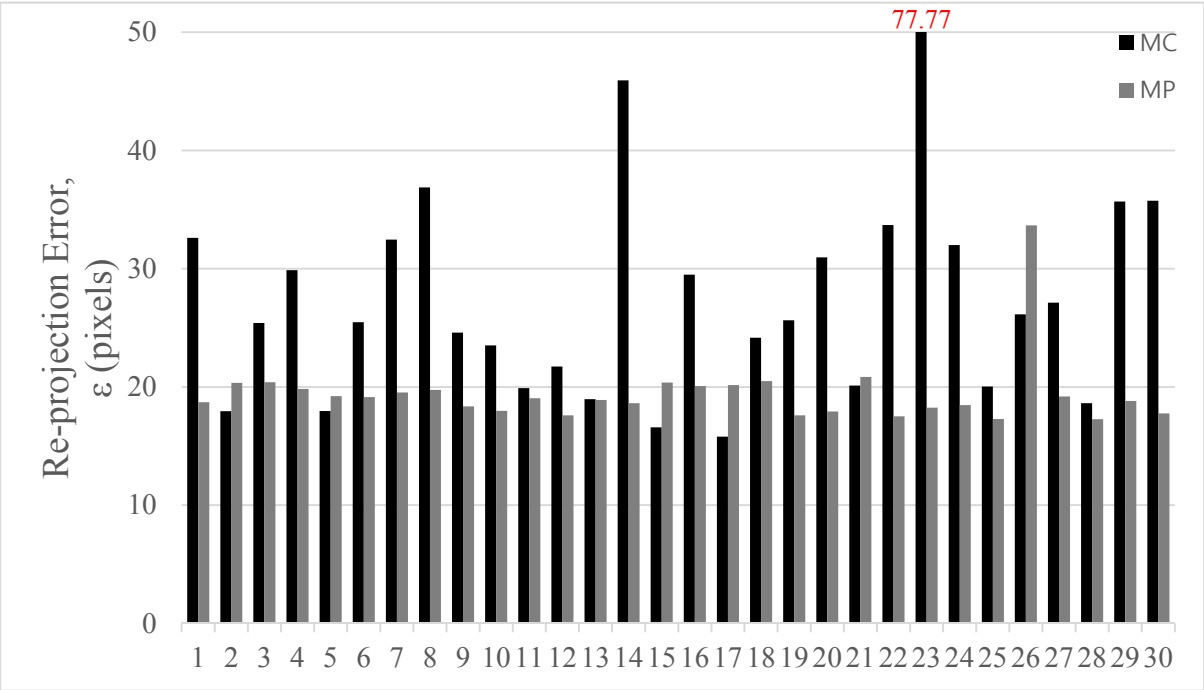


Figure 3.14 All results of camera-sensor calibration using KP method in layout 3

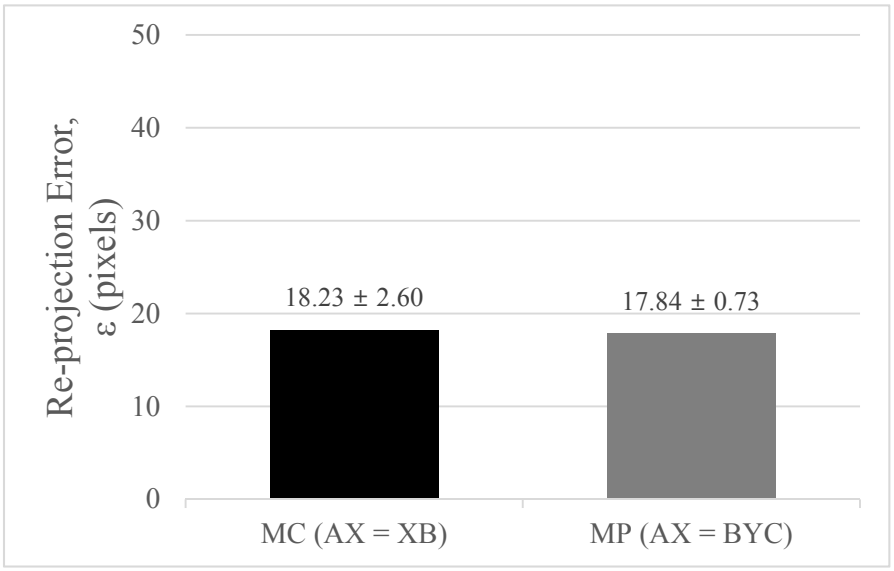


Figure 3.15 Average and standard deviation relative to results illustrated in Fig. 3.13

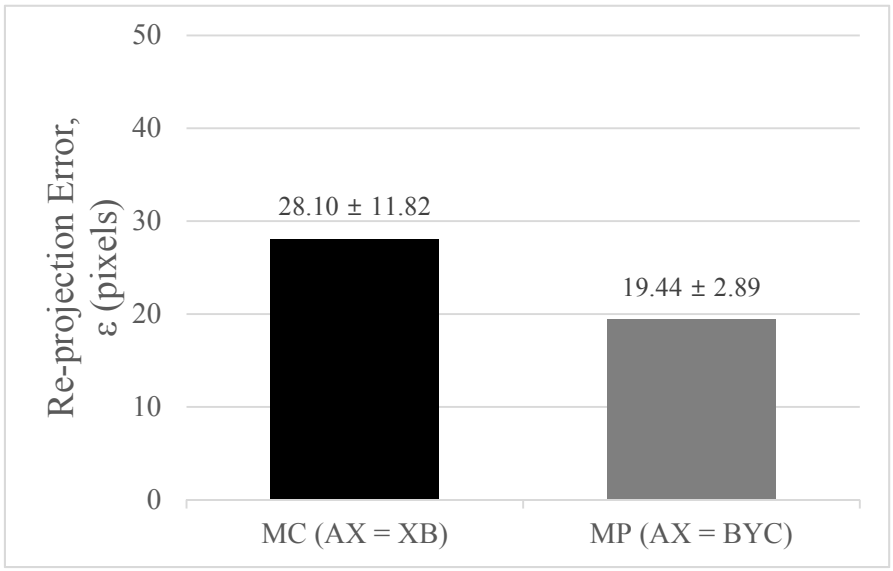


Figure 3.16 Average and standard deviation relative to results illustrated in Fig. 3.14

