AN OPTICAL METHOD FOR DETERMINING POSITION AND ORIENTATION OF MULTIPLE MARKERS IN 3D SPACE

DAREENA DARACHAI

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2010**

Thesis
entitled

AN OPTICAL METHOD FOR DETERMINING POSITION AND ORIENTATION OF MULTIPLE MARKERS IN 3D SPACE

Dareena Darachai
....
Miss Dareena Darachai
Candidate

....
Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Major-advisor

....
Asst. Prof. Warakorn Charoensuk,
Ph.D. (Electrical Engineering)
Co-advisor

....
Asst. Prof. Jackrit Suthakorn
Ph.D. (Robotics)
Co-advisor

B. Mahai
....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

....
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Program Director
Master of Science Program in
Technology of Information System
Management
Faculty of Engineering
Mahidol University

Thesis
entitled

AN OPTICAL METHOD FOR DETERMINING POSITION AND ORIENTATION OF MULTIPLE MARKERS IN 3D SPACE

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)
on
June 11, 2010

Dareena Darachai
....
Miss Dareena Darachai
Candidate

....
Asst. Prof. Banlur Emaruchi,
Ph.D. (Civil Engineering)
Chair

....
Asst. Prof. Warakorn Charoensuk,
Ph.D. (Electrical Engineering)
Member

....
Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Member

Supanat.
....
Mr. Supanat Kitcharoensakkul,
Ph.D. (Computer Science)
Member

....
Asst. Prof. Jackrit Suthakorn,
Ph.D. (Robotics)
Member

B. Mahai
....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

....
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

I would like to acknowledge and express gratitude to all those who gave me the possibility to complete this thesis.

First of all, I greatly appreciated Dr. Songpol Ongwattanakul, my major-advisor, who showed the way of this research and encouraged me to go ahead with my thesis. Thank you for his numerous advice, support, and valuable time spent editing this thesis as well as his good friendship.

I am especially grateful to my co-advisor, Asst. Prof. Warakorn Charoensuk and Asst. Prof. Jackrit Suthakorn, whose help, stimulating suggestions, explaining and re-explaining of important issues helped me all the time of my thesis.

I also would like to special thank Asst. Prof. Banlur Emaruchi and Dr. Supanat Kitcharoensakkul for providing valuable advice for the improvement and editing this thesis and participating in the defense.

I wish to thank all the teachers of The Technology of Information System Management Program for providing me their knowledge and suggestions. Thanks also go to the staff members of the Program for their kindness and support.

I would like to give my special thanks to my friends at The Technology of Information System Management Program for suggestion, inspiration and a lot of help.

Many thanks go to Miss Siripattra Laoapasuwong for their love, support and all the entertaining time that we have been shared.

Finally, the support and encouragement from my family were appreciated and greatly needed. Thank you for always believing in me.

Dareena Darachai

AN OPTICAL METHOD FOR DETERMINING POSITION AND ORIENTATION OF MULTIPLE MARKERS IN 3D SPACE

DAREENA DARACHAI 4737841 EGTI/M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORY COMMITTEE: SONGPOL ONGWATTANAKUL, Ph.D., WARAKORN CHAROENSUK, Ph.D., JACKRIT SUTHAKORN, Ph.D.

ABSTRACT

Human vision cannot precisely determine the 3D coordinates of an object. The optical tracking with marker-based system, which is capable of capturing position and orientation of an object, is implemented to correct this error. Medical applications such as image-guided surgery will benefit from this system, which can identify the position and orientation of a surgical device relative to the markers. This thesis provides and demonstrates a design framework for combining multiple sources of information. Initially, the system was implemented using MATLAB® to test the algorithm, so that the experiment simulates the actual condition at work. To create the image marker in 3D 2 images are combined with image markers from the left and right camera. Each image consists of 12 spheric markers that must have different length. The image is taken to the image processing in order to remove the unwanted parts in the image. After that, the computer calculates to find the position and orientation of the marker in 3D by using Fundamental Matrix. The position will be used to show the result in 3D graphic. Translation errors were tested for mistakes in the process, and the positions of markers were compared from the experiment. The Root Mean Square Error was less than 0.5 mm.

KEY WORDS: 3D COORDINATION / INFRARED OPTICAL TRACKING / MULTIPLE MARKERS / FUNDAMENTAL MATRIX

108 pages

กรรมวิธีทางแสงสำหรับการหาตำแหน่งมาร์กเกอร์ในสามมิติ

AN OPTICAL METHOD FOR DETERMINING POSITION AND ORIENTATION OF MULTIPLE MARKERS IN 3D SPACE

ดารีณา ดาราฉาย 4737841 EGTI/M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : ทรงพล องค์วัฒนกุล, Ph.D., วรากร เจริญสุข, Ph.D., จักรกฤษณ์ ศุทธากรณ์, Ph.D.

บทคัดย่อ

โดยทั่วไปการมองเห็นของมนุษย์ไม่สามารถบ่งบอกตำแหน่งในสามมิติของวัตถุใดๆ ได้อย่างแม่นยำ ระบบวัดพิกัดวัตถุมาร์กเกอร์ด้วยภาพจึงได้ถูกพัฒนาขึ้นเพื่อเพิ่มความแม่นยำในการระบุตำแหน่งดังกล่าว การหาพิกัดและการวางตัวใน 3 มิติของมาร์กเกอร์ สามารถนำมาประยุกต์เป็นต้นแบบเพื่อใช้ในระบบนำทาง เพื่อการผ่าตัดได้โดยการสร้างความสัมพันธ์ระหว่างตำแหน่งมาร์กเกอร์ และตำแหน่งพิกัดของอุปกรณ์ผ่าตัด ระบบวัดพิกัดวัตถุมาร์กเกอร์ด้วยภาพจะนำเสนอระบบและวิธีการ เพื่อให้สามารถระบุตำแหน่งพิกัดและทิศทางของมาร์กเกอร์ได้ โดยได้ทดลองระบบเพื่อทดสอบอัลกอริธึมด้วย MATLAB® ซึ่งการทดลองอัลกอริธึมนั้นได้จำลองการทดลองแทนการทำงานจริงโดยการสร้างภาพมาร์กเกอร์ในสามมิติขึ้นมา 2 ภาพ คือ ภาพมาร์กเกอร์จากกล้องทางด้านซ้ายและขวา ภาพมาร์กเกอร์ทั้ง 2 ภาพ จะประกอบไปด้วยมาร์กเกอร์ทรงกลม 12 มาร์กเกอร์ และแขนของมาร์กเกอร์จะต้องมีความยาวที่ไม่เท่ากัน ภาพจะถูกนำเข้าสู่กระบวนการ Image Processing เพื่อกำจัดส่วนที่ไม่ต้องการออกจากภาพ หลังจากนั้นการหาคำนวณหาตำแหน่งและทิศทางของมาร์กเกอร์ในสามมิติจะใช้วิธีการของ Fundamental Matrix ค่าพิกัดที่ได้จะนำมาแสดงผลเป็นกราฟสามมิติ นอกจากนี้ได้ทำการตรวจสอบค่าความผิดพลาดของระบบโดยการเปรียบเทียบตำแหน่งของมาร์กเกอร์ที่คำนวณได้ กับตำแหน่งของมาร์กเกอร์ที่ตั้งไว้ โดยการหาค่า Root Mean Square Error ต้องอยู่ในเกณฑ์ไม่เกิน 0.5 mm.

108 หน้า

CONTENTS

CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont.)

CHAPTER I
INTRODUCTION

1.1 Problem Statement

Recent medical applications such as a surgical navigation system introduce a great deal of engineering challenges. This navigation system assists the surgeon during the operation so that he or she can coordinate the movement with precision even though the surgery is invisible to the naked eyes. Dental implantation is one of the applications that require such coordination which allows the surgeon to pinpoint the drill bit at a designated location and orientation in relative to the patient’s head. The location and orientation is normally given by a path-planning application. The hand of a dentist who navigates the drill bit may require positioning and orientating accuracy beyond human capacity to complete the surgery exactly as plan. Therefore, an optical 3D coordination system can help dentist to locate the site at the right orientation.

The main purpose of a tracking system is to determine the three-dimensional coordinate of an object. There are many types of 3-D tracking, for example, acoustic tracking, mechanical tracking, magnetic tracking, and optical tracking. Each method has different advantages and disadvantages in aspects of performance and price as discriminating in Table 1.1

Table 1.1 Comparison of tracking techniques

Detection Techniques	**Description**	**Advantages**	**Disadvantages**
Acoustic Tracking	Based on measurement of time-of-flight of a sound signal from an emitter to a receiver.	- Inexpensive	- Low accuracy - Suffer from acoustic reflections in surrounded conditions
Mechanical Tracking	Object is connected to a mechanical armature. When the object moves, its position is known from related device movement.	- Fast - High accuracy	- Encumbered movement - Limited working area
Electromagnetic Tracking	Use the distortion of electromagnetic signals to determine position and orientation relative to source coils.	- Require no contact to the sensor - Not require direct line of sight between the source and sensor - High accuracy	- High cost - Accuracy affected by electromagnetic generator devices such as motor - Accuracy affected by Earth's magnetism

Table 1.1 Comparison of tracking techniques (cont.)

Detection Techniques	**Description**	**Advantages**	**Disadvantages**
Optical Tracking	Use 2 or more cameras to detect visual signals and involve camera information and epipolar geometry to calculate positions	- Require no contact to the sensor - Easy to develop - High accuracy - inexpensive	- Limited by light source intensity - Require clear line of sight

The system that we are interested is the optical tracking, which involves imaging with cameras mounted at a fixed location and image processing. Optical tracking has many benefits over other tracking. For instance, it offers a wireless interaction, presents accuracy measurements even with low cost, and requires no additional sensors when there are additional tracking objects. However, there are some disadvantages from applying the optical tracking system. For example, it may have an occlusion problem, needs computational time, and may require specific set-up on the environment such as the illumination on the scene. The optical tracking is widely used in many applications for example; augmented reality, computer animation, human motion analysis, and image guided-surgery. There are various methods to detect objects such as color-based, shape-based, feature-based method.

1.2 Introduction to Dental Implants

The Dental Implant is the knowledge of dentistry for replace of the lost teeth. By planting root replacement on the jawbones .The Dental Implant ease for fix the denture such as the removable denture and fixed partial denture.

1.2.1 The anatomy of implants

The Dental Implant is consist of 3 important portion

1.2.1.1 Fixture made of titanium, it similar a root of a tooth, it is implanting in the jawbones which intimately fixed. For this reason, it no have a symptom of inflame of tissue and no have an adverse effect to the health of patient.

1.2.1.2 Abutment, to wait for 4-6 months after the dental implant on the jawbones for the implant connects to the jawbones. Then put of the abutment on the implant.

1.2.1.3 Crown, is a part of the top of the gum ridge, it made of the porcelain which most similar the teeth.

1.2.2 The method of the dental implant

At first, the dentist will X-Ray the patient for assess the thick of soft tissue, it on the gum ridge .Then operate for implant the fixture into the bones and sew up a wound. After 7 days, cut the silk and for the implant connect to the jawbones, about 3-6 months for the top teeth and 2-3 months for the bottom teeth. Second, the dentist will connect the abutment for used to support the crown and printing a mouth for send to lab for make a crown .The last, after that about 1-4 weeks, the dentist will wearing a crown which a porcelain crown.

In this thesis, the focus is on the optical tracking system that detects multiple markers. The system includes an imaging subsystem and markers. This imaging system employs stereoscopic vision which composes of two cameras that are highly sensitive in infrared region. One marker is attached at a fix location on the surgical instrument and another one is attached to the patient. The goal of the system is to accurately identify the 3-D position of the markers from two point-of-views that are simultaneously captured. Marker detection is based on infrared optical tracking, which can perfectly distinguish the markers and background in the surgical environment.

1.3 Objectives

The main objective of this research is to find appropriate algorithms that are suitable for the implementation of surgical navigation applications. The mission is to develop fast and accurate 3-D coordinate algorithm experimental platform that allows the measurement of position and orientation errors. An extensive error study and analysis will be conducted to ensure the robustness of the system functionality and accuracy.

1.4 Scope of work

The infrared optical tracking by simulated stereoscopic cameras will be setup as a prototype. The camera system will be placed with fixed distance. The markers in scene are no shade. Our tracking system will be able to track specific markers and identify their 3-D positions and orientations. These pose estimations can be correlated to the position of the defined tool.

1.5 Expected results

The expected results include

1.5.1 Statistical analysis of :

a. Effectiveness defined by precision and accuracy,
b. Efficiency defined by computation time

CHAPTER II
LITERATURE REVIEW

2.1 Introduction

Every image acquired by a camera is distorted. The distortion is caused by the lenses, improper manufacturing and positioning of the camera sensor and even by temperature and vibration [Nal93]. Figure 1.1 depicts forming of an image of a scene. The scene is observed by a camera. Due to a lower cost, an analogue camera is typically used which required a frame grabber to convert an analogue signal into an appropriate digital form. The *line jitter* distortion caused by different sampling frequencies in the camera and the frame grabber is introduced during the conversion. After the digitalization, the image is ready to be displayed and visualized by a computer**.**

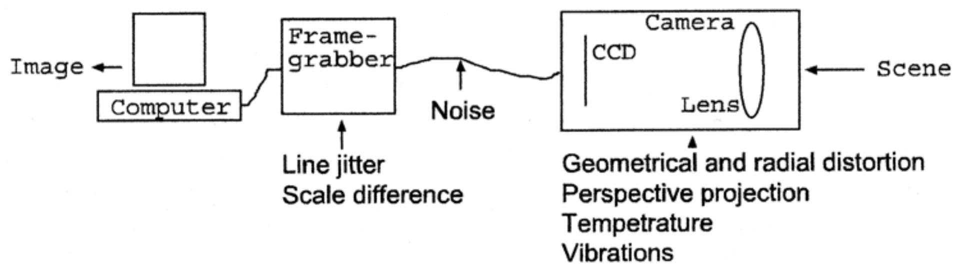


Figure 2.1 A scheme of an image forming

A mapping of a 3D scene into a 2Dimage, called perspective projection, can also be considered as a distortion because it does not preserve angles and distances. The *perspective projection* of a square is a general quadrangle as demonstrated in Figure 2.2

All the mentioned distortions are linear because they can be expressed by a linear (matrix) algebra. But there exist nonlinear distortions too. The first of them, *radial distortion*, displaces points in the image plane inwards to or

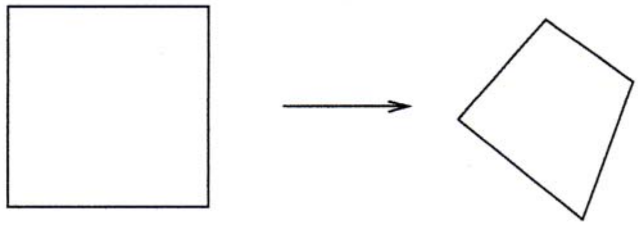


Figure 2.2 Perspective projection of a square

Outwards from its optical center. The displacement is a function of a distance from the center and it is circular symmetric. If the points are shifted towards the center, the distortion is called *barrel*, otherwise *pincushion*. These names were chosen after the shapes to which a square is transformed by influence of this distortion, see Figure 2.3

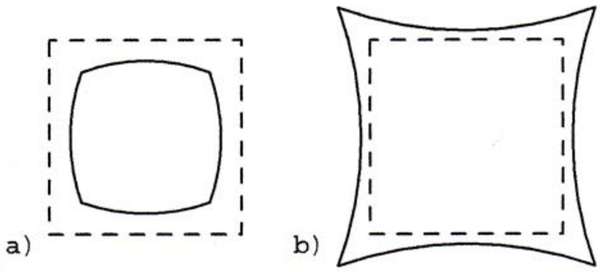


Figure 2.3 Radial distortion: a) barrel, b) pincushion

Another group of nonlinear distortions, called *lens distortions*, generally have two components, radial and tangential, as depicted in Figure 2.4 The most significant lens distortion, called *decentring*, is due to the fact that the optical centers of lens elements in composed lenses are not strictly collinear. Another one, *thin prism distortion*, arises from imperfect camera assembly.