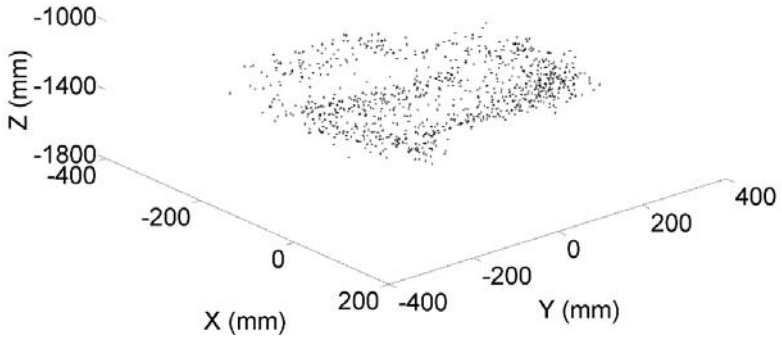


Figure 3.17 Range of motion of camera in layout 3

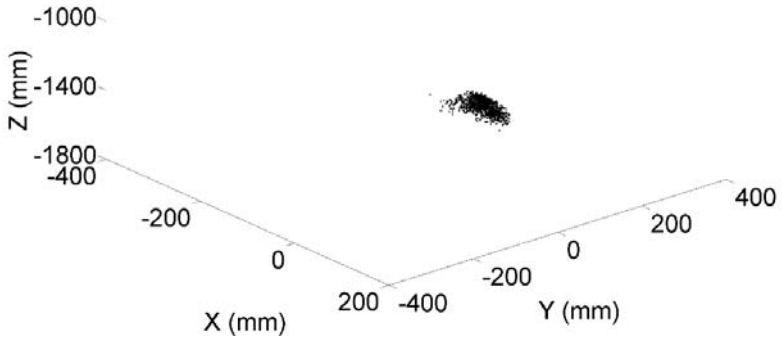


Figure 3.18 Range of motion of calibration pattern in layout 3

Table 3.5 Range of motion of each method in layout 3

Method	**AX** = **XB** method (Conventional method)	**AX** = **BYC** method (Proposed method)
X Range (mm)	-286~94	-235~-35
Y Range (mm)	-265~314	111~220
Z Range (mm)	-1483~-1194	-1728~-1512
Total Volume (mm^3)	63585780	4708800

Table 3.6 Results of t-test in layout 3

	P-value
Dual quaternion	0.2122
Kronecker product	0.0003

Ⅳ. DISCUSSION AND CONCLUSION

The aim of this study is to make the process of camera-sensor calibration easy and to improve the accuracy by reducing the influence of the spatial error of optical tracking system. Therefore, we proposed the new paradigm in performing camera-sensor calibration, which is to move the calibration pattern instead of moving the camera, the conventional paradigm. Our method is easily achieved by attaching the additional markers on the calibration pattern at the properly location.

From several results, we found layout 1 was the best position for improving the performance of camera-sensor calibration and the accuracy can be improved by using the proposed method. In this time, the distance between the calibration pattern and the markers mounted on the calibration pattern for the layout 1, the layout 2, and the layout 3 is about 69.87 mm, 100.53 mm, and 100.53 mm, respectively. The distance was calculated by using the method described in 2.2.3 section. The range of motion significantly depends on the distance; therefore, if the distance becomes shorter, the performance of the proposed method will be more improved. This also affects the conventional method. However, as mentioned before, the location of the markers to be mounted on endoscopes or microscopes is restricted by its structure. It means our method is more profitable for reducing the range of motion.

In the layout 2 and 3, the influence of the spatial error when the camera is moved is less than that in layout 1 since the spatial error is mainly influenced with z-axis of optical tracking system. By using the proposed method, the influence of the spatial error cannot be significantly reduced but can be still reduced. The facts were verified by performing t-test (Table 3.4, and 3.6). When DQ method was used in the layout 2 and 3, the proposed method was less effective. On the contrary, the proposed method using KP method was significantly effective. As mentioned before, KP method does not ensure the exact translation due to the normalization step to make the rotation matrix orthogonal. In addition, it has not the strong representation of rotation. The factors make KP method to be

more sensitive to the noise. Therefore, if the spatial error reduced, the performance of KP method increases since in noise free case, all methods to solve **AX** = **XB** or **AX** = **BYC** problem ideally solve the problems without errors.

Several results of the conventional method in each layout had very high re-projection error such as 19^{th} in layout 1, 26^{th} in layout 2, and 23^{th} in layout 3. As mentioned before, data acquisition program recorded the error relative to each pose. In 19^{th} in layout 1, 26^{th} in layout 2, and 23^{th} in layout 3, we observed the error relative to the specific poses was 1.52 mm, 0.83 mm, and 1.32 mm, respectively. We think those results were from the error of dynamic motion caused hand-tremor when we captured the pose or missing a marker of four markers mounted on the camera due to the limitation of angle between normal vector formed from marker and each sensor of optical tracking system [26].

In this paper, we suggested that our method is more convenient than the conventional method. However, we didn't perform any experiments to prove the convenience of our method. The only what we tried was to check the time consumed from camera-sensor calibration in each method. Although we don't check the time for every calibration, there was the obvious difference of time consuming in performing camera-sensor calibration. When we used the proposed method, we took about 40 seconds to capture 30 poses which is the number for a camera-sensor calibration, while we took about 4 minutes to capture 30 poses when we used the conventional method. The camera-sensor calibration is pre-processing work. However, in the operating room, it is very important to reduce the setting time due to the time of anesthesia of a patient. Since it is hard to configure AR system before the surgery, the time to perform camera-sensor calibration becomes very important factor. Therefore, to obtain accurate result from camera-sensor calibration once, the reliable method is required. It means the result from camera-sensor calibration must be always reliable if possible. From results, our method provides very stable results.

In the proposed method, we have to consider the position and the orientation of the added marker mounted on

the calibration pattern during camera-sensor calibration. In comparison with the proposed method, the conventional method does not require the additional marker for the calibration pattern. Therefore, the line of sight of optical tracking system, which have to be considered, is only one. Aforementioned things mean moving the calibration pattern could be limited in some layout. Through various experiments, we knew these problem was frequently occurred from layout 3. Since the size of the endoscope is usually larger than the calibration pattern, this makes frequently the line of sight from optical tracking system to the calibration pattern disconnected; therefore, the movement of the endoscopes or the calibration pattern were significantly limited.

In this time, we used the re-projection error to evaluate the proposed method. However, this is not gold-standard method. In order to use the re-projection error to evaluate the proposed method, we need to perform camera calibration. However, this has the error itself, and we need to extract the points very exactly from images. Although we assume the intrinsic parameters are very accurately calculated, the extrinsic parameter may have the error since homograhpy is not accurate due to inaccurate points extracted from image and orthogonal property of rotation. Plus, we also have to extract exact pose data of optical tracking system, but this is also impossible due to the spatial error. The other methods to evaluate **X** are the predicted camera pose introduced by Tsai and Len [12], the error of **AX** – **XB** or **AX** – **YB**. The methods also have the same problems.

In this paper, although the experiments were performed with only the endoscopes, the proposed method is also applicable to camera-sensor calibration of microscopes in the low magnification. However, in the high magnification, we cannot use the pin-hole camera model since the perspective characteristic disappears. It means the camera calibration will be failed if we use general methods based on the pin-hole camera model.

Although the proposed method is very simply achieved, it is powerful solution in practice to ensure the high accuracy and the convenience in the operating room. Most of all, the convenience is very important factor for

surgeons; therefore, we believe that this method is useful for surgeons in building AR system.

Ⅴ. REFERENCES

[1] M.Siessegger, RA.Mischkowski, BT. Schneider, B.Krug, B.Klesper, JE.Zöller, “Image guided surgical navigation for removal of foreign bodies in the head and neck”, Journal of Cranio-Maxillofacial Surgery, vol. 29(6), pp. 321-325, 2001.

[2] R.Ewers, K.Schicho, G.Undt, F.Wanschitz, M.Truppe, R.Seemann, A.Wagner, “Basic research and 12 years of clinical experience in computer-assisted navigation technology: a review”, International Journal of Oral and Maxillofacial Surgery, vol. 34(1), pp. 1-8, 2005.

[3] R.Sindwani, “Image-guided surgery of the paranasal sinuses and skull base”, Missouri medicine, vol. 105(3), pp. 257-261, 2008.

[4] L.Masterson, E.Agalato, C.Pearson, “Image-guided sinus surgery practical and financial experiences from a UK centre 2001–2009”, The Journal of laryngology and otology, vol. 126(12), pp. 1224-1230, 2012.

[5] U.Mezger, C.Jendrewski, M.Bartels, “Navigation in surgery”, Langenbeck’s Archives of Surgery, vol. 398, pp. 501-514, 2013.

[6] T.Koivukangas, JP.Katisko, JP.Koivukangas, “Technical accuracy of optical and the electromagnetic tracking systems”, SpringerPlus, vol. 2(1), pp. 90-96, 2013.

[7] J.Hong, M.Hashizume, “An effective point-based registration tool for surgical navigation”, Surgical Endoscopy, vol. 24(4), pp. 944-948, 2010.

[8] M.Ken, J.Hong, “Advanced Imaging and Robotics Technologies for Medical Applications”, International Journal of Optomechatronics, vol. 5(4), pp. 299-321, 2011.

[9] G.Eggers, J.Muhling, R.Marmulla, “Image-to-patient registration techniques in head surgery”, International Journal of Oral and Maxillofacial Surgery, vol. 35, pp. 1081-1095, 2006.

[10] CR Jr.Maurer, GB.Aboutanos, BM.Dawant, RA.Margolin, RJ.Maciunas, JM.Fitzpatrick. “Registration of CT and MR brain images using a combination of points and surfaces”, Proceedings of SPIE, vol. 2434, pp. 109-123, 1995.

[11] PJ.Edwards, AP.King, DJ.Hawkes, O.Fleig, CR Jr.Maurer, DL.Hill, MR.Fenlon, DA.de Cunha, RP.Gaston, S.Chandra, J.Mannss, AJ.Strong, MJ.Gleeson, TC.Cox, “Stereo Augmented Reality in the Surgical Microscope”, vol. 62, pp. 102-108, 1999.

[12] R.Tsai, R.Lenz, “A new technique for fully autonomous and efficient 3D robotics hand/eye calibration”, IEEE Transactions on Robotics and Automation, vol. 5(3), pp. 345-358, 1989.

[13] Y. Shiu, S. Ahmad, “Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB”, IEEE Transactions on Robotics and Automation, vol. 5(1), pp. 16-29, 1989.

[14] J.C.K.Chou, M.Kamel, “Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions”, The International Journal of Robotics Research, vol. 10(3), pp. 240-254, 1991.

[15] K.Daniilidis, “Hand-Eye Calibration Using Dual Quaternions”, The International Journal of Robotics Research, vol. 18(3), pp. 286-298, 1999.

[16] N.Andreff, R.Horaud, “Robot Hand-Eye Calibration using Structure-from-Motion”, International Journal of Robotics Research, vol. 20(3), pp. 228-248, 2001.

[17] H. H. Chen, “A screw motion approach to uniqueness analysis of head-eye geometry”, IEEE Proceedings of Computer Vision and Pattern Recognition, vol. xx, pp. 145-151, 1991.

[18] H. Zhuang, Z. S. Roth, R. Sudhakar, “Simultaneous Robot/World and Tool/Flange Calibration by Solving Homogeneous Transformation Equations of the form AX=YB”, IEEE Transactions on Robotics and Automation, vol. 10(4), pp. 549-554, 1994.

[19] A. Li, L. Wang, D. Wu, “Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product”, International Journal of the Physical Sciences, vol. 5(10), pp. 1530-1536, 1995.

[20] F.Dornaika, R.Horaud, “Simultaneous Robot-World and Hand-Eye Calibration”, IEEE Transactions on Robotics and Automation, vol. 14(4), pp. 617-622, 1998.

[21] A.D. Wiles, D.G. Thompson, D.D. Frantz, “Accuracy assessment and interpretation for optical tracking system”, Proceedings of the SPIE, vol. 5367, pp. 421-432, 2004.

[22] S. Fanhuai, W. Jianhua, L. Yuncai, “An Approach to Improve Online Hand-Eye Calibration”, Proceedings of the Second Iberian conference on Pattern Recognition and Image Analysis, vol. 3522, pp. 647-655, 2005.

[23] M. Shah, “Comparing two sets of corresponding six degree of freedom data”, Computer Vision and Image Understanding, vol. 115(10), pp. 1355-1362, 2011.

[24] J.Schmidt, H.Niemann, “Data-Selection for Hand-Eye Calibration: A Vector Quantization Approach”, The International Journal of Robotics Research, vol. 27(9), pp. 1027-1053, 2008.

[25] Z.Zhengyou, “A Flexible New Technique for Camera Calibration”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22(11), pp. 1330-1334, 2000.

[26] Northern Digital Inc., “Passive Polaris Spectra User Guide”, Revision 5, Northern Digital Inc., 2011.

요 약 문

수술용 내시경과 현미경을 위한 간편하고 정확한 카메라-센서간 캘리브레이션

증강 현실 기반의 수술 내비게이션 시스템은 내시경 혹은 현미경을 통해 얻은 영상 위에 3차원으로 재구성된 중요 기관이나 장기를 중첩 표현해준다. 증강 현실을 구현하는 데 있어 정확도에 많은 영향을 주는 것은 카메라-센서간 캘리브레이션이다. 카메라-센서간 캘리브레이션은 카메라-외부트렉커간의 좌표 정합을 의미한다. 기존의 방법은 오직 카메라만을 움직여야 카메라-센서간 캘리브레이션을 성공적으로 수행할 수 있다. 하지만 수술 현장에서 사용되는 내시경이나 현미경 같은 경우, 크기와 무게가 일반 웹 카메라에 비해 크고 무거워 캘리브레이션의 불편함을 유발한다. 또한 캘리브레이션 편리를 위해 사용할 수 있는 로봇 시스템 역시 수술 현장에서는 수술 공간과 비용 때문에 사용이 거의 불가능하다. 무엇보다, 외부 트렉커는 거리에 따라 공간 오차가 증가하는 특성을 지니고 있기 때문에, 다양한 포즈를 얻기 위해 넓은 범위의 움직임이 필요한 카메라는 공간 오차에 대한 영향을 많이 받게 된다. 즉, 카메라-센서간 캘리브레이션을 수행할 시 공간 오차에 대한 영향을 덜 받기 위해, 다양한 포즈를 만들 때 보다 적은 범위를 사용하는 것이 중요하다. 본 논문에서는 앞서 언급된 단점들을 극복하기 위해 카메라 대신 카메라-센서간 캘리브레이션에서 항상 필요로 하는 캘리브레이션 패턴을 움직여 해를 구하는 방법을 제안한다. 단순히 캘리브레이션 패턴에 외부 트렉커가 추적할 수 있도록 마커를 부착함으로써 제안된 방법에 도달할 수 있다. 캘리브레이션 패턴을 움직이면 AX = BYC 형태의 수식이 세워지고, 이를 Dual quaternion이나 Kronecker product를 이용하여 풀 수 있다. 패턴은 카메라에 비해 크기와 무게가 작고 가벼우며, 마커의 부착 위치 또한 자유롭기 때문에 사용자에게 편리함을 제공하고 움직임의 변위를 보다 적게 할 수 있다는 것이 장점이다.