Besides these *geometrical distortions* there exist so called *radiometrical distortions* which characterize various luminous degradations during the acquisition of images. An overview of all the distortions caused by a real camera is given in Figure 2.5

To acquire information from an image of a scene such as shapes of objects, distances between them, color characteristics of the scene and so on, the camera *has to be calibrated*. A *camera calibration*[9] is a process of acquiring knowledge about the relation between an image of a scene taken by the camera and the scene itself, usually including the position of the camera in the scene and its internal characteristics.

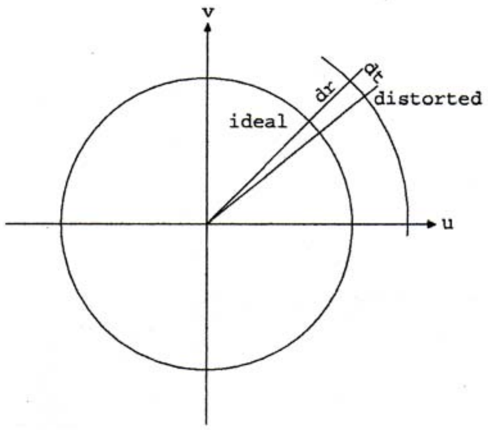


Figure 2.4 Radial (dr) and tangential (dt) components of non-linear lens distortions [25]

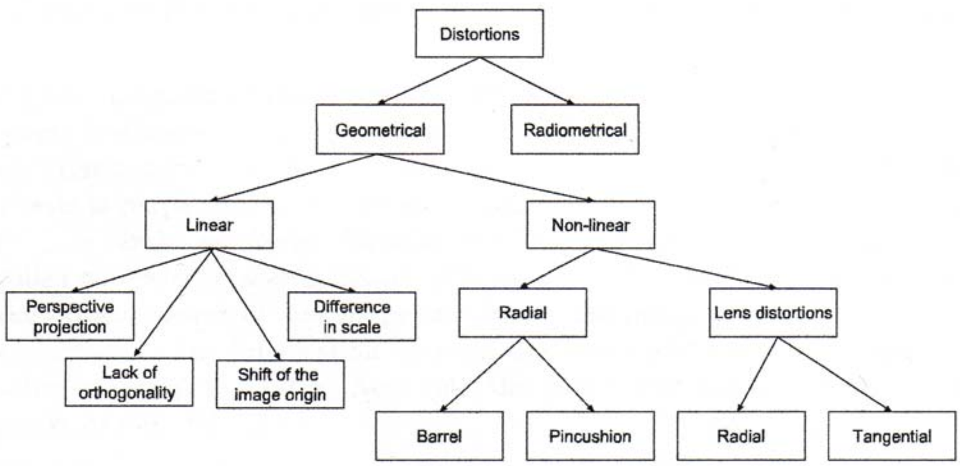


Figure 2.5 An overview of distortion caused by a real camera

The calibration consists of two stages: *geometrical* and *radiometrical*. The radiometrical calibration[12],[3] is performed to acquired the information on how the camera distorts the luminous characteristics of the scene such as color, luminance, brightness etc. The geometrical calibration leads to knowledge of rotation and position of the camera (commonly called the *extrinsic camera parameters*) and its internal characteristics (*intrinsic camera parameters*) such as a focal length, position of the principal point, difference in scale of the image axes and so on.

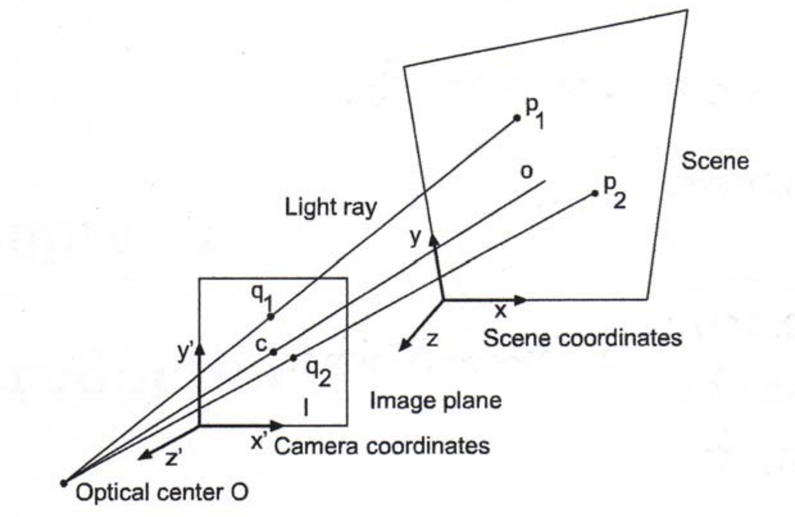


Figure 2.6 Projection of a scene into an image

Figure 2.6 demonstrates the projection of a scene into an image. Pi represents points in a scene and qi are the corresponding points in the image plane. The coordinate system of the camera (x1, y1, z1) is chosen in such a way that the z1 axis is perpendicular to the image plane I. The scene coordinate system (x, y, z) can be set freely. All light rays from a scene intersect in one point, called and optical center or center of perspective O. The line o going through the optical center O and perpendicular to the image plane I is called an optical axis. The intersection between the axis o and the image plane I is called a principal point c. Note that this point does not have to be in the center of the CCD array.

The geometrical distortions significantly change positions of points in the image of a scene acquired by a camera. Therefore it is important to compensate such displacements. The next chapter is focused on how this is accomplished.

2.2 Geometrical Calibration of a Camera

As noted before, the calibration process consists of radiometrical and geometrical stages. The radiometrical camera calibration is typically neglected in standard computer vision methods, although it plays a very important role in special application such as astronomy imaging, photometry, color processing and so on. On the other hand, the geometrical calibration has to be performed for most of the vision tasks.

During the geometrical camera calibration, an image of some special scene is acquired. The scene contains so called calibration objects or targets, whose shapes and dimensions are a priori known. On these objects, there are visually detectable features whose coordinates are used for the calibration. It should be noted that there also exist so called self-calibration methods which do not require explicitly known calibration objects[28],[23]. Nevertheless, the known calibration target is assumed in this work.

In this Chapter, calibration objects and various types of control points are discussed first. Then, existing camera models are presented and compared.

2.3 Calibration Objects

For the camera calibration, a set of scene-image coordinate pairs has to be provided. The scene coordinates are measured directly on the calibration targets, the corresponding image coordinates has to be detected in the image. In order to allow such a detection, the targets are equipped with visually significant features painted on them. These features, such as intersections of Lines, centers of gravity of circles, corners of squares and so on are then treated as control (reference) points. After detecting the points in the image and establishing the relation between their scene and image coordinates (called matching) the camera is ready to be calibrated.

The calibration objects can be three dimensional, virtual 3D (simulated by multiple views of a 2D target) or two dimensional (also called coplanar) as depicted in Figure 2.7 The 3D calibration objects are difficult to manufacture, also the measurement of the coordinates of the control points in 3D space is complicated. On the other hand, such objects allow a precise calibration. In addition, most of the calibration methods require the 3D calibration objects in order to estimate the complete set of the camera parameters.

There is a possibility to simulate a 3D calibration object by multiple views of a 2D calibration target. The 2D plane can be moved freely between the particular image acquisitions[11] or the movement can be constrained. An usual constraint is that the plane is moved only along one direction. The main disadvantage of this approach is that the plane has to be moved precisely on a specified path and it should not be rotated or shifted during the movements. This demand can be ensured by manipulating the plane by a robot or with a help of special equipments (such as precisely made holders), Both not so commonly available yet.

If a coplanar calibration object is used, the full set of camera parameters cannot be acquired without some a priori knowledge[16]. For example if the focal length is not known, then the distance of the camera from the calibration object cannot be determined. In addition, when choosing initial estimates of the parameters, errors are introduced into the calibration process leading to inaccurate calibration results.

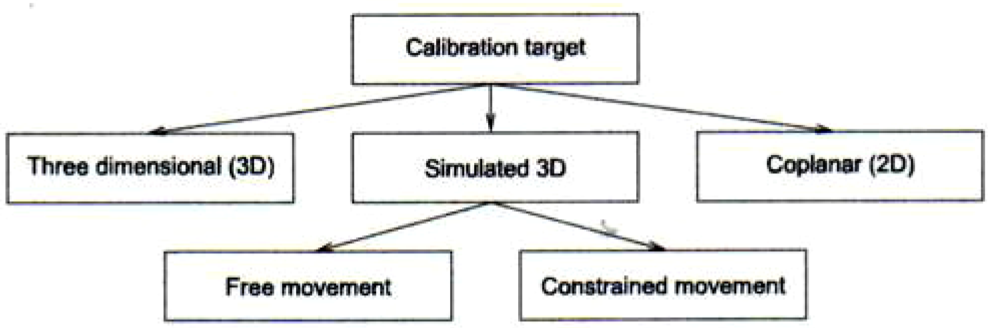


Figure 2.7 Types of calibration objects

2.4 Control Points

The control points are typically represented by visually detectable features painted on the surface of the calibration object. The points can be expressed for example as centers of gravity, intersection of lines, corners of squares or by a checkerboard as depicted in Figure 2.8

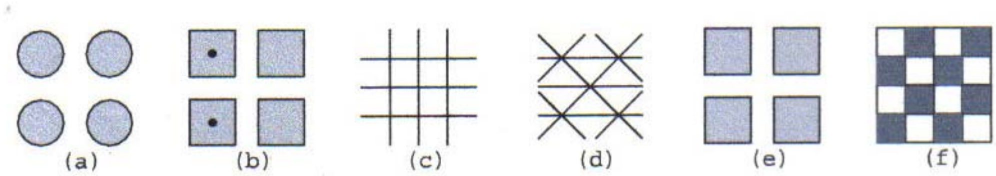


Figure 2.8 Variants of control points: centers of gravity of circles (a) and squares (b), intersections of lines in rectangular (c) and triangular (d) meshes, corners of squares (e) and checkerboard (f). The points are marked as dots

Positions of the control points in the acquired images can be detected manually or by some of the auto-detection algorithms such as[5],[6]or[20]. The automatic detection methods typically require black-and-white images. Because the camera provides gray scale or color images, these images have to be converted first. The conversion, called thresholding is very sensitive to the correct setting of the threshold value. Improper thresholding makes the determination of correct positions of control points in the threshold image difficult.

For example, the line crossings method fails when the thresholded lines are too thin so they can be corrupted by noise or their parts can be invisible as depicted in Figure 2.9(a). On the other hand, when the lines are too thick and span over several pixels as demonstrated in Figure 2.9(b), their intersection cannot be properly determined

Figure 2.9 Improper thresholding of lines: (a) underthresholding causes corruptedand non-continuous lines. (b) overthresholding leads to lines spanning multiple pixels

A similar thresholding problem also applies to corners of squares. If the squares touch each other in the calibration grid (like in a checkerboard) then their thresholded images can overlap or can be separated as depicted in Figure 2.10(a). In such situations, the corners cannot be detected directly. If the squares do not form a checkerboard, their images after improper thresholding can have different sizes as demonstrated in Figure 2.10(b). The corners of such squares are the detected in wrong positions.

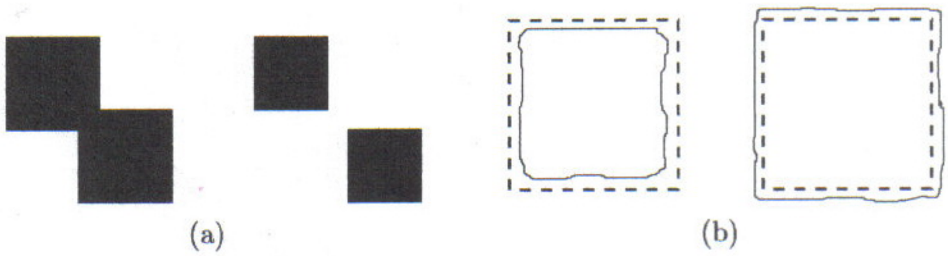


Figure 2.10 Improper thresholding of a checkerboard (a) and separated squares (b)

A detection of centers of gravity of circular or square targets is not so sensitive to the thresholding errors but it introduces another problem. A perspective projection of a square is a general quadrangle and a circle is projected to an ellipse. In both cases, the center of gravity of the original object is not the same as the center of gravity of its perspective projection. Figure 2.11(a) depicts this situation for a square, Figure 2.12(b) for a circle. A method to solve this problem was introduced in [10].