다양한 레이아웃에서도 제안된 방법이 기존 방법보다 안정적이고 정확한 결과를 가져다 준다는 것을 검증하기 위해 3가지의 레이아웃을 택하여 실험을 하였다. 카메라를 움직여서 구한 결과와 캘리브레이션 패턴을 움직여서 구한 결과를 비교하였고, 이를 통해 캘리브레이션 패턴을 움직여서 구한 결과가 모든 레이아웃 상에서 보다 안정적인 결과를 보였음을 확인하였다. 레이아웃 2번을 제외하고 정확도 역시 모두 향상됨을 보였다. 하지만 제안된 방법은 특정 레이아웃에서 캘리브레이션 패턴의 움직임이 제한된다는 단점을 지닌다. 기존 방법에서는 외부 트렉커 기준으로 캘리브레이션 패턴의 위치와 방위를 고려하지 않아도 되기 때문에, 외부 트렉커가 감지할 수 있는 범위 내에서 카메라가 자유롭게 움직일 수 있는 반면, 제안된 방법에서는 카메라뿐만 아니라 캘리브레이션 패턴의 위치도 외부 트렉커 기준으로

고려해야 하고, 카메라의 크기에 의해 캘리브레이션 패턴에 부착된 마커가 외부 트렉커로부터 가려지는 상황이 빈번하게 발생하기 때문에 캘리브레이션 패턴의 움직임이 자유롭지 못하다. 그럼에도 불구하고 제안된 방법은 기존 방법보다 더욱 편리하고, 정확하며, 안정적임을 확인하였다.

핵심어: 카메라 캘리브레이션, 핸드-아이 캘리브레이션, 수술 내시경, 수술 현미경.

APPENDIX A

Results using Conventional Method in Layout 1												
Index	Kronecker Product						Dual Quaternion					
	Rx (rad)	Ry (rad)	Rz (rad)	Tx (mm)	Ty (mm)	Tz (mm)	Rx (rad)	Ry (rad)	Rz (rad)	Tx (mm)	Ty (mm)	Tz (mm)
1	0.419761	-1.7521	1.657839	-60.4655	-283.452	-37.1536	0.412548	-1.76407	1.647711	-59.8074	-281.554	-36.8803
2	0.429071	-1.76934	1.638087	-57.8827	-281.064	-34.9708	0.41497	-1.76166	1.649615	-60.5863	-281.473	-35.8954
3	0.449916	-1.75439	1.658926	-61.565	-280.622	-33.9059	0.419954	-1.76494	1.648972	-59.584	-281.282	-35.5947
4	0.40121	-1.73718	1.650796	-58.9675	-280.961	-36.0394	0.408856	-1.75889	1.645988	-58.2211	-282.219	-37.0038
5	0.442991	-1.77345	1.644471	-61.6413	-285.943	-33.7191	0.410215	-1.7649	1.646908	-59.1792	-282.526	-36.4191
6	0.435233	-1.77501	1.641615	-60.6248	-280.494	-32.5492	0.413992	-1.76476	1.647536	-59.9651	-281.69	-36.4817
7	0.436348	-1.76504	1.665333	-62.9541	-285.064	-37.5068	0.405737	-1.75933	1.65384	-61.1828	-281.925	-36.1143
8	0.447758	-1.77631	1.640726	-63.0513	-284.712	-35.8366	0.404151	-1.75643	1.657561	-60.8572	-282.848	-37.702
9	0.449792	-1.76935	1.654222	-62.267	-279.212	-35.0681	0.420489	-1.77005	1.64382	-58.8233	-281.473	-36.0493
10	0.430087	-1.77291	1.643835	-61.1697	-282.809	-35.5202	0.417687	-1.76695	1.644126	-59.775	-281.578	-37.0142
11	0.405387	-1.75605	1.640251	-58.7235	-282.856	-35.3359	0.417054	-1.7669	1.647351	-59.7722	-280.671	-34.7837
12	0.419533	-1.76942	1.639318	-59.0556	-280.822	-34.2175	0.408292	-1.76105	1.653148	-59.6352	-280.864	-35.9793
13	0.433644	-1.75668	1.648327	-60.7759	-283.564	-35.717	0.413595	-1.76656	1.644625	-58.806	-281.829	-35.5691
14	0.448017	-1.76812	1.65336	-64.4822	-284.064	-35.4253	0.412022	-1.76128	1.647798	-59.0027	-281.5	-36.5246
15	0.433318	-1.75566	1.645356	-58.9416	-282.335	-34.3125	0.413965	-1.76679	1.638874	-56.9633	-281.904	-36.1606
16	0.413364	-1.76386	1.655591	-60.2626	-282.609	-33.2626	0.394787	-1.74349	1.664463	-59.5723	-282.211	-35.6419
17	0.437162	-1.77259	1.663528	-64.2337	-283.454	-38.4398	0.411628	-1.76755	1.652473	-61.2905	-281.767	-36.4556
18	0.432004	-1.79362	1.641335	-60.6593	-281.866	-34.7409	0.419026	-1.77312	1.650233	-61.0377	-281.951	-34.9747
19	0.408426	-1.70396	1.696434	-64.3537	-265.991	-37.9266	0.404561	-1.75341	1.653443	-62.182	-282.666	-36.9406
20	0.423058	-1.77849	1.64505	-59.6061	-283.334	-35.3448	0.415352	-1.76279	1.657072	-61.1073	-281.165	-36.2103
21	0.424983	-1.75834	1.649017	-59.7154	-282.82	-36.5587	0.412573	-1.75815	1.65606	-60.5624	-282.335	-36.2231
22	0.418931	-1.75322	1.648073	-58.7741	-280.302	-35.8819	0.413003	-1.76154	1.648836	-59.6136	-281.535	-36.4606
23	0.400293	-1.75946	1.638106	-57.7135	-282.645	-34.0934	0.411821	-1.76019	1.646417	-58.6441	-280.961	-35.6562
24	0.420939	-1.75577	1.654014	-59.3195	-283.198	-36.6945	0.41279	-1.76148	1.651153	-59.7119	-282.043	-35.5856
25	0.428351	-1.78377	1.62864	-59.9013	-283.682	-33.6909	0.418012	-1.76897	1.651563	-59.1097	-280.802	-34.6068
26	0.432898	-1.79088	1.627005	-59.2348	-282.014	-33.3204	0.419741	-1.76724	1.645345	-58.6028	-281.569	-37.0423