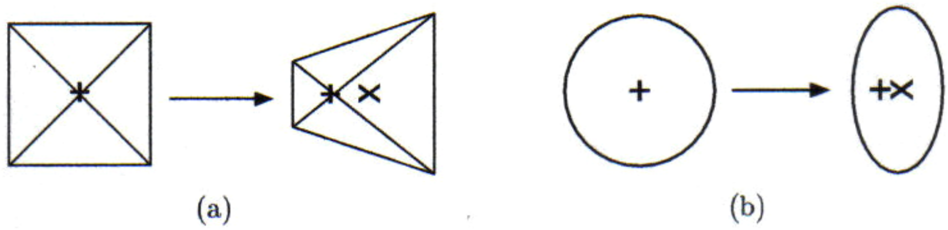


Figure 2.11 Differences between centers of gravity of an object (+) and its perspective projection (x): a) square, b) circle

The proper choice of the calibration target depends on the size of the calibration object, focal length of a camera, distance between the object and the camera, light conditions in the scene and many other factors. An overview of various calibration targets with their properties is summarized in Table 2.1

Table 2.1 A comparison of various calibration targets

Target shape	Control points	Example Figure	Main advantage	Main disadvantage	Problem Figure
checkerboard	corners	2.2(f)	lower sensitivity to thresholding errors, large shape	overlapped or separated squares after thresholding	2.4(a)
squares	corners	2.2(e)	large shape allows larger distance from camera	thresholding can change position of a control points significantly	2.4(b)
squares	centers of gravity	2.2(b)	low sensitivity to thresholding errors	center of gravity of the projection is not a projection of the center of gravity of the square	2.5(a)
circles	centers of gravity	2.2(a)	low sensitivity to thresholding errors	center of gravity of the projection is not a projection of the center of gravity of the circle	2.5(b)
lines	intersections	2.2(c),(d)	can be easily detected	successful thresholding depends on thickness of lines and distance between an object and a camera	2.3

2.5 Models of a Camera

During the calibration, a real camera is simulated by a theoretical model which describes how a scene is transformed into an image? There are various models of a camera with different capability to cover the camera characteristics. Many of them are based on physical camera parameters but there also exist models representing only a projection of the scene points into the image. The former are called explicit, the latter implicit camera models.

An example of an explicit camera model is so called DLT model[1] which provides the position of a camera in a scene, its focal length, principal? point and linear distortion coefficients. The extended DLT model[17] also simulates nonlinear lens distortions. On the other hand, the implicit camera models such as two-plane method[27] do not provide any physical camera parameters at all. An overview of camera models is presented in Figure 2.12

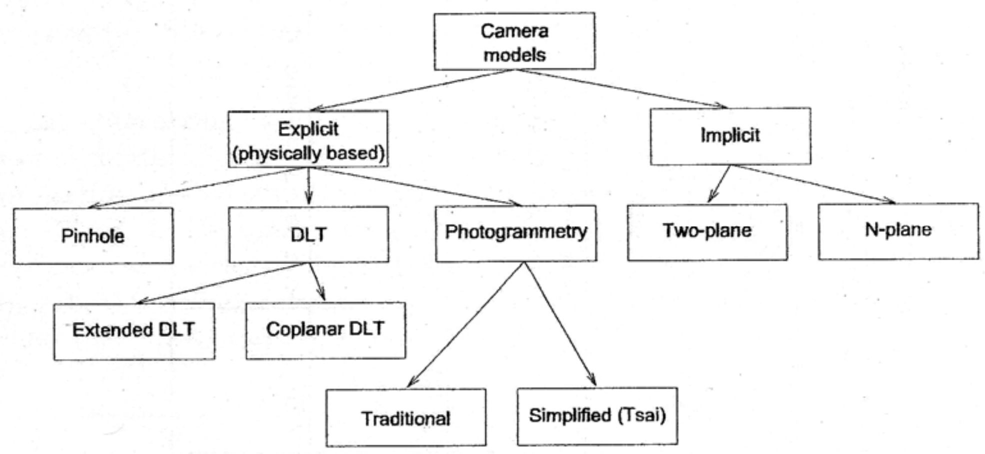


Figure 2.12 An overview of camera models

2.5.1 Pinhole Model

So called pinhole [8] is a model of an ideal camera. The pinhole model is very simple, in fact it represents only rotation and translation (together called rigid body transformation) of the camera followed by a perspective projection. Another distortion (such as shift of the image origin or lens distortions) are not taken into an account. Nevertheless, the pinhole gives a good approximation of a real camera and therefore it is used as a base for other calibration methods.

The model can be formally described as

$$q_i = FMTp_i \tag{2.1}$$

where, written in homogeneous coordinates, $p_i = [x_i, u_i, z_i, 1]^T$ are scene coordinates of the *i*-th point and $q_i = [w_i u_i, w_i u_i, w_i]^T$ are the appropriate coordinates in the image plane. The matrices F, M, T :

$$\mathrm{F} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \tag{2.3}$$

$$T = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \end{pmatrix} \tag{2.4}$$

Represent focal length, rotation and translation of the camera, respectively. The relations between the focal length f and the matrix **F** and the translation (x_0, y_0, z_0) and the matrix **T** are straightforward. On the other hand, the expression of the rotation in the matrix **M** is more complicated. The matrix is orthonormal and its columns express coordinates of the axes of the rotated coordinate system, but it has nine parameters and only three degrees of freedom.

A general rotation is typically expressed by_*Euler angles* as three sequential elementary rotations $\omega_x, \emptyset_y$ and k_z along the axes of the Cartesian coordinate system. Euler angles have an unwanted singularity when $\emptyset_y = \frac{\pi}{2}\ or\ \emptyset_y = \frac{3\pi}{2}$. This could be acceptable when the orientation of the camera is restricted in some directions. An alternative is to use *quaternions* (also called *Euler parameters*), which are four parameter singularity free representation of rotation. The quaternions have a descriptive geometrical meaning: if the axis of rotation is the unit vector $r = [r_x, r_y, r_z]^T$ and the angle of rotation along this axis is β, the rotation is represented by the quaternion

$$g = \left[cos\frac{\beta}{2}, r_x \sin\frac{\beta}{2}, r_y sin\frac{\beta}{2}, r_z sin\frac{\beta}{2}\right]^T \quad (2.5)$$

2.5.2 Direct Linear Transform (DLT) Model

So called *direct linear transform (DLT)*[1] is an extension of the pinhole model. In addition to the pinhole, DLT models lace of orthogonality between image axes, shift of the image origin and difference in scale of images axes. It can be formally described as

$$q_i = A\, p_i\,, \quad (2.6)$$

where the DLT matrix **A** of the size 3×4 is composed from primitive matrices **V, B, F, M, T** as follows:

$$A = \lambda\, V^{-1} B^{-1} FMT. \quad (2.7)$$

The factor $\lambda \neq 0$ express an overall scaling, the matrices **F, M** and **T** represent focal length, rotation and translation as in the pinhole model (Eq. 2.2, Eq. 2.3, Eq. 2.4), the matrix **V,**

$$V = \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_o \\ 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

represents the shift of the image origin and the matrix **B,**

$$B = \begin{pmatrix} 1 + b_1 & b_2 & 0 \\ b_2 & 1 - b_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

Compensates the difference in scale and lack of orthogonality between image axes. Note that the matrix **V** is clearly invertible and the matrix **B** is invertible when $b_1^2 + b_2^2 < 1$. The values of b_1 and b_2 are commonly in the order of 10^{-4} to 10^{-6}.

2.5.3 Coplanar DLT Model

When all object points are coplanar, that means they lie in one plane, the coplanar DLT model can be used. If the world coordinate system is chosen to satisfy z_i = 0 for all points, the third column of the DLT matrix **A** can be ignored, yielding the transformation (Eq. 2.6) in the form

$$q_i = A_{(3,3)} p_i, \tag{2.10}$$

where the coplanar DLT (CDLT) matrix $A_{(3,3)}$ of the size 3×3 is defined by Eq. 2.7, except that the translation matrix **T** has to be expressed as

$$\mathrm{T} = \begin{pmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & -z_0 \end{pmatrix} \tag{2.11}$$

2.5.4 Extended DLT Models

In his dissertation[17], T. Melen introduced an extension to the DLT model where the nonlinear compensation term δ models both radial and decentring distortion using five more parameters. The detected image coordinates are corrected by this term after the standard DLT model is estimated. An iterative approach to determine the term δ is proposed in the paper. J. Heikkil$\ddot{a}$ and O. Silven[11] use the DLT model as the initial guess for a nonlinear minimization of the following error function:

$$\sum_{i=1}^{n}(U_i - u_i)^2 + \sum_{i=1}^{n}(V_i - v_i)^2, \tag{2.12}$$

where n is the number of control points, (u_i, v_i) are the corresponding coordinates predicted by the model.

A similar approach is also applied in [15].An optimization is used to minimize the error function Eq. 2.12 with a help of multiple views of the same scene containing a 3D calibration object**.**

2.5.5 Traditional Photogrammetry Model

In photogrammetry, the extended pinhole model[21] is used to achieve high precision results. The correction of the shift of the image origin and the compensation term δ = $[\delta_u, \delta_v]$ are added to the original pinhole model. The term δ models both linear and nonlinear distortion, giving the transformation in the form:

$$q_i^l = v + \delta(q_i), \tag{2.13}$$

where $v = [u_0, v_0]$ is the center of the image plane, q_i is a perspective projection of an object point p_i and q_i^l is an appropriate point in the image plane. The accurate results are achieved at the cost of long computational time and precise calibration objects.

2.5.6 Simplified Traditional Models

Regarding the different tasks and available equipments, R.Tsai[24],[14] proposed a simplified traditional camera model suitable for computer vision applications. Only the radial distortion, shift of the image origin and differences in scale of the image axes are compensated by so called radial alignment constant. This camera model became very popular when its implementation in C was published freely on the Internet[26]. More precise model, which also covers the decanting and thin prism distortions, was introduced by J.Weng[25].

2.5.7 Two-plane Camera Models

All above methods try to estimate the physical camera parameters, but there are situations when only the knowledge on how the camera transforms a scene into an image is needed. This does not necessary require that the physical camera parameters to be estimated at all.

Two-plane camera models define a light ray by its intersections with two parallel planes in the object coordinate system. The appropriate mapping from the image plane to these planes is estimated by a special calibration procedure based on a dense grid of control points which should cover the largest possible area of the image plane to achieve reliable results. The mapping is typically described as a combination of power terms. More than two planes and B-spline patches can also be used [4]. In addition, he parallel planes can be simulated by a static camera equipped with zoom lenses as proposed in [7].

2.5.8 Comparison of Camera Models

The choice of an appropriate camera model depends mainly on the type of application. In the field of a computer vision, the use of CCD cameras instead of precise photogram metric equipments allows the simplified traditional or extended DLT models to be suitable for most cases. An important criterion is also the numerical stability of the particular model. When a nonlinear search is introduced during the calibration, there is a possibility that the optimization routine can stick in a local extremal point or it does not converge at all. A direct computation of the camera parameters parameters based on a linear model (the pinhole, standard or coplanar DLT) can serve as a good initial guess. A comparison of the camera calibration models presented in previous sections is given in Table 2.2

Table 2.2 Comparison of the camera calibration models

Camera model	Number of parameters	Advantages	Disadvantages
Pinhole	7	linear, simple, stable	does not model a real camera
DLT (Direct Linear Transform)	12	linear, stable, needs only 6 points to compute parameters	does not model non-linear distortions, needs 3D calibration object
CDLT (Coplanar DLT)	11	linear, stable, needs only 6 points to compute parameters, uses 2D calibration object	does not model non-linear distortions, needs an initial guess of some of the parameters
Extended DLT	14-18	models lens distortions	iterations or non-linear search
Traditional (Photogrammetry)	14+	accurate results	non-linear search, computationally expensive, needs precise calibration object
Simplification of traditional model (Tsai)	11-16	good results, source code of the method is freely available	non-linear search, depends on the data precision

Table 2.2 Comparison of the camera calibration models (cont.)

Camera model	**Number of parameters**	**Advantages**	**Disadvantages**
Two-plane	no physical parameters	possibility to use coplanar target, back projection can be computed	does not necessary model a real camera, accuracy strongly depends on the area of the image plane covered by the control points

2.6 Camera Calibration with a Virtual 3D Calibration Object

A new camera calibration method based on a virtual 3D calibration object is proposed. First, the motivation is presented. In the next Section, an overview of the proposed method is given. Then, particular steps of the method and its iterative refinement are described in more details. Finally, the proposed iterative approach is summarized and compared with other calibration methods.

2.6.1 Motivation

Various calibration approaches and camera models were characterized in the previous Chapter. In practice, the choice of a suitable calibration method depends mainly on the required precision of the calibration and on available resources. For the best results, the traditional photogrammetry methods have to be used. But these methods assume very fine manufactured and measured calibration objects. Which are not widely available yet.

As it was already motes, a 3D calibration object is required to achieve precise and reliable results. When a coplanar object is used, the accuracy of the calibration depends on the initial guess of some of the camera parameters. If a 3D object is simulated by a multiple views of a 2D plane, it is important whether the plane can be moved freely or the movement is constrained, The latter approach requires special and usually expensive equipments such as a robot.

From the mentioned facts it follows that the most effective approach is to exploit a virtual 3D calibration object simulated by unconstrained movements of a coplanar target. Current calibration methods based on this principle either include a nonlinear search over large number of parameters[11] or do not provide the physical camera parameters at all[27]. To overcome this deficiency, the proposed method is based on a geometrical construction of the object.

2.6.2 Overview of the Proposed Calibration Method

The proposed method uses multiple views of a 2D calibration target (80 called *calibration plate*) to simulate a 3D object, which is needed for precise and reliable calibration results. The calibration plate can be moved freely between the image acquisitions. An important assumption of the method is that the intrinsic parameters of the camera are constant for all views. This constraint allows the simulation of the 3D object to be based on relative positions of the plate between particular acquisitions.

The method consists of the following four steps (see Figure 2.13):

2.6.2.1 An initial estimation of the intrinsic parameters of a camera.

2.6.2.2 An estimation of the extrinsic camera parameters for each view.

2.6.2.3 A construction of a virtual 3D calibration object from multiple 2D planes.

2.6.2.4 A complete camera calibration based on the virtual 3D object.

Due to the use of a coplanar calibration target, an initial guess of the intrinsic camera parameters is needed. These parameters are supplied by the user in the first step of the proposed method.

In the second step, the extrinsic camera parameters are estimated for each view with respect to the provided intrinsic parameters. Any explicit coplanar camera calibration method can be applied here.

In the third step, the virtual 3D calibration object is constructed. The construction exploits the knowledge on the camera positions for all views provided by the previous step.