27	0.424674	-1.78957	1.636189	-58.1153	-282.72	-33.3536	0.424215	-1.78164	1.641272	-59.9388	-281.132	-34.2477
28	0.447565	-1.78175	1.655236	-61.0146	-280.438	-35.4919	0.41779	-1.76915	1.648601	-59.8641	-281.508	-35.8224
29	0.439313	-1.78997	1.677231	-61.2752	-271.814	-31.6459	0.406412	-1.75472	1.667081	-60.0735	-280.312	-34.368
30	0.402229	-1.7749	1.630809	-57.3516	-282.782	-32.8159	0.400956	-1.75805	1.649797	-59.9044	-281.91	-35.6652
Avg.	**0.42788**	**-1.7667**	**1.64896**	**-60.47**	**-281.59**	**-35.018**	**0.41254**	**-1.7632**	**1.65006**	**-59.779**	**-281.64**	**-36.002**

Results using Proposed Method in Layout 1												
Index	**Kronecker Product**						**Dual Quaternion**					
	Rx (rad)	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**	**Rx (rad)**	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**
1	0.405068	-1.77117	1.636508	-59.722	-280.605	-33.2618	0.42225	-1.78629	1.641947	-59.8745	-280.629	-33.9725
2	0.416469	-1.76846	1.644283	-60.3675	-280.093	-33.9915	0.409209	-1.79074	1.646855	-59.7132	-280.445	-34.0996
3	0.417392	-1.76815	1.648127	-60.3275	-279.52	-34.7102	0.418424	-1.78551	1.632563	-59.6845	-280.195	-34.1456
4	0.413049	-1.77247	1.641031	-60.2372	-279.684	-33.7042	0.4152	-1.7862	1.643734	-59.9085	-280.514	-34.1491
5	0.415584	-1.77265	1.644277	-60.2422	-279.884	-33.8108	0.41099	-1.79057	1.643069	-59.7924	-280.695	-34.0463
6	0.408711	-1.76764	1.641534	-59.5435	-279.29	-34.4364	0.41656	-1.79235	1.635868	-59.5356	-279.413	-33.9824
7	0.414658	-1.77005	1.641572	-60.1795	-279.552	-33.8521	0.414828	-1.78652	1.638841	-59.6413	-280.623	-33.9437
8	0.412267	-1.76876	1.64235	-60.1869	-280.502	-33.4725	0.422519	-1.78984	1.633915	-59.7047	-280.54	-33.9005
9	0.407557	-1.77258	1.639185	-59.2733	-280.434	-33.6847	0.422408	-1.78134	1.64736	-60.1002	-280.676	-34.21
10	0.413993	-1.77452	1.641047	-59.9397	-279.773	-33.843	0.423012	-1.78546	1.638472	-59.7702	-280.434	-34.1785
11	0.418438	-1.77073	1.645481	-60.3722	-279.202	-33.8908	0.410416	-1.79073	1.641659	-59.7196	-280.598	-33.7978
12	0.409884	-1.77036	1.640694	-59.8094	-279.208	-33.8148	0.421	-1.78073	1.638817	-59.778	-280.483	-34.1433
13	0.411041	-1.77296	1.63941	-60.0741	-280.465	-33.6196	0.41455	-1.78181	1.638485	-59.953	-280.812	-34.1125
14	0.417096	-1.77631	1.643892	-59.8496	-279.456	-34.3488	0.422521	-1.77941	1.631356	-59.681	-280.049	-34.0328
15	0.416517	-1.77242	1.644774	-60.8671	-280.7	-33.2945	0.419897	-1.78352	1.640016	-59.8505	-281.067	-34.0667
16	0.413064	-1.76648	1.645201	-60.0522	-278.932	-34.0725	0.416592	-1.78061	1.636713	-59.7654	-280.479	-34.2861
17	0.414025	-1.76596	1.646714	-60.376	-279.854	-34.4314	0.424019	-1.78162	1.628424	-59.8618	-280.458	-33.9835
18	0.411583	-1.76926	1.641933	-59.7142	-279.34	-33.6967	0.422136	-1.78575	1.640171	-59.8251	-280.083	-34.1342
19	0.412517	-1.76425	1.645901	-60.0654	-279.342	-34.2065	0.406745	-1.79237	1.646196	-59.7654	-280.5	-34.0674
20	0.4147	-1.76876	1.643972	-59.9404	-279.189	-34.1213	0.420389	-1.78422	1.633106	-59.5912	-280.379	-34.114
21	0.414686	-1.77363	1.637513	-59.8965	-279.157	-33.5866	0.415714	-1.79286	1.641148	-59.7018	-280.464	-34.1134
22	0.412438	-1.77511	1.635583	-59.5598	-279.027	-33.6054	0.419313	-1.7869	1.62549	-59.4723	-280.547	-34.2696

23	0.413186	-1.76861	1.642269	-60.2203	-281.546	-34.0653	0.396576	-1.78846	1.64207	-59.5663	-281.506	-33.9892
24	0.41021	-1.76909	1.642501	-59.8486	-281.477	-34.3086	0.420235	-1.7839	1.635112	-59.7995	-281.177	-33.9769
25	0.421981	-1.77251	1.644849	-60.4874	-279.722	-33.8193	0.417666	-1.78579	1.628468	-59.7062	-281.05	-34.0069
26	0.410728	-1.77198	1.638976	-59.7605	-281.21	-34.1753	0.414193	-1.7833	1.639108	-59.6903	-281.124	-34.1108
27	0.407241	-1.77068	1.637146	-59.5016	-280.245	-33.6687	0.421285	-1.78357	1.628081	-59.5875	-281.072	-34.1289
28	0.409349	-1.77049	1.638165	-59.3876	-280.077	-34.1379	0.413582	-1.78519	1.635802	-59.6507	-280.77	-34.2673
29	0.414548	-1.77001	1.643989	-60.1404	-280.339	-34.0105	0.419498	-1.78908	1.635661	-59.7394	-280.732	-33.9712
30	0.412821	-1.76867	1.641585	-59.9001	-280.152	-33.8106	0.409449	-1.78479	1.646711	-59.9015	-280.626	-34.1664
Avg.	**0.41303**	**-1.7705**	**1.64202**	**-59.995**	**-279.93**	**-33.915**	**0.41671**	**-1.786**	**1.63784**	**-59.744**	**-280.6**	**-34.079**