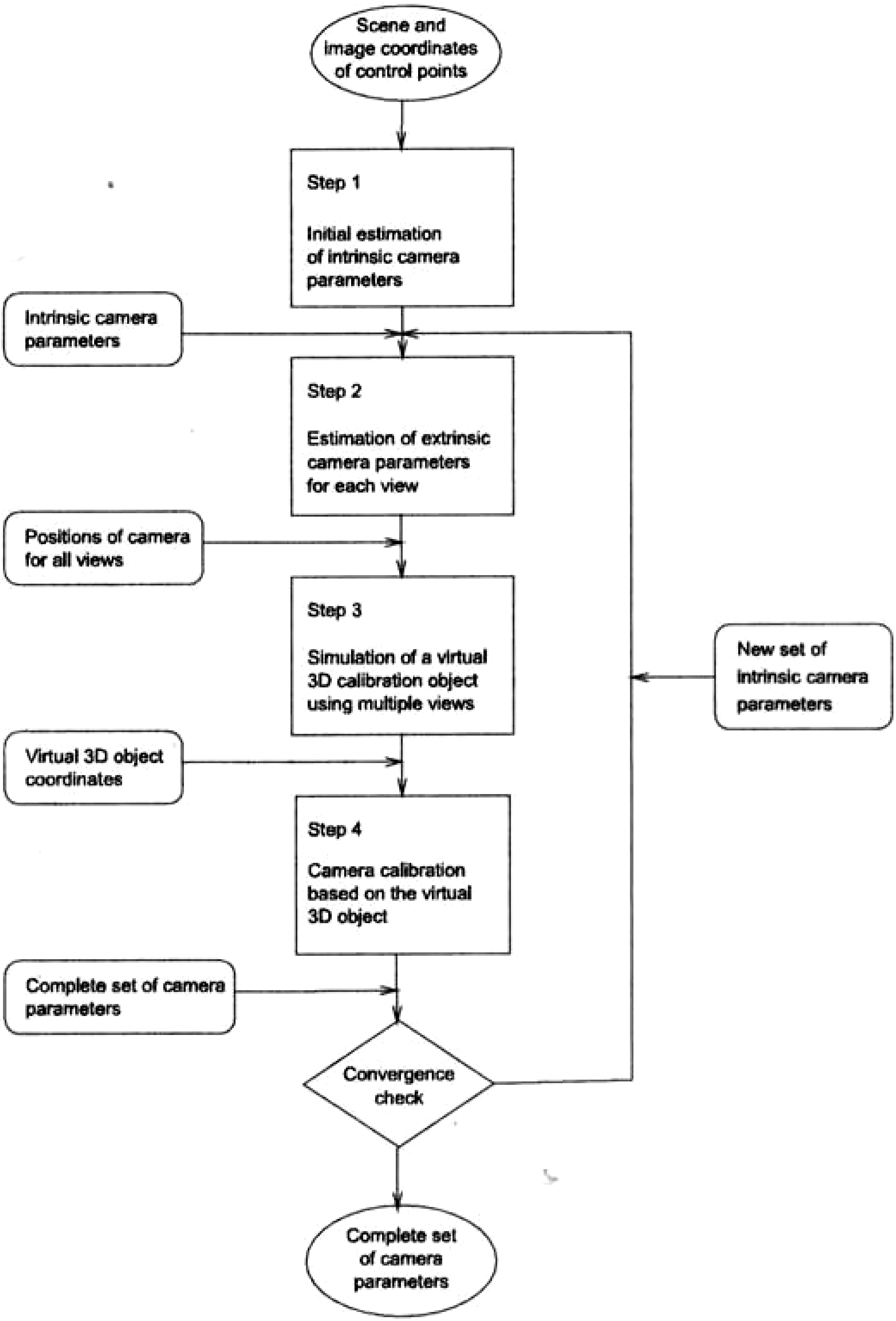


Figure 2.13 A scheme of the proposed calibration method

The pairs of scene and image coordinates of the control points of the simulated calibration object are finally passed to the fourth step of the proposed method, in which a complete set of the camera parameters is estimated. Any explicit 3D based camera calibration method can be used for this purpose.

As a result of the calibration, both intrinsic and extrinsic camera parameters are provided. Regarding the face that the proposed calibration method needs an initial guess of the internal characteristics of the camera, its accuracy can be improved as follows: The obtained intrinsic parameters are put back to the second step of the method and a new set of the camera parameters is estimated. If the new estimate of the intrinsic camera parameters differ from the previous one, the whole computation is iterated. The iterations finish when some convergence check is fulfilled.

2.7 Step 1 – Initial Estimate of the Intrinsic Camera Parameters

With respect to the coplanar calibration target an initial guess of the intrinsic camera parameters has to be is provided for the calibration. The proposed method requires a priori knowledge on the following parameters:

- Focal length f
- Coordinates of the principal point u_0, v_0
- Linear distortion coefficients b_1 and b_2

The appropriate values can be obtained from a data sheet of the camera. If the parameters are not available, rough estimates (such as nominal focal length, center of the image and no linear distortions) should suffice. These parameters are iteratively refined during the calibration process.

2.8 Step 2 – Estimation of the Extrinsic Camera Parameters for Each View

In case of a coplanar calibration target, the estimation of the camera parameters is possible only under the knowledge of a subset of its parameters. When the intrinsic camera parameters are known, the appropriate position of the camera can be computed from the model. The intrinsic parameters are constant for all views which ensures the consistency of the estimates. The computation of this step depends on particular model of the camera, the coplanar DLT was chosen here. The parameters are computed directly from the model, then their values are refined in an optimization routine.

2.8.1 Direct Computation of the Extrinsic Parameters

The CDLT matrix A (Eq. 2.7) is the result of the camera calibration with coplanar target based on the CDLT model. This matrix represents the camera and the extrinsic parameters can be extracted from it under the knowledge of the intrinsic characteristic of the camera. The matrix A can be written as:

$$A = \lambda V^{-1} B^{-1} FMT \tag{2.14}$$

where the matrices F (Eq. 2.2), B (Eq. 2.9) and V (Eq.2.8) represent the known intrinsic camera parameters f, u_0, v_0, b_1 and $b_2$respectively. The rotation matrix M (eq.2.3), the translation matrix T (Eq.2.11) and the scaling factor λ are unknown. They have to be extracted from the matrix A in order to compute their values. The matrix A can be premultiplied with matrices F, B and F, the result is the matrix Q:

$$Q = F^{-1} BVA \tag{2.15}$$

From Eq. 2.14 follows that:

$$Q = \lambda MT \tag{2.16}$$

This means that the matrix Q represents the unknown position of the camera and the scaling factor λ.

The extraction of the rotation matrix M is based on the orthogonality constraint on its columns.

Eq. 2.16 can be written in a vector form as:

$$\begin{pmatrix} \vdots & \vdots & \vdots \\ q_1 & q_2 & q_3 \\ \vdots & \vdots & \vdots \end{pmatrix} = \lambda \begin{pmatrix} \vdots & \vdots & \vdots \\ m_1 & m_2 & m_3 \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} 1 & 0 & \vdots \\ 0 & 1 & -t \\ 0 & 0 & \vdots \end{pmatrix} \tag{2.17}$$

Yielding that the first two columns of the matrix M can be estimated directly from the following equations:

$$\lambda m_1 = q_1 \tag{2.18}$$

$$\lambda m_2 = q_2 \tag{2.19}$$

The scale factor λ should be the same in both equations but this is not true in real situations. Regarding that, λ is approximated by the value:

$$\lambda = \frac{\|\mathbf{q}_1\| + \|\mathbf{q}_2\|}{2} , \tag{2.20}$$

Where the operator $\|\cdot\|$ denotes the vector size of a particular vector. After the scaling factor λ is known, the first two columns of the rotation matrix M are approximated by the vectors:

$$\widetilde{\mathbf{m}}_1 = \frac{1}{\lambda}\mathbf{q}_1 \tag{2.21}$$

$$\widetilde{\mathbf{m}}_2 = \frac{1}{\lambda}\mathbf{q}_2 . \tag{2.22}$$

Being the part of the rotation matrix, these two vectors have to be orthonormal. With respect to the fact that only approximate values of the internal camera parameters and the scaling factor λ are provided, this constraint does not have to be fulfilled. Thus, the vectors are normalized first, yielding:

$$\widetilde{\widetilde{\mathbf{m}}}_1 = \frac{1}{\|\widetilde{\mathbf{m}}_1\|}\widetilde{\mathbf{m}}_1 \tag{2.23}$$

$$\widetilde{\widetilde{\mathbf{m}}}_2 = \frac{1}{\|\widetilde{\mathbf{m}}_2\|}\widetilde{\mathbf{m}}_2 . \tag{2.24}$$

Then, the orthognalization is based on the following idea (see Figure 2.14): The vectors determine a plane. To become orthogonal, the vectors have to be rotated in this plane. This operation is accomplished by transforming them into the 2D coordinate system, orthogonalizing (see Figure 2.14) and transforming the orthogonalized vectors back into the 3D space. The transformation to the 2D plane is performed by a rotation matrix G. The matrix G is created from a quaternion g:

$$g = \left[\cos\frac{\gamma}{2}, r_x \sin\frac{\gamma}{2}, r_y \sin\frac{\gamma}{2}, r_z \sin\frac{\gamma}{2}\right]^T = [g_0, g_1, g_2, g_3]^T. \quad (2.25)$$

where γ = arccos $(n_m \odot [0,0,1])$ is the angle of rotation along the axis $r = [r_x, r_y, r_z]$. The vector $n_m = \tilde{\tilde{\mathrm{m}}}_1 \times \tilde{\tilde{\mathrm{m}}}_2$ is the cross vector product of $\tilde{\tilde{\mathrm{m}}}_1$ and $\tilde{\tilde{\mathrm{m}}}_2$, $r = n_m \times [0,0,1]$. The operator ⊙ denotes the scalar product of vectors. The matrix G is :

$$G = \frac{1}{\|g\|}\begin{pmatrix} 1-2(g_2^2+g_3^2) & 2(g_1g_2) & 2(g_1g_3-g_0g_2) \\ 2(g_1g_2) & 1-2(g_1^2+g_3^2) & 2(g_2g_3+g_0g_1) \\ 2(g_1g_3) & 2(g_2-g_3) & 1-2(g_1^2+g_2^2) \end{pmatrix} \quad (2.26)$$

Transforming vectors $\tilde{\tilde{\mathrm{m}}}_1$ and $\tilde{\tilde{\mathrm{m}}}_2$ by the matrix G give the desired coordinates of these vectors in the 2D coordinates system:

$$t_1 = G \ \tilde{\tilde{\mathrm{m}}}_1 \quad (2.27)$$

$$t_2 = G \ \tilde{\tilde{\mathrm{m}}}_2 \quad (2.28)$$

Now the vectors are made orthogonal as demonstrated in Figure 3.2. The angle $\propto$ between the vectors

$$\propto = arccos(t_1 \odot t_2) \quad (2.29)$$

Is changed to $\frac{\pi}{2}$ by rotating both vectors in the opposite directions by the angle $\frac{\beta}{2}$:

$$\frac{\beta}{2} = \frac{1}{2}\left(\frac{\pi}{2} - \alpha\right) \quad (2.30)$$

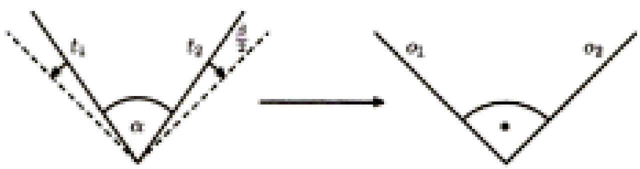


Figure 2.14 Orthogonalization of two vectors in a plane: t_1 and t_2 are the original vectors, o_1 and o_2 the orthogonalized ones

The orthogonal vectors o_1 and o_2 are then transformed back to the original plane by the inverse matrix G^{-1}:

$$m_1 = G^{-1}o_1 \tag{2.31}$$

$$m_1 = G^{-1}o_2 \tag{2.32}$$

With respect to orthogonality property of the matrix M, its third column m_3 is computed as the vector product of m_2 and m_3 :

$$m_3 = m_1 \times m_2. \tag{2.33}$$

Note that the final matrix M,

$$M = \begin{pmatrix} \vdots & \vdots & \vdots \\ m_1 & m_2 & m_3 \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{2.34}$$

Is a proper matrix of rotation.

After the rotation matrix M is estimated, the translation matrix T is computed directly from Eq. 2.17:

$$q_3 = -\lambda Mt \tag{2.35}$$

Yielding the vector t as:

$$q_3 = -\frac{1}{\lambda}M^{-1}q_3 \tag{2.36}$$

The translation matrix T is then constructed as:

$$T = \begin{pmatrix} 1 & 0 & \vdots \\ 0 & 1 & -t \\ 0 & 0 & \vdots \end{pmatrix} \tag{2.37}$$

The results of the decomposition of the CDLT matrix A under the knowledge of the intrinsic camera parameters are the appropriate rotation matrix M and the translation matrix T, which represent the extrinsic camera parameters. In fact, these matrices encode the relative position of the camera in the particular view.

2.8.2 Finding the Optimal Extrinsic Parameters

The direct computation of the extrinsic camera parameters gives only rough estimate of rotation and translation, mainly because only approximations of the intrinsic parameters were used in Eq. 2.15 The optimal values of the extrinsic parameters have to be found so that the difference between the coordinates of the control points detected in the image and the coordinates predicted by the model is minimal. The approximated rotation and translation can be used as the initial guess for the standard least square minimization of the error function:

$$\sum_{j=1}^{n} \left\| g_i - \tilde{g}_j \right\|^2 \tag{2.38}$$

where n is number of control points in the particular view, $g_j = (u_j, v_j)$ are the detected coordinates of the points and $\tilde{q}_j$ are their coordinates predicted by the model:

$$\tilde{q}_j = Ap_j \tag{2.39}$$

Where the matrix A is composed as (See Eq. 2.7):

$$A = \lambda V^{-1} B^{-1} FMT \tag{2.40}$$

The matrices **F, B** and **V** represent the known intrinsic parameters, the rotation matrix **M** is defined by three parameters $\sigma, p, \emptyset$ and the translation matrix **T** is determined by parameters x_0, y_0, z_0 (Eq. 2.11).

The easiest way is to minimize the value of the error function Eq. 2.38 over the full set of unknowns (three for rotation $-\sigma, p, \emptyset -$ and three for translation $-x_0, y_0, z_0$) in R^6:

$$\min_{\sigma, p, \emptyset, x_0, y_0, z_0} \sum_{i=1}^{n} \left\| q_i - \tilde{q}_i \right\|^2 \tag{2.41}$$

But such a large number of unknowns may cause the optimization routine to fail. It would be better, if the dimension of the searching space was reduced.

2.8.3 Reduction of the Dimension of the Searching Space

The number of unknowns of a minimized function (the dimension of the searching space) can influence the speed and stability of the optimization routine. Therefore it is better to find a smallest possible number of parameters of the minimized function.

When reducing the dimension of searching space, some of the unknowns have to be made dependent of the rest of them. The question is whether to reduce the number of parameters expressing the rotation or the translation. There are constraints on the matrix of rotation, which are used in the decomposition of the CDLT matrix. On the other hand, the translation matrix is computed directly at the end of decomposition, thus it is a better candidate for the reduction of the number of parameters.