Results using Conventional Method in Layout 2

Index	Kronecker Product						Dual Quaternion					
	Rx (rad)	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**	**Rx (rad)**	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**
1	-0.0301	0.142386	-1.55228	81.0891	-30.8242	320.6798	-0.05926	0.181083	-1.5464	79.86128	-29.4609	316.7194
2	-0.07192	0.19425	-1.5488	78.36921	-29.6003	315.9606	-0.05776	0.193571	-1.54638	79.27564	-30.7405	315.3278
3	-0.05497	0.14582	-1.54236	78.88883	-31.1702	312.5871	-0.05231	0.178842	-1.54147	80.49562	-30.2737	316.2745
4	-0.0388	0.211094	-1.5412	79.36769	-26.7562	324.5031	-0.04663	0.177381	-1.54298	80.9825	-28.2369	317.6003
5	-0.02731	0.173656	-1.53042	80.16047	-27.4605	321.7235	-0.03875	0.191186	-1.54212	79.08998	-28.5988	315.8655
6	-0.03263	0.152711	-1.54991	82.01775	-30.3139	318.6791	-0.04172	0.183826	-1.54222	79.90664	-29.9584	315.8397
7	-0.05293	0.205453	-1.53744	80.41105	-26.7544	312.0371	-0.04083	0.185845	-1.53997	80.71821	-27.6684	315.5576
8	-0.0391	0.177824	-1.53755	76.06789	-30.2796	311.3837	-0.04672	0.192253	-1.54017	79.28996	-30.3129	314.1542
9	-0.02528	0.151876	-1.57367	80.4945	-29.7417	316.0997	-0.03838	0.171939	-1.5487	79.97198	-27.6099	316.4381
10	-0.0106	0.179049	-1.5488	80.55804	-25.2066	313.9064	-0.0343	0.18338	-1.54465	80.05816	-27.5381	315.787
11	-0.02804	0.181067	-1.54184	81.65594	-28.2567	318.6792	-0.04439	0.1886	-1.5431	79.74997	-29.5907	315.2346
12	-0.06319	0.144311	-1.54552	79.87606	-31.4104	313.8855	-0.04512	0.187336	-1.5435	79.74022	-28.4375	315.2628
13	-0.02364	0.139051	-1.56197	82.27091	-30.1639	316.1082	-0.04797	0.186386	-1.54587	80.06467	-29.0653	315.2014
14	-0.02784	0.191155	-1.54283	81.02273	-27.4173	316.0409	-0.04845	0.199037	-1.54111	78.67134	-29.8541	314.5141
15	-0.04285	0.185024	-1.55772	83.10668	-29.0663	316.605	-0.05701	0.188055	-1.54832	80.48787	-29.0832	314.9874
16	-0.02109	0.191794	-1.55646	79.07385	-21.9088	312.9986	-0.0268	0.180953	-1.54814	78.73776	-26.0399	316.0862
17	-0.01524	0.176742	-1.53863	82.77586	-27.3528	317.2335	-0.04206	0.189016	-1.5431	79.10285	-29.6225	314.955
18	0.001528	0.196378	-1.54108	79.32732	-22.6654	319.5454	-0.04269	0.193084	-1.54469	78.39379	-27.515	315.5409

19	-0.03045	0.193703	-1.54432	79.20668	-30.043	319.3223	-0.0514	0.19344	-1.54728	78.84987	-31.3739	315.1779
20	0.011367	0.202643	-1.52238	85.16292	-24.9288	318.1366	-0.03426	0.188932	-1.54669	80.93829	-29.0746	316.6726
21	-0.00105	0.172335	-1.5109	86.39406	-28.3689	313.5312	-0.04452	0.203797	-1.54143	81.35542	-30.1094	312.0057
22	-0.02148	0.19448	-1.56286	79.12161	-28.41	314.1615	-0.02919	0.195884	-1.54907	78.82146	-28.2217	315.4177
23	-0.02932	0.194476	-1.54388	77.97476	-28.8125	316.329	-0.03985	0.201527	-1.54417	78.42937	-30.2792	315.1226
24	-0.00975	0.199195	-1.55813	82.23525	-27.3942	316.2605	-0.02803	0.19678	-1.54735	79.99031	-28.6843	315.5734
25	-0.02549	0.228092	-1.53374	80.5733	-25.4923	315.5911	-0.03868	0.19977	-1.54487	78.7904	-29.6307	314.5101
26	-0.05211	0.359421	-1.45457	71.11865	-12.3239	308.0583	-0.05447	0.240621	-1.53538	80.06604	-28.6298	309.606
27	-0.02823	0.18709	-1.55939	81.98057	-27.3786	316.2366	-0.04605	0.1796	-1.54693	81.22503	-29.5118	315.8977
28	-0.03167	0.217741	-1.54782	78.02105	-26.6273	313.7985	-0.03276	0.198961	-1.54685	78.81303	-28.2291	315.422
29	-0.00304	0.180249	-1.56065	81.69024	-28.4179	322.5676	-0.03933	0.191461	-1.54827	78.59348	-29.8954	315.4954
30	-0.02549	0.173725	-1.53758	82.14847	-28.9972	317.4082	-0.03709	0.194357	-1.54388	79.93032	-30	316.1457
Avg.	**-0.0284**	**0.18809**	**-1.5428**	**80.4054**	**-27.451**	**316.335**	**-0.0429**	**0.19123**	**-1.5445**	**79.68**	**-29.108**	**315.28**

Results using Proposed Method in Layout 2

Index	**Kronecker Product**						**Dual Quaternion**					
	Rx (rad)	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**	**Rx (rad)**	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**
1	-0.05674	0.166209	-1.54222	79.70509	-28.9156	314.7557	-0.05093	0.175566	-1.54343	79.50451	-27.7694	314.1515
2	-0.04922	0.164542	-1.54555	80.28311	-28.2425	315.7272	-0.0569	0.17697	-1.55152	79.24234	-28.105	315.1036
3	-0.04544	0.163713	-1.54515	80.6181	-28.7268	314.9158	-0.04296	0.174074	-1.54243	79.62693	-27.7062	314.6775
4	-0.04847	0.162258	-1.54431	80.14372	-28.3358	314.4906	-0.04951	0.175933	-1.54332	79.49246	-27.7776	314.3037
5	-0.05351	0.167149	-1.54406	79.94326	-28.8375	314.3525	-0.04728	0.168561	-1.5459	79.92203	-28.1557	314.5026
6	-0.04756	0.165396	-1.54357	79.91722	-28.2458	314.734	-0.04619	0.17084	-1.54914	79.80179	-27.8656	314.1939
7	-0.06337	0.162016	-1.54983	77.58846	-30.3468	316.632	-0.02928	0.169509	-1.53322	80.10929	-27.4117	314.9854
8	-0.04539	0.157142	-1.54116	81.2938	-28.2153	315.0495	-0.05468	0.167859	-1.54163	79.89218	-28.0822	315.032
9	-0.0473	0.168378	-1.54546	79.92235	-27.6929	314.7091	-0.04805	0.181559	-1.55168	79.59044	-27.5209	314.499
10	-0.05554	0.169426	-1.54303	79.56032	-28.1778	315.2626	-0.05424	0.169303	-1.54364	79.61316	-28.3566	315.0754
11	-0.04596	0.166318	-1.54436	80.24924	-28.1122	314.3996	-0.04663	0.171865	-1.54361	79.85223	-27.6211	314.1644
12	-0.04941	0.168742	-1.54326	80.12618	-27.5321	314.4188	-0.04904	0.173064	-1.55272	79.70099	-27.7061	314.2691
13	-0.05009	0.17487	-1.54309	79.63484	-27.7748	315.1211	-0.06055	0.174221	-1.53945	79.13646	-28.0488	315.0234
14	-0.05078	0.172885	-1.54336	79.77141	-27.6372	314.1227	-0.05337	0.165788	-1.5491	79.85929	-28.0815	314.3997