The basic idea is that, under the knowledge of the coordinates pairs of control points and their images, the x_0 and y_0 parameters of the translation can be expressed by the appropriate z_0 parameter. From the equation of perspective projection:

$$(u_j, v_j) = \left(\frac{x_j}{1-\frac{z_j}{f}}, \frac{y_j}{1-\frac{z_j}{f}}\right) \tag{2.42}$$

where (u_j, v_j) is a point in an image, $(x_j, y_j z_j)$ is its source point in a scene and f represents the focal length, it can be noticed the coordinates of image points are function of z coordinate of their sources. When (u_j, v_j) and $(x_j, y_j z_j)$ are known, the equation can be reformulated. Then, for a given rotation only one unknown, z_0, to determine the translation is needed, thus reducing the dimension of the searching space to R^4 (compare with Eq. 2.41):

$$\min_{z_0,\sigma,\rho,\phi} \sum_{j=1}^{n} \|q_i - \tilde{q}_i\|^2 \tag{2.43}$$

The unknown translation has to be expressed by the matrix **A** and the coordinate pairs of control points.

Denoting $S = T^{-1}A$ and with respect to Eq. 2.40, the matrix S can be computed as:

$$S = V^{-1}B^{-1}FM \tag{2.44}$$

The matrix **S** represents all transformations covered by the CDLT model, except translation. Translating the coordinates of control points p_j with the unknown translation T and then transforming the resulting points with matrix **S** is equal to transforming the control points with the matrix **A:**

$$q_j = Ap_j = TT^{-1}Ap_j = TSp_j \tag{2.45}$$

The result are the image coordinates of control points. Let $q' = [u_j', v_j']$, where:

$$\begin{pmatrix} w_j u_j' \\ w_j v_j' \\ w_j \end{pmatrix} = s^{-1} \begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix} \tag{2.46}$$

Be the coordinates of the control points translated by the unknown translation and $q = [u_j, v_j]$ are coordinates of images of control points. Then the unknown translation can be expressed by z_0 reformulating Eq. 2.42 and computing the average value to minimize the error as:

$$x_0 = \frac{1}{n}\sum_{j=1}^{n} v_j' z_0 - x_j \tag{2.47}$$

$$y_0 = \frac{1}{n}\sum_{j=1}^{n} v_j' z_0 - x_j \tag{2.48}$$

where n is the number of control points and $p = [x, y_j, 0]$ is the *j*-th scene point. As a result of this step, the optimal translation and rotation respective to the specified values of the intrinsic parameters are estimated.

2.9 Step 3 – Construction of a Virtual 3D Calibration Object

As it was already noted, a 3D calibration object is simulated by using multiple of the same coplanar target. All views of the target are acquired by one camera in different positions, thus the internal camera parameters are assumed to be constant. One of the views is chosen as a reference view, let it be the first one.

The virtual 3D object consists of *N* planes, where *N* is the number of views. The planes are transformed so that the image coordinates of the control points observed by the camera in the reference position are the same as the image coordinates of the same points acquired by the camera in the position as R_i.

Because the positions of the camera are estimated in the previous step for all views, the transformation of the planes can be derived from them. A relationship between two positions of the camera C_{1i} has to be found to express the transformation of the 2D plane R_i. The construction of the virtual 3D object can then be formally specified as:

$$q_i = A_1 R_i p_i, i = 1 \ldots N \tag{2.49}$$

where p_i are the scene coordinates of control points respective to the *i*-th vies, q_i are the appropriate images, A_1 is the matrix representing the camera in the reference position. Note that C_{11} and R_1 are the identity matrices.

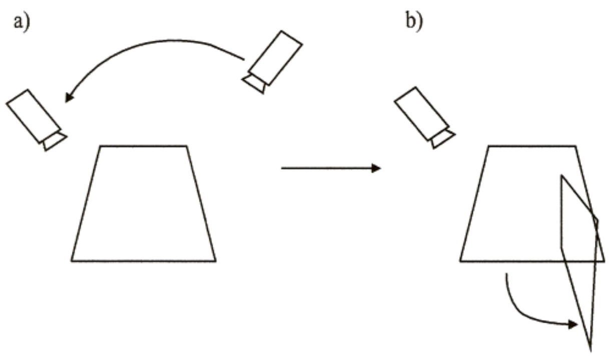


Figure 2.15 Using more views of 2D plane (a) to simulate a 3D object (b)

The relation between two positions of the camera can be expressed as**:**

$$C_{ij} = T_{ij} M_{ij} \tag{2.50}$$

where T_{ij} stands for translation and M_{ij} for rotation. The relation is described as follows: First, the camera has to be moved back to the center of coordinates of the scene. This is done by inverse translation. Then it has to be rotated back to the initial orientation. This is done by inverse rotation. Then the camera is moved to the the second position by rotating and translating:

$$C_{ij} = T_{ij} M_j M_i^{-1} T_i^{-1} \tag{2.51}$$

where T_i and M_i are the translation and the rotation parts of the first position of the camera and T_j and M_j are the translation and the rotation parts of the second position of the camera. Homogeneous coordinate have to be used here, yielding the translation and rotation matrices in the 4 x 4 form instead of 3 x 3.

When constructing the virtual 3D object it is the 2D plane which is moved, not the camera (see Figure 3.3). The unknown position of the 2D plane in the scene coordinate system is acquired by reformulating the above expression of the relationship between the position of the camera. Utilizing homogeneous coordinates, the transformation of the 2D plane is expressed as:

$$R_i = T_i M_i M_1^{-1} T_i^{-1} \tag{2.52}$$

where T_1 and M_1 are the translation and the rotation parts of the first position of the camera and T_1 and M_1 are the translation and the rotation parts of the position of the camera respective to the *i*-th view. It can be noticed, that $R_i = C_{i1}^{-1}$.

Now the image coordinates off the control points acquired by the camera in the position respective to the *i-th* view are considered as observed by the camera in the reference position. Their appropriate scene coordinates can be found by transforming the original scene coordinates with the matrix R_i :

$$P_i' = R_i . P_i \tag{2.53}$$

where P_i are the original scene coordinates and P_i' are the transformed scene coordinates.

After the matrices R_i are found, the virtual 3D object V is constructed. The object V consists of control points. Let P_i be all the coordinates of control points respective to the *i*-th view. Then the virtual 3D object V can be formally specified as:

$$V = \bigcup_{i=1}^{N} \mathrm{R}_i P_i \quad . \tag{2.54}$$

The appropriate images Q of the control points forming the virtual 3D object V are given as:

$$Q = \bigcup_{i-1}^{N} Q_i \tag{2.55}$$

where Q_i are the coordinates of images of control points respective to the i-th view. Now there is a virtual 3D object which can be used for a complete 3D based camera calibration.

2.10 Step 4 – Calibration with the Virtual 3D Calibration Object

The virtual 3D calibration object created in the previous step is represented by control points V. The appropriate images of these points Q are also part of the output of the previous step. These coordinate pairs can be passed to any explicit camera calibration method based on a 3D calibration object. It this case, the DLT was chosen.

The DLT matrix **A** can be estimated using the method proposed in [17]. Then a complete set of camera parameters is obtained by decomposition of the DLT matrix as proposed in [17]. Thus the parameters representing the rotation, the translation, the focal length, principal points and linear distortion coefficients are obtained.

2.11 Iterative Refinement of the Camera Parameters

Provided by the complete set of the camera parameters from the previous step and with respect to the need of an initial guess, iterations are used to refine the values of the camera parameters. The intrinsic parameters obtained from the previous step are passed to the second step of the method.

At the end of each iteration, the calibration error expressing the difference between the coordinates of control points predicted by the camera model and their positions detected in the image is computer as:

$$\frac{1}{m}\sum_{i=1}^{m} \|q_i - \hat{q}_i\| \tag{2.56}$$

where q_i are the detected points, $\tilde{q}_i$ are the points predicted by the model (see Eq. 2.39) and *m* is the total number of control points in all views. If the condition of convergence is satisfied, the method is finished, else the next iteration is started. A typical conditions are for example the improvement of the camera parameters, the value of the calibration error or the maximum number of iterations.

2.12 Practical Realization

The proposed method is independent on camera calibration models used in the particular steps, any explicit model can be sued, In our case, the DLT model and its coplanar variant were chosen. They were selected because of their simplicity and ease of implementation.

As an input, the method requires the coordinate pairs of control points consisting of the position in the scene and in the image plane. The initial guess of the focal length, principal point and linear distortion coefficients has to be supplied by the user. Such values can be acquired from the data sheet of a camera. Note that the method requires the focal length to be specified in pixels. When the conversion between the millimeters and pixels cannot be determined, a good initial guess in to multiply the focal length in millimeters by 100. The center of the CCD array is a good approximation of the unknown principal point and 0 is usually set as a value of the linear distortion coefficients. The conditions under which the iterative process stops have also be specified by the user. This includes the maximum number of iterations and desired precision level.

The proposed method is implemented in a System for Numerical Computation and Visualization[22] . Some of the subroutines were kindly provided by Radim Halif (estimation of DLT and CDLT matrices linear algebra and more), other were written by the author (main method, construction of virtual 3D object, decomposition of CDLT matrix, utility functions).

MATLAB was chosen because it is widely used software in scientific community and because off the efficiency of coding. The MATLAB has a high level language with many useful build-in functions. For example the method of R.Tsai[26] implemented in C has 5886 lines of code plus the optimization routine consisting of

3753 lines of code. The proposed method has 1624 lines of code including the utility functions.

The minimization of the error function defined in Eq. 2.38 requires some optimization routine to be used. In our case, SolvOpt[13], a freeware nonlinear optimization toolbox for the MATLAB, was chosen. The commercial optimization toolbox was not available for us, although the routines included in it can provide better and faster convergence.

2.13 Markers

The infrared optical tracking system can use either active or passive tracked objects. Active optical tracking is the use of light-emitting diodes such as infrared-emitting diodes (IREDs) as markers or tracked objects. (See 2.16(a) Passive optical tracking can be retroreflective materials in forms of bead, paint, or tape attached to markers. (See 2.16(b) The wavelength used for active or retroreflective marker is normally 880 nm. There are some differences for using passive or active markers. Choosing markers depends on applications and constraints of each system. Passive marker presents several advantages for example, it is simple to use, it is convenient to locate on a patient, and it requires no power. On the other hand, passive marker may have the precision problem due to the partial occlusion and small distortion from infrared reflection, which lead to miscalculation of marker centroid. Alternatively, an active marker works even though there is an obstruction at the line of sight between marker and cameras and it can be controlled on-off state to help in tracking algorithm. However, the active marker still requires power cable.

(1) (2)

Figure 2.16 (a) Active markers, (b) Passive markers

Moreover, the tracked markers have some features needed to be considered such as the field of emission, size and shape of the marker, the inter-marker distance constraints and the occlusion problem.

Therefore, wireless IR-LEDs active markers are presented to solve the problem of the inconveniences and offered alternating control if there are more than one set of markers. To design the wireless active markers, there are some conditions needed to be considered as follows[45] :

1. Markers should be small enough to attach to any medical tools.
2. Wireless communication must have some error detection.
3. Markers should be compatible to any personal computer and Microsoft Windows.
4. Markers can communicate at least 10 meters and consume low power.

2.13.1 Wireless active marker system

From the trends mentioned above, the power from battery is used to make the system small and low power consumption. We chose the small size PIC12F509 Microcontroller from Microchip Technology Inc.[46] to control and communicate the module via wireless serial port (TLP-434 and RLP-434 from Laipac Technology Inc.)[47] at 434 MHz frequency. The wireless marker system consists of two parts: master transmitter and slave receiver or active marker.

2.13.2 Transmitter

PIC12F509 Microcontroller communicates to personal computer via serial port (RS232) to control the On/Off of the markers. Furthermore, the microcontroller arranges the sending data for transmission through module TLP-434 by Amplitude Shifted Keying (ASK) Modulation. The additional data is added to the preamble or header, which are the 20 bits of synchronization between the transmitter and receiver, 1 byte of active marker address, 2 bytes of the On/Off data of IR-LEDs, and 1 byte of CRC checksum for error detection of all data. All data is encoded by Manchester Encoding. (See 2.17)

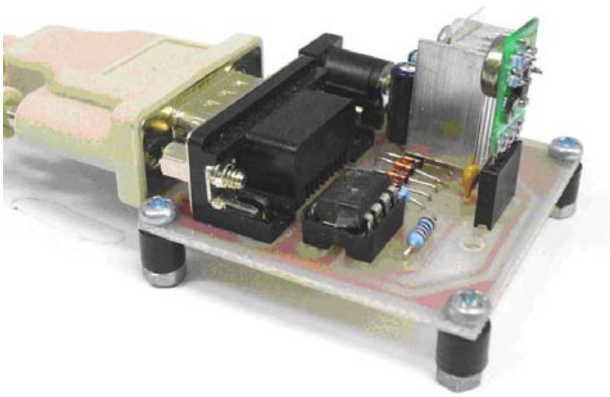


Figure 2.17 Transmitter circuit

2.13.3 Receiver

Data packet transmitted in the air is received by module RLP-434. This module transforms the ASK modulated data signal to digital signal for the microcontroller PIC12F509. The microcontroller will adjust itself to synchronize to the transmitted data by checking 20-bit preamble. When they are synchronized, the microcontroller will save the next 4 bytes to the internal memory for error checking. The first checking is to test whether the data in the first byte is the same as the identification number of the receiver or not. If they are different, all of this data packet will be ignored and the receiver will be reset. If they are alike, the microcontroller will check the correctness of data by CRC checksum in the last byte. If the data is wrong, the receiver will be reset. If the data is correct, the microcontroller will serially shift the data to the IC 74LS595, which translates the data to the On/Off state of the IR-LEDs that connected to this IC. (See Figure 2.18)

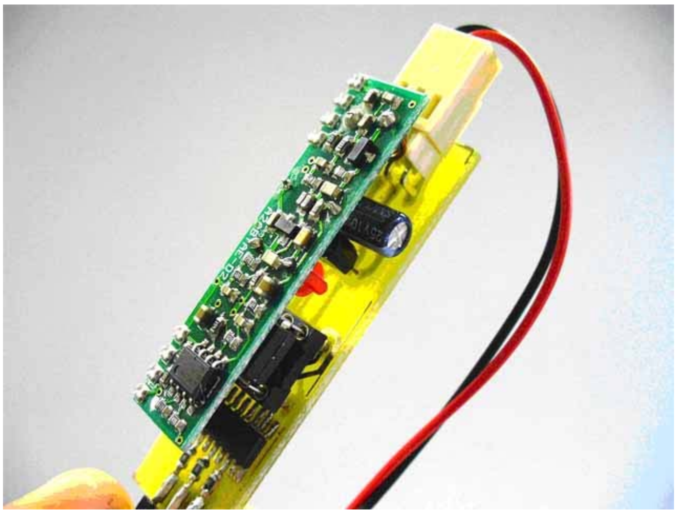


Figure 2.18 Receiver circuit

2.14 Image Processing and Implementations

When the two cameras are connected to the computer, there must be some interfaces for manipulating the camera settings, acquiring images and image processing to determine the coordinates of markers in three-dimension. First, we use MATLAB® to test the feasibility of the system. After that, we choose Microsoft DirectShow and OpenCV library to implement the system for real-time application.

2.14.1 Image Processing Algorithms

2.14.1.1 Thresholding

Thresholding is a method to distinguish the interesting objects in an image from background by creating a binary partitioning of the image intensities. The procedure is accomplished by comparing the desired intensity value called threshold to intensities of all pixels. The pixels that have intensity value greater than the threshold are assigned to one class. Otherwise, they are assigned to another class.

The threshold value can be set manually or automatically to a percentage of the maximum intensity within the image. The simple thresholding can partition the image into only two classes. However, some drawbacks could occur. For example, noise and intensity variation may be sensitive to the threshold segmentation, which can affect to the connectivity of desired component.

2.14.1.2 Connected Components Labeling

Connected components labeling scans an image and groups its pixels into components based on pixel connectivity[50]. All pixels in a connected component have the same intensity values and connect to each other through its neighbors. Each pixel in the same group is labeled with a gray level or a color (color labeling) that is assigned as shown in 0.

Connected components labeling can be used in both binary and grayscale image with different measures of connectivity such as 4-, or 8-connectivity. The algorithm scans through the image pixel-by-pixel to find the connected regions. For a binary image, it searches for connected pixels that have intensity value V = {1}. For a grayscale image, the same component may have the intensity in a range of values such as V= {51, 52, 53, ... , 77, 78, 79, 80}. For the following, the image is assumed as the binary image to ease the explanation. Connected component labeling scans across each row until finding the pixel P that has intensity value V={1}and examines the neighbors as follows:

- If all neighbors have intensity values V= {0}, then create a new label to P.
- If there is only one neighbor that has V= {1}, then assign the label of that neighbor to P.
- If there are more than one neighbor that have V= {1} and have different labels, then assign one of the labels to P and make a note of equivalences.

When the image is all scanned, equivalent label pairs are sorted into equivalence classes and the labels are reassigned to unique labels. Then scan the image again and replace the labels with the distinctive labels from equivalence classes.

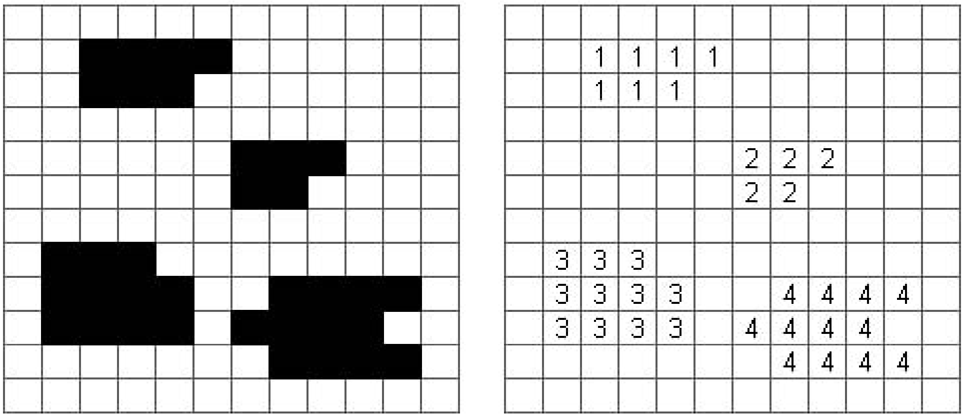


Figure 2.19 An example of connected component labeling

2.14.1.3 Finding centroids of markers

After the labeling is performed to locate the area of markers, centroids of markers are determined to represent the position of the marker. The calculation of centroids has the same formula as the computation of center of mass, which are

$$\bar{x} = \frac{\sum x\ m}{\sum m} \tag{2.57}$$

$$\bar{y} = \frac{\sum y\ m}{\sum m} \tag{2.58}$$

where $\bar{x}$ is a centroid position of a marker in x axis.

$\bar{y}$ is a centroid position of a marker in y axis.

x is a pixel position of marker area in x axis.

y is a pixel position of marker area in y axis.

m is the intensity value of marker area in each pixel.

2.15 Image Acquisition Toolbox and Image Processing Toolbox in MATLAB®

For the implementation of this project, we initially used MATLAB® to test the algorithms. MATLAB® is a high-performance language for technical computing, which contains many toolboxes for computation, visualization, and programming. The major toolboxes, which are helpful to the project, are Image Acquisition Toolbox and Image processing Toolbox.

Image Acquisition Toolbox supports a variety of image acquisition operations as follows[51] :

- Acquiring images from image acquisition devices such as frame grabbers, USB-based Webcams, and FireWire Cameras
- Viewing a preview of the live video stream
- Triggering acquisitions
- Configuring callback functions that execute when certain events occur
- Bringing the image data into the MATLAB® workspace

We employed this toolbox to capture frames from two cameras. The basic steps to accomplish the operation must be performed for the following:

- Install and configure the image acquisition device.
- Retrieve hardware information that uniquely identifies the image acquisition device to the Image Acquisition Toolbox. To get the information, '*imaqhwinfo*' function is used, which will tell us about the adaptor name, device ID, and supported video format.
- Create a video input object to establish the connection between MATLAB® and an image acquisition device. The '*videoinput*' function is used with the retrieved information about adaptor name, device ID, and video format. The video input object can be used to control the image acquisition process.
- Preview the video stream (Optional). After creating the video input object, one may want to see the preview output of the video stream for further adjustments i.e., the camera position, lights, focus, or other

image acquisition setup. The ‘*preview*’ function opens a window and displays the live video stream.

- Configure image acquisition object properties (Optional). Some characteristics may be changed by using ‘*get*’ and ‘*set*’ function. Using properties of the object, some acquisition process can be controlled, such as the amount of video data to capture.
- Acquire image data. There are some steps to process the capturing. First, start the video input object to prepare data by using ‘*start*’ function. Next, trigger the acquisition. Triggers depend on the TriggerType property that is previously configured. If the property is set to ‘immediate’, the object executes the trigger and acquires image immediately after the ‘*start*’ function is called. After that, to bring data into the MATLAB® workspace, the ‘getdata’ function is called. The data is brought in multiple frames and then it can be manipulated as you want.
- Clean up. Once the image capturing is finished, the image acquisition device should be removed from the memory and MATLAB® variables should be cleared using ‘delete’ and ‘clear’ functions.

When images are brought to the MATLAB® workspace, next step is the image processing using Image Processing Toolbox. This toolbox provides a wide range of image processing operations in which we attend in morphological operations. Morphology is a technique of image processing based on shapes. The value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. The toolbox has the function ‘*bwlabel*’, which can be used to determine the areas of markers existed in the frame.

CHAPTER III
METHODOLOGY

This experiment is about tracking multiple markers in 3D space. By simulate experiment instead real work. Beginning from generate image of marker .The image from right and left camera as generate with 3Ds MAX program that provide powerful, integrated 3D modeling, animation, rendering, and compositing that enable artists and designers to more quickly ramp up for production. And then import the image into process for track position of marker that matches position from right and left camera. The functional process follow is

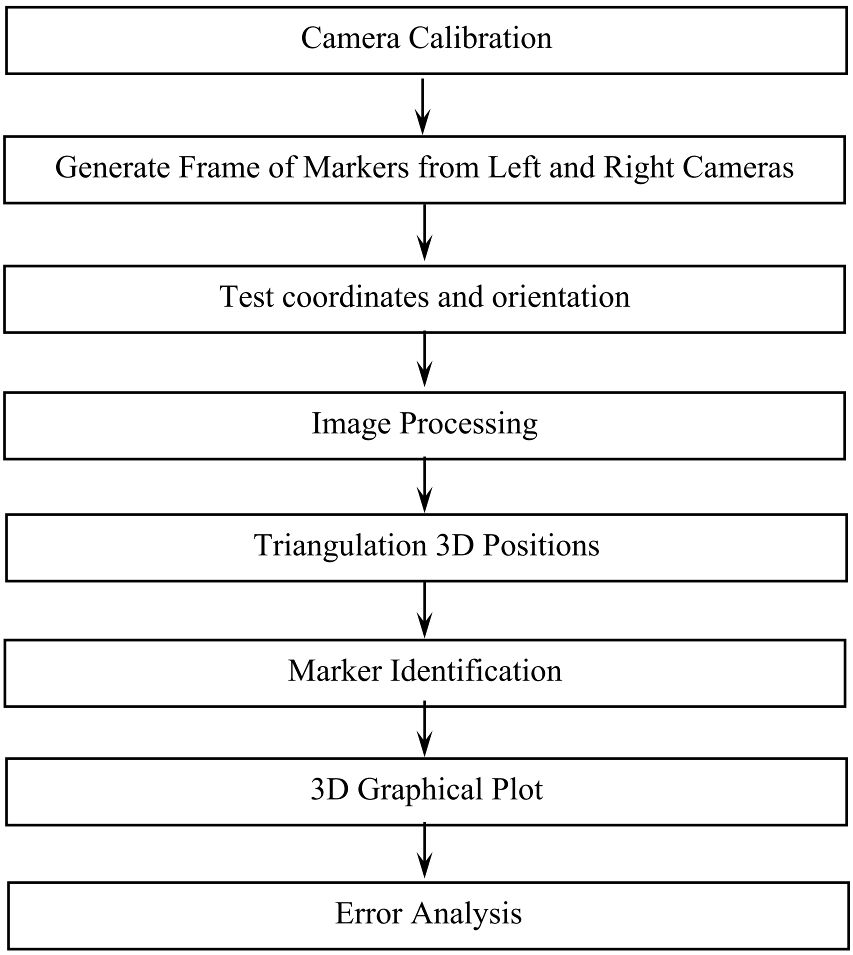


Figure 3.1 Diagrammatic representation of research

3.1 Camera Calibration

The camera calibration is performed using the Camera Calibration Toolbox for MATLAB® developed by Jean-Yves Bouguet, MRL - Intel Corp[58]. It is mainly based on Zhang[59], Heikkila and SilvCn [38], and Tsai camera calibration method[30] , which is used to fit the camera model parameters to the captured images. The calibration process requires a calibration target. A checkerboard pattern, with the size of 20cm x 20cm containing 10 x 10 squares, is employed for this purpose. The target is placed in front of the cameras in various positions/orientations, and the images are captured (02) The corners of the checkerboard pattern are determined. The positions of corners are known from relative coordinate system of the target. The process of corner detections uses edge detection, straight line fitting to detect linked edges and intersecting the lines to obtain the image corners. After that image corners are matched to 3-D target checkerboard corners by counting if number of squares is all visible in the image. The number of squares is predefined for comparing. As a result, the coordinate pairs of images frame and world frame are found which can be employed for calculation to find the intrinsic parameters and also extrinsic parameters[60]. The cameras are calibrated once before the tracking process.

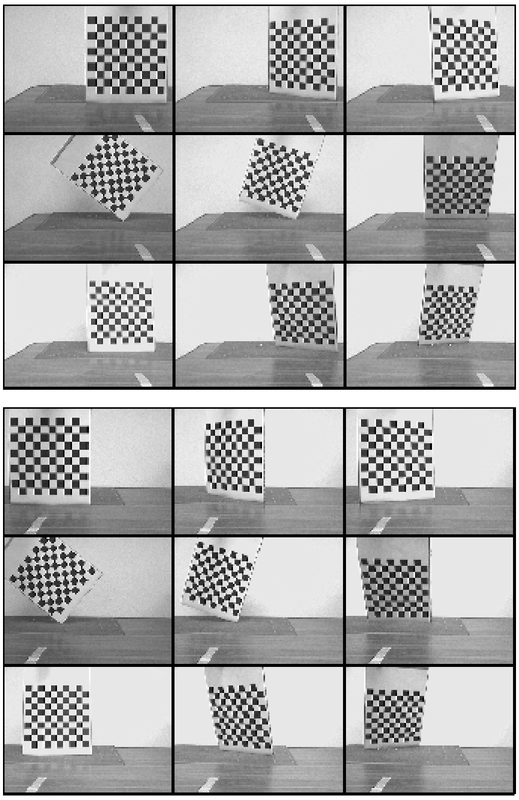


Figure 3.2 Examples of the target images for calibration from left and right camera.

3.2 Generate Frame of Markers from Left and Right Cameras

Images will generate with 3Ds MAX all about 60 images that divide images from right camera 30 images and left camera 30 images which are images of same scene. The images are jpg format and size 640 x 480

The each image compose with sphere marker 3 mm 12 markers, length of arm of markers don't equal and divide 3 groups are group1, group2 and group3.

Group1: compose markers	1-4	represent A
Group2: compose markers	5-8	represent B
Group3: compose markers	9-12	represent C.

The process in this experiment has 5 cases that consist of

Case1 : A, B and C are separated represent $A \rightarrow B \rightarrow C$.

Case2 : A is near B but away from C represent $A \& B \rightarrow C$.

Case3 : B is near C but away from A represent $B \& C \to A$.

Case4 : A is near C but away from B represent $A \& C \to B$.

Case5 : A, B and C are near represent $A \& B \& C$.

All of the cases of experiment have 6 patterns that each case comes from Gaussian distribution. Gaussian distribution is an absolutely continuous probability distribution with zero cumulants of all orders higher than two. The graph of the associated probability density function is bell-shaped, with a peak at the mean, and is known as the Gaussian function or bell curve.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where μ and σ^2 are the mean and the variance of the distribution. The Gaussian distribution with

$\mu = 0$ and $\sigma^2 = 1$ is called the standard normal distribution. The graph of Gaussian distribution is showed in figure 3.3

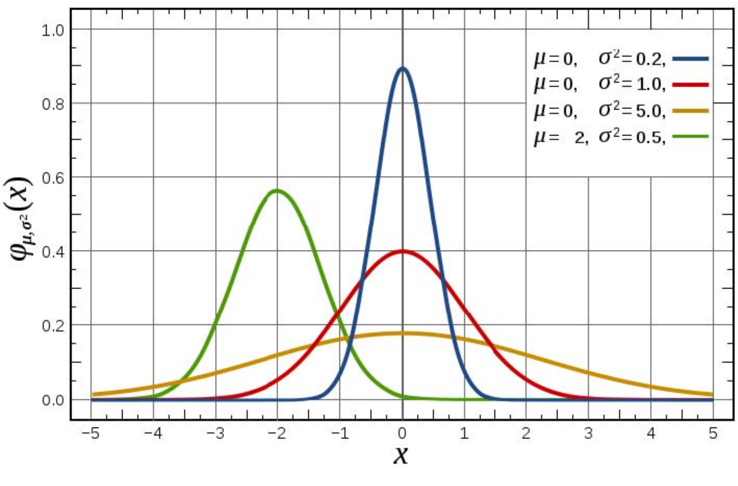


Figure 3.3 Graph of Gaussian distribution.