15	-0.04324	0.161943	-1.54387	80.6297	-28.0554	314.5179	-0.04801	0.174948	-1.54564	79.74781	-27.7334	314.4013
16	-0.04988	0.172513	-1.54393	79.72523	-27.7285	314.8753	-0.05386	0.176515	-1.53747	79.50829	-27.7016	314.5534
17	-0.05403	0.166424	-1.54398	79.75141	-28.8115	314.6876	-0.04645	0.174275	-1.54538	79.80419	-27.6914	314.6666
18	-0.04599	0.1651	-1.54437	79.91699	-28.2735	314.0134	-0.04408	0.166638	-1.55203	79.97233	-27.805	314.0025
19	-0.04709	0.167796	-1.54305	80.19867	-27.9212	314.9958	-0.04682	0.179257	-1.54988	79.6964	-27.6718	315.0069
20	-0.04979	0.17411	-1.54281	79.89249	-27.5769	314.085	-0.0505	0.180408	-1.53555	79.39434	-27.583	314.5363
21	-0.04577	0.172904	-1.54332	80.00341	-27.214	315.6268	-0.04242	0.17328	-1.54792	79.87289	-27.7613	315.1371
22	-0.04867	0.17213	-1.54246	79.64497	-28.1635	314.4671	-0.04565	0.172481	-1.54599	79.79899	-28.1123	314.6403
23	-0.04674	0.170121	-1.5445	79.94604	-28.0938	313.6731	-0.04666	0.178226	-1.54617	79.53903	-27.8103	314.2877
24	-0.05161	0.175957	-1.54294	79.56167	-28.075	314.7568	-0.05083	0.177379	-1.54688	79.53041	-27.6902	314.7354
25	-0.04122	0.166503	-1.54424	80.42929	-28.0275	315.3345	-0.04583	0.168644	-1.54986	79.84091	-27.9833	314.5678
26	-0.04979	0.175053	-1.54208	79.86317	-28.0014	315.3041	-0.04347	0.177495	-1.54782	79.85237	-27.4457	314.4817
27	-0.05457	0.165676	-1.54247	79.79655	-28.1211	314.2387	-0.04971	0.172406	-1.53756	79.64082	-28.1157	314.8455
28	-0.04741	0.169205	-1.54363	80.0225	-28.1898	314.3913	-0.04342	0.175721	-1.54838	79.74627	-27.6553	314.1389
29	-0.0491	0.166534	-1.54363	79.6782	-28.0865	314.3573	-0.04685	0.172196	-1.54539	79.77552	-27.727	314.5471
30	-0.04887	0.170311	-1.54204	79.91077	-28.1492	314.8886	-0.04355	0.174294	-1.53932	79.68835	-27.8187	314.4665
Avg.	**-0.0494**	**0.16804**	**-1.5437**	**79.9243**	**-28.176**	**314.763**	**-0.0479**	**0.17364**	**-1.5451**	**79.6918**	**-27.817**	**314.58**

Results using Conventional Method in Layout 3

Index	**Kronecker Product**						**Dual Quaternion**					
	Rx (rad)	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**	**Rx (rad)**	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**
1	-2.1241	-2.06982	0.347402	-43.2834	-23.5983	-330.993	-2.12884	-2.05524	0.346984	-41.9009	-20.7425	-326.388
2	-2.10918	-2.05229	0.334416	-42.3033	-23.2853	-327.026	-2.12021	-2.06226	0.345514	-41.8477	-21.0984	-325.097
3	-2.12808	-2.03177	0.350003	-40.4866	-23.6725	-329.384	-2.12386	-2.05871	0.344609	-40.8748	-21.1547	-326.944
4	-2.11588	-2.01802	0.356375	-37.2601	-26.3505	-321.907	-2.11686	-2.06778	0.357374	-45.0299	-20.5251	-322.681
5	-2.10361	-2.03203	0.332885	-38.1868	-23.7739	-322.574	-2.11822	-2.0611	0.354226	-40.9625	-20.9653	-324.345
6	-2.10284	-2.04138	0.256159	-41.9768	-29.6088	-323.381	-2.11935	-2.05976	0.347017	-43.799	-20.3099	-324.551
7	-2.10111	-2.08558	0.273459	-43.8365	-19.7488	-319.341	-2.11093	-2.05825	0.3468	-40.5347	-21.6556	-324.486
8	-2.12136	-2.03346	0.336027	-38.4258	-25.2784	-328.716	-2.12307	-2.05649	0.341505	-42.5503	-20.9817	-325.938
9	-2.11831	-2.06346	0.336292	-40.094	-20.4125	-325.165	-2.12182	-2.05966	0.338121	-40.6444	-20.2619	-325.448
10	-2.13863	-2.05434	0.336408	-45.035	-18.626	-326.497	-2.12628	-2.06107	0.346039	-41.7946	-20.2775	-326.873

11	-2.1325	-2.05355	0.337062	-43.3323	-22.9461	-326.045	-2.12717	-2.05656	0.354431	-41.1514	-20.8605	-325.277
12	-2.12944	-2.02761	0.365014	-40.4572	-22.6789	-329.225	-2.12589	-2.05345	0.357651	-41.8184	-20.9923	-326.913
13	-2.11596	-2.05899	0.34992	-40.9497	-22.0826	-325.248	-2.11689	-2.06224	0.357611	-42.3808	-20.1612	-324.793
14	-2.17343	-2.07632	0.342407	-51.08	-13.6088	-322.693	-2.12501	-2.0574	0.366453	-43.2586	-20.8156	-326.045
15	-2.11652	-2.02921	0.360924	-40.1537	-22.4423	-326.68	-2.12087	-2.04837	0.351113	-42.5208	-21.5045	-325.503
16	-2.11939	-2.06646	0.35197	-41.8783	-23.2284	-328.861	-2.12236	-2.05661	0.348429	-41.448	-21.7205	-327.354
17	-2.11534	-2.05918	0.353992	-42.2215	-20.6144	-323.855	-2.11848	-2.05785	0.349713	-41.0827	-20.1914	-323.299
18	-2.12253	-2.05988	0.320453	-40.9741	-23.4047	-324.454	-2.12441	-2.06385	0.326739	-41.6107	-21.5405	-324.796
19	-2.10678	-2.05229	0.327039	-43.689	-25.0566	-322.45	-2.12403	-2.05727	0.346089	-45.1231	-21.3747	-324.654
20	-2.08281	-2.03669	0.340394	-39.0336	-19.8293	-323.718	-2.10789	-2.06265	0.34891	-40.6779	-20.8631	-325.392
21	-2.11088	-2.05705	0.345358	-42.443	-22.9518	-327.948	-2.11915	-2.06294	0.357169	-41.2088	-20.3599	-324.414
22	-2.10112	-2.04552	0.313626	-43.4607	-28.2887	-323.641	-2.114	-2.06169	0.34209	-42.6855	-21.459	-322.609
23	-2.14673	-1.90453	0.486972	-36.0027	-27.879	-318.465	-2.11518	-2.05852	0.357872	-41.6146	-20.6188	-322.362
24	-2.14256	-2.03309	0.278594	-42.6863	-28.2126	-324.917	-2.12355	-2.0586	0.332215	-41.7997	-22.165	-325.636
25	-2.13005	-2.06997	0.325675	-42.2495	-20.4056	-324.526	-2.1205	-2.05749	0.343458	-40.9568	-21.1774	-325.393
26	-2.12818	-2.04459	0.281767	-47.1191	-22.8293	-324.403	-2.11275	-2.05504	0.340258	-40.2801	-21.8602	-324.388
27	-2.11458	-2.04181	0.370176	-37.7874	-17.4634	-323.198	-2.11655	-2.0657	0.358095	-41.2503	-19.1004	-324.972
28	-2.12292	-2.05288	0.335977	-41.2092	-22.8458	-324.042	-2.11988	-2.06325	0.34401	-41.1864	-21.1251	-324.755
29	-2.15067	-2.0275	0.388153	-37.2783	-19.8547	-329.565	-2.1173	-2.05918	0.346163	-39.7687	-19.8016	-324.543
30	-2.13034	-2.06362	0.266525	-41.2212	-24.8789	-323.02	-2.12255	-2.06325	0.349059	-41.1126	-20.2625	-325.022
Avg.	**-2.1219**	**-2.0448**	**0.33671**	**-41.537**	**-22.862**	**-325.06**	**-2.1201**	**-2.0594**	**0.34819**	**-41.763**	**-20.864**	**-325.03**