The 6 patterns are pattern 1, pattern 2, pattern 3, pattern 4, pattern 5 and pattern 6. Values of 6 patterns are

Pattern 1: Use value of coordinate of markers from image.

But value of pattern 2-6 are calculated by equation that is

Number of pattern = value of before pattern + S.D.* randn(m,n) ,

where S.D. is 0.1,0.2,0.25,0.35 and 0.4 so, the equation of pattern 2-6 are

Pattern 2 = value of pattern 1 + 0.1* randn (m,n)

Pattern 3 = value of pattern 2 + 0.2* randn (m,n)

Pattern 4 = value of pattern 3 + 0.25* randn (m,n)

Pattern 5 = value of pattern 4 + 0.35* randn (m,n)

Pattern 6 = value of pattern 5 + 0.4* randn (m,n)

3.3 Test coordinates and orientation

This procedure brings 24 coordinates that come from coordinates of makers in left and right camera. Bring every marker to test for track some marker that missing. As some marker missing may be happen from shade of some marker. They will see in each other same scene because if some marker is missing, the experiment has wrong result. Besides this procedure test shade of markers, it tests noise of images too.

3.4 Image Processing

In this procedure, import images are processed to obtain 2D position of centroids of markers. The steps are following

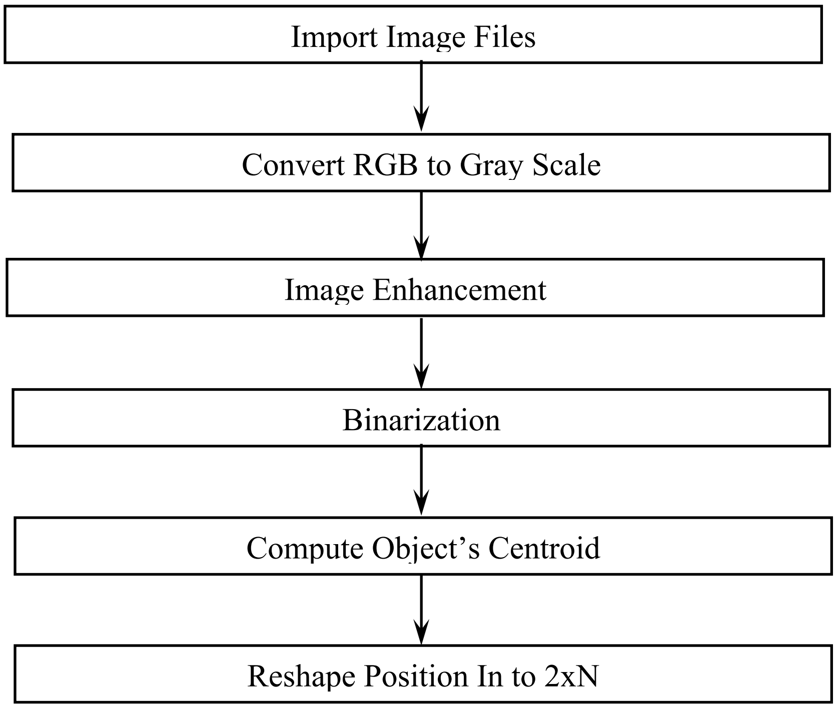


Figure 3.3 Diagrammatic representation of image processing

3.4.1 Import Image Files

This procedure is imported the image to preprocessing. As 60 images divide to 5 cases follow in 3.1. By import each one image until 60 images. Value of images will keep for next step.

3.4.2 Convert RGB to Gray Scale

This procedure is converted image from RGB to Gray Scale because in actual work, marker must to use LED that has white color. It make RGB image hasn't necessary. Without background, RGB is image in 3D and complicate. But Gray Scale is image in 2D which has a few complicate and compile result better.

3.4.3 Image Enhancement

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques.

Image enhancement techniques can be divided into two broad categories:

1. Spatial domain methods, which operate directly on pixels, and
2. Frequency domain methods, which operate on the Fourier transform of an image.

Unfortunately, there is no general theory for determining what good image enhancement is when it comes to human perception. If it looks good, it is good! However, when image enhancement techniques are used as pre-processing tools for other image processing techniques, then quantitative measures can determine which techniques are most appropriate.

3.4.4 Binarization

This procedure is making the image to binary in Threshold process. Threshold is a method to distinguish the interesting objects in an image from background by creating a binary partitioning of the image intensities. The procedure is accomplished by comparing the desired intensity value called threshold to intensities of all pixels. The pixels that have intensity value greater than the threshold are assigned to one class. Otherwise, they are assigned to another class. That follow to set value Threshold = 220 .Position which has value lower 220 is set to black color (value=0), higher 220 is set to white color (value=1). Value position to use is 1 or equal to 1.

3.4.5 Compute Object’s Centroid.

This procedure is tracking the position of marker’s centroid in each one. In which to find the position of marker, use to calculate from the white areas of the black and white images are labeled the connect components to calculate marker centroids.

According to our markers that composed of three spheres, the centroids of markers that we are determined are a set of three pixel coordinates.

3.4.6 Reshape Position in to 2xN

This procedure is reshape from Nx2 to 2xN for use in triangulation 3D Position because Triangulation 3D Position uses row vector. N is number of marker. This procedure has N=12.

3.5 Triangulation 3D Positions

After we obtain the information from calibrated stereo cameras, the next step is to compute the 3-D position of a point in space from the known disparity map and the geometry of the stereo cameras. This process called Triangulation is to find the intersection of two projection lines in two images of a point in space. The simple way is to use the mid-point method[41]. The mid-point of the common perpendicular to the two lines is chosen by dividing the common perpendicular in proportion to the distance from the two camera centers. The illustration of this method can be modeled as figure 3.4

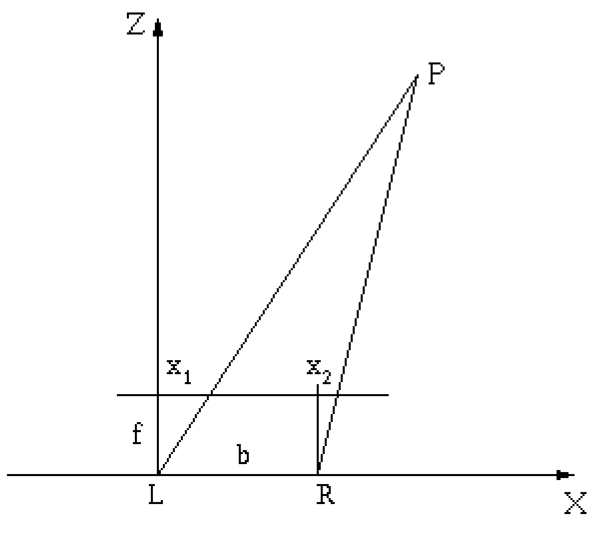


Figure 3.4 Standard models of stereo cameras

Let left camera reference system be the 3-D world reference frame. The right camera is translated and rotated with respect to the left camera frame. In this case the two optical axes are parallel, which the translation of the right camera is only on the X axis. The optical axes lie on the XZ plane. Also, let f be the focal length of both camera and *b* is the distance between the two lens centers. The stereo triangulation can be expressed as:

$$Z = \frac{(b \times f)}{(x_1 - x_2)}$$

$$X = x_1 \times \frac{Z}{f}$$

$$Y = y_1 \times \frac{Z}{f}$$

The centroids of markers from left and right cameras are brought to the function '*stereo_triangulation'* in Camera Calibration Toolbox as shown below, which also takes both extrinsic and intrinsic calibration parameters as inputs for correcting deformity from the imaging hardware.

[pos_L, pos_R] = stereo_triangulation(centroidL, centroidR, om, T,
fc_left, cc_left, kc_left, alpha_c_left, fc_right, cc_right,
kc_right, alpha_c_right);

where pos_L is the calculated 3D positions of the markers relative to left camera

pos_R is the calculated 3D positions of the markers relative to right camera

centroidL is the extracted marker centroids in image of left camera

centroidR is the extracted marker centroids in image of right camera

om is the rotation matrix that identifies the orientation of 2 cameras obtained from the calibration

T is the translational matrix that identifies the position of 2 cameras obtained from the calibration fc_left, cc_left, kc_left, alpha_c_left are the intrinsic parameters of the left camera (output of stereo calibration)

fc_ right, cc_ right, kc_ right, alpha_c_ right are the intrinsic parameters of the right camera (output of stereo calibration).

This function computes the 3D coordinates of the markers relative to the position of the camera. However, with no marker identification, the centroids from left and right cameras can be paired in any combination. This situation can lead to incorrect matching of centroids pairs and incorrect 3D positions. Therefore, marker identification for tracking algorithms is proposed to solve this matter, which has details in the next issue.

3.6 Marker Identification

The proposed method for tracking markers in this system is to use fundamental matrix. In computer vision, the fundamental matrix F is a 3×3 matrix which relates corresponding points in stereo images. In epipolar geometry, with homogeneous image coordinates, x and x′, of corresponding points in a stereo image pair, Fx describes a line (an epipolar line) on which the corresponding point x′ on the other image must lie. That means, for all pairs of corresponding points holds

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0.$$

Being of rank two and determined only up to scale, the fundamental matrix can be estimated given at least seven point correspondences. Its seven parameters represent the only geometric information about cameras that can be obtained through point correspondences alone.

The fundamental matrix can be determined by a set of point correspondences. Additionally, these corresponding image points may be triangulated to world points with the help of camera matrices derived directly from this fundamental matrix. The scene composed of these world points is within a projective transformation of the true scene.

Say that the image point correspondence $xl \leftrightarrow xr$ derives from the world point X under the camera matrices $\left(P, P^{'}\right)$ as

$$xl = PX$$
$$xr = P^{'}X$$

Say we transform space by a general homography matrix H_{4x4} such that $X_0 = HX$.

The cameras then transform as

$$P_0 = PH^{-1}$$

$$P_0^{'} = P^{'}H^{-1}$$

$$P_0 X_0 = PH^{-1}HX = PX$$
$$P^{'}{}_0 X_0 = P^{'}H^{-1}HX = P^{'}X$$

Still get us the same image points

After that we get point of xl and xr to calculate determinant. If the result is equal or nearly, that means it is same position.

3.7 3D Graphical Plot

This procedure is use coordinates x-axis; y-axis and z-axis that take from 3.6 to plot on 3D space for see position of 12 markers.

3.8 Error Analysis

This procedure is finding precision value that from experiment. To find precision value, we will use Root Mean Square Error (RMSE). The root mean square deviation (RMSD) or root mean square error (RMSE) is a frequently-used measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. RMSD is a good measure of precision. These individual differences are also called residuals, and the RMSD serves to aggregate them into a single measure of predictive power. The

RMSD of an estimator $\hat{\theta}$ with respect to the estimated parameter θ is defined as the square root of the mean squared error:

$$\operatorname{RMSD}(\hat{\theta}) = \sqrt{\operatorname{MSE}(\hat{\theta})} = \sqrt{\operatorname{E}((\hat{\theta}-\theta)^2)}.$$

For an unbiased estimator, the RMSE is the square root of the variance, known as the standard error.

In some disciplines, the RMSD is used to compare differences between two things that may vary, neither of which is accepted as the "standard". For example, when measuring the average distance between two oblong objects, expressed as random vectors

$$\theta_1 = \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,n} \end{bmatrix} \quad \text{and} \quad \theta_2 = \begin{bmatrix} x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{2,n} \end{bmatrix}.$$

The formula becomes:

$$\operatorname{RMSD}(\theta_1,\theta_2) = \sqrt{\operatorname{MSE}(\theta_1,\theta_2)} = \sqrt{\operatorname{E}((\theta_1-\theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i}-x_{2,i})^2}{n}}.$$

RMSE from experiment will not less than 0.5 mm. or percent of RMSE not more than 20% that guarantee this algorithm can trust and good precision.

CHAPTER IV
EXPERIMENTAL RESULTS

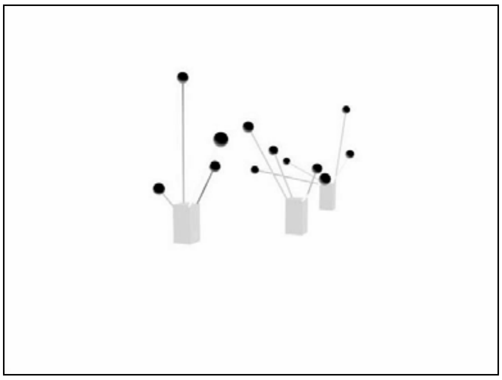
The experiment was the implementation of the system by MATLAB® and shows the result from the error analysis. So, the result of image will show in invert images because the invert images are cleary.

4.1 Generate Frame of Markers from Left and Right Cameras

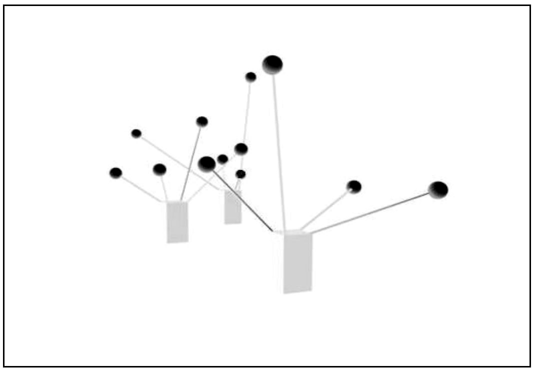
The proposed is composing 24 markers that divide 12 markers of image left and right camera that each image divide markers are 3 groups are group1, group2 and group3. Group1 is compose markers 1-4 represent A, Group2 is compose markers 5-8 represent B and Group3 is compose 9-12 represent C. This experiment divide 5 case is case1 -5.

Case1	:	A, B and C are separated	represent $A \rightarrow B \rightarrow C$.
Case2	:	A is near B but away from C	represent $A \& B \rightarrow C$.
Case3	:	B is near C but away from A	represent $B \& C \rightarrow A$.
Case4	:	A is near C but away from B	represent $A \& C \rightarrow B$.
Case5	:	A, B and C are near	represent $A \& B \& C$.

The result is



(a)



(b)

Figure 4.1 Image of markers

(a) Image from left camera (b) Image from right camera

4.2 Test coordinates and orientation

Result from the experiment has position that happens shade of marker but not shade all of marker. It makes to improve image which has not shade of marker until return image that has all markers and no shade.