Results using Proposed Method in Layout 3												
Index	**Kronecker Product**						**Dual Quaternion**					
	Rx (rad)	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**	**Rx (rad)**	**Ry (rad)**	**Rz (rad)**	**Tx (mm)**	**Ty (mm)**	**Tz (mm)**
1	-2.11814	-2.0641	0.330199	-40.8625	-21.7122	-323.038	-2.12155	-2.06754	0.338842	-41.3588	-21.0269	-323.9
2	-2.1162	-2.06297	0.33757	-39.9142	-21.2931	-322.884	-2.123	-2.06217	0.340018	-40.8833	-21.0154	-323.35
3	-2.1162	-2.06407	0.340671	-40.3567	-21.1778	-324.539	-2.11823	-2.06835	0.34045	-41.3145	-21.0806	-324.358
4	-2.11918	-2.06561	0.339846	-40.8237	-21.6925	-323.915	-2.11509	-2.07027	0.341571	-41.2855	-20.9025	-323.742
5	-2.12143	-2.07141	0.333943	-41.341	-20.9826	-323.552	-2.11456	-2.0721	0.340825	-41.355	-20.742	-323.662
6	-2.11727	-2.0639	0.337278	-40.4516	-21.4672	-323.201	-2.1201	-2.06786	0.341841	-41.151	-20.8279	-324.026

7	-2.11686	-2.06394	0.345338	-40.1629	-21.0768	-323.76	-2.11984	-2.06704	0.338421	-41.2391	-21.0674	-324.094
8	-2.12376	-2.0682	0.329312	-41.3963	-22.024	-322.642	-2.1217	-2.07463	0.332573	-41.5271	-20.7574	-323.074
9	-2.11583	-2.06535	0.333526	-40.7027	-21.3366	-322.95	-2.11365	-2.07325	0.342172	-40.7762	-20.7478	-323.625
10	-2.11741	-2.06144	0.338349	-41.0457	-21.7982	-324.355	-2.11014	-2.06952	0.343767	-40.7967	-21.7786	-324.601
11	-2.11867	-2.06246	0.340476	-40.3965	-21.516	-323.208	-2.11973	-2.06546	0.343059	-41.1542	-20.9385	-323.869
12	-2.11898	-2.06691	0.33945	-40.9303	-20.7918	-323.047	-2.11714	-2.06905	0.343762	-41.255	-20.8203	-323.847
13	-2.11796	-2.06564	0.34184	-40.7602	-21.5119	-323.608	-2.1167	-2.06887	0.34354	-41.0086	-20.9294	-323.739
14	-2.12031	-2.06914	0.334426	-41.2622	-21.068	-323.571	-2.11639	-2.07044	0.339234	-41.2399	-20.9555	-323.923
15	-2.11516	-2.06289	0.348114	-39.6736	-21.0305	-323.716	-2.12082	-2.06905	0.339708	-41.3155	-20.8894	-323.686
16	-2.11943	-2.06739	0.335507	-41.3421	-21.7145	-324.923	-2.11418	-2.06913	0.340932	-41.0606	-21.3088	-324.246
17	-2.11608	-2.06093	0.345567	-39.3302	-20.9626	-322.371	-2.11645	-2.07148	0.343393	-41.0704	-20.9129	-323.061
18	-2.11025	-2.06385	0.360466	-39.1793	-20.576	-324.484	-2.11305	-2.0709	0.342134	-40.9815	-20.7407	-324.084
19	-2.12081	-2.06663	0.343181	-41.0177	-21.0032	-323.165	-2.12334	-2.07383	0.341375	-41.4767	-20.5293	-323.31
20	-2.11431	-2.06633	0.338323	-40.686	-20.8965	-323.295	-2.11823	-2.06767	0.340074	-41.1832	-20.6685	-323.552
21	-2.11866	-2.06969	0.340601	-40.2972	-20.6489	-323.19	-2.1211	-2.06829	0.341228	-41.1661	-20.8228	-323.718
22	-2.12015	-2.06574	0.33805	-41.174	-21.2528	-323.342	-2.11823	-2.06959	0.340708	-41.2987	-20.9715	-323.64
23	-2.11503	-2.06525	0.337225	-40.6435	-21.2375	-323.369	-2.11764	-2.07517	0.334439	-41.4689	-21.2006	-323.445
24	-2.11819	-2.06319	0.338659	-40.6844	-21.2899	-323.677	-2.12187	-2.06943	0.336055	-41.7919	-21.19	-323.874
25	-2.1201	-2.06783	0.337041	-41.0326	-20.8856	-322.22	-2.11498	-2.07366	0.339511	-41.4256	-20.8796	-323.17
26	-2.1421	-2.06617	0.279762	-41.8732	-20.6704	-321.282	-2.12812	-2.04977	0.325046	-42.3889	-22.4512	-325.03
27	-2.11831	-2.06363	0.338854	-40.452	-21.3925	-323.31	-2.12187	-2.06798	0.339375	-41.4217	-21.094	-323.505
28	-2.11975	-2.0673	0.329424	-41.4759	-21.2359	-322.863	-2.11709	-2.06927	0.342536	-41.108	-21.0123	-323.374
29	-2.12187	-2.07004	0.333938	-41.5164	-21.0091	-324.011	-2.11618	-2.07059	0.338358	-41.2284	-20.9947	-323.928
30	-2.11667	-2.06532	0.341614	-40.8017	-21.0817	-323.702	-2.11642	-2.07325	0.338369	-41.3643	-21.0781	-323.93
Avg.	**-2.1188**	**-2.0656**	**0.33695**	**-40.72**	**-21.211**	**-323.37**	**-2.1182**	**-2.0692**	**0.33978**	**-41.27**	**-21.011**	**-323.78**