4.3 Camera Calibration

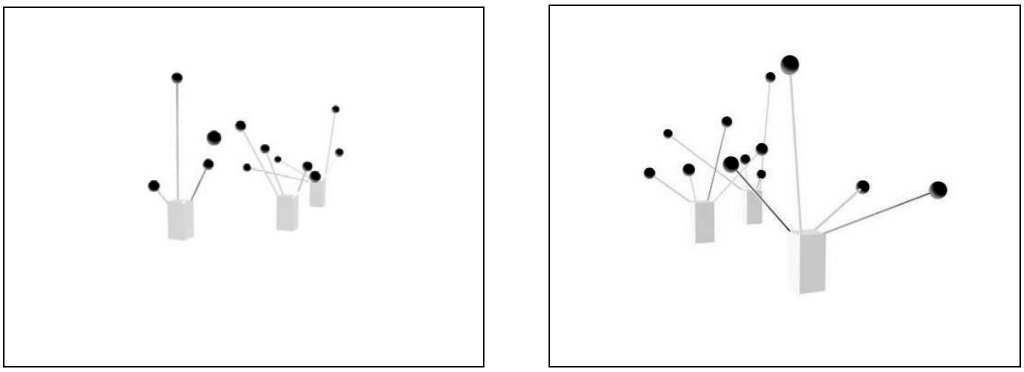
After calibration, the list of parameters may be stored in the matlab data file Calib_Results.mat**.** This section gives a detailed description of all the calibration parameters (Intrinsic and Extrinsic) and their corresponding matlab variable names.

4.4 Image Preprocessing

The procedure is preprocess the image files from left and right camera consists of

4.4.1 Import image files

This method is import 60 images into preprocessing that has jpg format. The images divide 30 pair from left and right camera that is generating images.

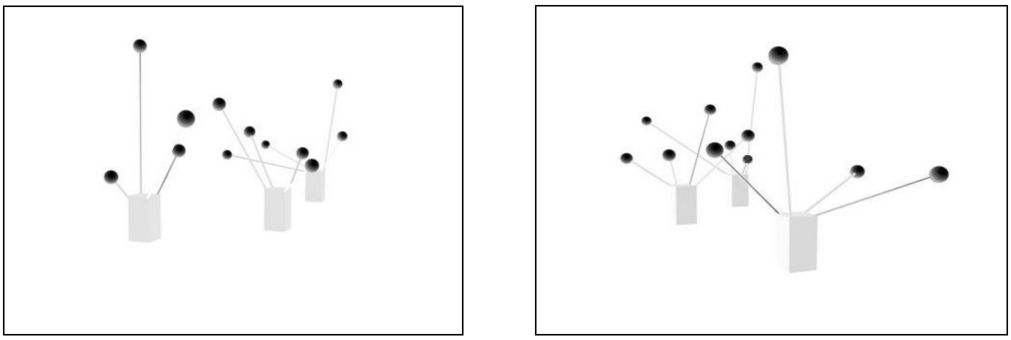


(a) **(b)**

Figure 4.2 Import image of markers
(a) Image from left camera (b) Image from right camera

4.4.2 Convert RGB to Gray Scale

This method is to decrease complicate in process because RGB is image in 3D space and complicate. But Gray Scale is image in 2D space.

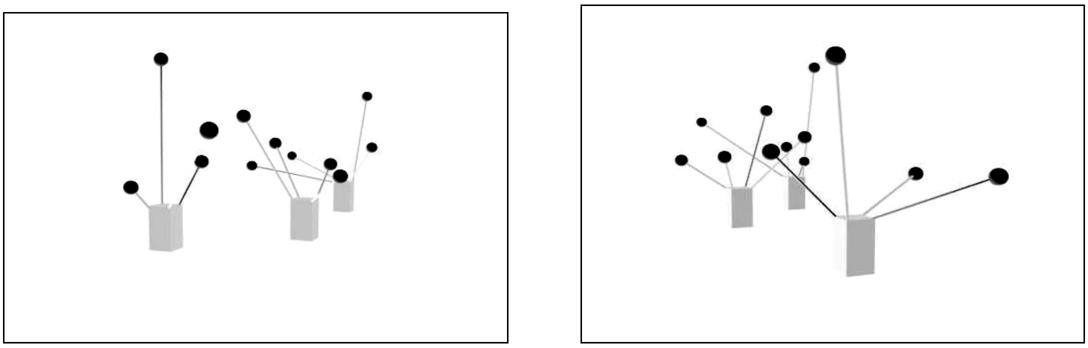


(a) **(b)**

Figure 4.3 Convert Image of markers from RGB to Gray Scale
(a) Image from left camera (b) Image from right camera

4.4.3 Image Enhancement

This procedure is to improve the interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques



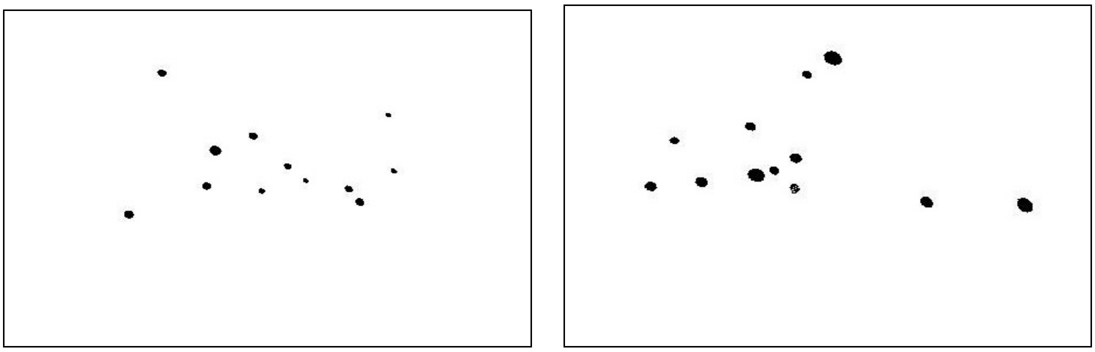
(a) **(b)**

Figure 4.4 Image Enhancement

(a) Image from left camera (b) Image from right camera

4.4.4 BinariZation

This procedure is making the image to binary in threshold process that set value according to 3.3.4. After this procedure, we get the interested position that the sphere of marker 12 makers in each one image , so we get 24 markers of the image of left and right camera.



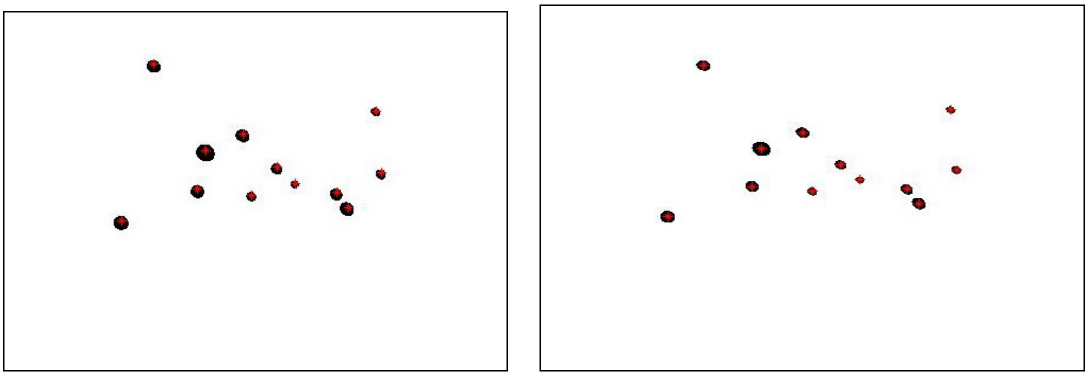
(a) **(b)**

Figure 4.5 Binarization

(a) Image from left camera (b) Image from right camera

4.4.5 Compute Object's Centroid

This procedure is finding the position of marker's centroid in each one by MATLAB program.



(a) (b)

Figure 4.6 Centroids of markers

(a) Image from left camera (b) Image from right camera

4.4.6 Reshape Position in to 2xN

This method is to transpose matrix from Nx2 to 2xN or column vector to row vector for use in Triangulation 3D positions.

4.5 Triangulation 3D Positions

This function computes the 3D coordinates of the markers relative to the position of the camera. However, with no marker identification, the centroids from left and right cameras can be paired in any combination. This situation can lead to incorrect matching of centroids pairs and incorrect 3D positions. Therefore, marker identification for tracking algorithms is proposed to solve this matter, which has details in the next issue.

4.6 Markers Identification

This procedure for tracking markers in this system is to match by fundamental matrix algorithm.

4.7 3D Graphical Plot

This procedure is to bring coordinates after markers identification to plot on 3D space for indicate position of markers as illustrate in figure 4.7

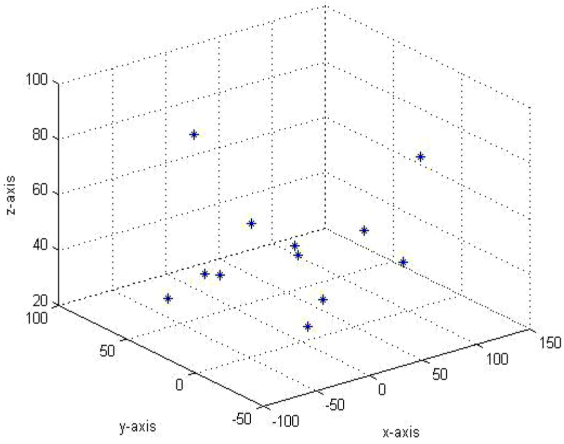


Figure 4.7 Calculated 3D marker positions

4.8 Error Analysis

Markers were attached to the manipulator and moved from the defined reference origin (0, 0, 0). We tested the translational errors for 5 cases by each case have 6 patterns according to 4.1.

The experiment consider in Root Mean Square Error Form (RMS). This algorithm can use for work better.The experiment has 5 cases that result is

4.8.1 Case 1: A, B and C are separated

Table 4.1 Result case 1: A, B and C are separated

Pattern	RMSE		
	x-axis	y-axis	z-axis
1	0.0400	0.0317	0.0464
2	0.0306	0.0475	0.0446
3	0.0506	0.0446	0.0529
4	0.0536	0.0417	0.0577
5	0.0583	0.0517	0.0519
6	0.0498	0.0424	0.0429

4.8.2 Case 2: A is near B but away from C

Table 4.2 Result case 2: A is near B but away from C

Pattern	RMSE		
	x-axis	**y-axis**	**z-axis**
1	0.0945	0.1167	0.0459
2	0.0882	0.0998	0.0372
3	0.0909	0.0757	0.0416
4	0.1020	0.1293	0.0380
5	0.1005	0.0974	0.0461
6	0.1083	0.1063	0.0557

4.8.3 Case 3: B is near C but away from A

Table 4.3 Result case 3: B is near C but away from A

Pattern	RMSE		
	x-axis	**y-axis**	**z-axis**
1	0.1046	0.1384	0.1713
2	0.0963	0.1282	0.1877
3	0.1259	0.1926	0.2441
4	0.1272	0.1790	0.1380
5	0.0807	0.1288	0.1199
6	0.1009	0.1133	0.1879

4.8.4 Case 4: A is near C but away from B

Table 4.4 Result case 4: A is near C but away from B

Pattern	RMSE		
	x-axis	y-axis	z-axis
1	0.1578	0.0957	0.1074
2	0.1973	0.1135	0.1721
3	0.1286	0.0954	0.0890
4	0.1542	0.1130	0.1885
5	0.1765	0.1228	0.1725
6	0.1508	0.1010	0.1378

4.8.5 Case 5: A, B and C are near

Table 4.5 Result case 5: A, B and C are near

Pattern	RMSE		
	x-axis	y-axis	z-axis
1	0.2123	0.2475	0.1872
2	0.2501	0.2456	0.1874
3	0.2412	0.2229	0.2170
4	0.1823	0.2125	0.2136
5	0.1335	0.1702	0.1919
6	0.2343	0.1515	0.1635

Result RMSE of table 4.1-4.5 has less than 0.5 mm. that can accept, so markers identify in 3D space can use fundamental matrix algorithm for identify position of markers because it has precision and trust.

CHAPTER V
DISCUSSION

The implementations of system in this thesis are discussed as follows.

5.1 Summary of the experiment

This experiment is about to find the position of marker in 3D.By the way we will take the image from marker in right and left camera. As the marker image from same scene but position are different, to find position that we see the marker picture from right and left camera. In which position is the same position. This experiment that compute from algorithm for match marker of two image by use the principal of Fundamental Matrix to identify each markers. We have 12 markers but in this experiment we adjust to 5 cases. In each case divide to 6 patterns, total are 30 times. When we get the result already, use the result to find accurate value. By get it from Root Mean Square Error (RMSE). This formula that find error value from the experiment. Error value not more than 0.5 mm.

After the experiment, we will get the value and put in 30 tables. Each one of table we will find RMSE of axis, x-axis, y-axis and z-axis. We can look at the table 4.8.1-4.8.5. RMSE value of each one table from experiment not more than 0.5 mm. and we can find mean of RMSE as follow x-axis=12.41%, y-axis=12.19% and z-axis=12.13%. As follow from condition are accepting. Algorithm in Identify Markers process can trust and accurate for each one of Identify Markers .Total of markers are12 markers.

The result from experiment and RMSE value can accept that support to identify marker method. This method is accurate and trust. That is tracking markers by fundamental Matrix on principle to identify marker.

5.2 Tendency to adjust the experiment

5.2.1 Generate Frame of Markers from Left and Right Cameras

We must to design the marker to layout in the right position (not shading). It easy for detect the marker. Not only that, all arm of marker should be long and different. Color of marker should be use white and different color with background. Include axis and arm of marker, that easy to remove noise. We would be design size of marker to close with the real work. Include limits of work also for the right result.

5.2.2 Image Processing

Result from image processing is get from data that we interested in image. That is each one of coordinate of marker. We can know the position of marker in the image from right and left camera with all of centroid of marker also. Process is very important because it is preprocess of image before we identify marker. So in this process we must try to remove noise from image much more over. It will be easy for identify marker such as erosion process, delation process, etc. When we remove noise much more, algorithm would have efficiency.

5.2.3 Marker Identification

Result of this process that identifies markers by guideline of fundamental Matrix to tracking Markers. Result of RMSE will show at table 4.1-4.5 that told about this guideline is accurate and trust. But in case of number of marker increase, process of algorithm will be late. It still good work. So guideline to adjust may be more pre allocation memory for work of process. It makes process fast.

5.3 Future works

5.3.1 Test algorithm with hardware

This proposes of experiment is tracking markers by simulated stereoscopic cameras that is not test with hardware. So, we should test algorithm with hardware for test process of algorithm because the result of experiment is precision but we don’t know the result of experiment when use algorithm with hardware that precision or not.

5.3.2 Markers

This proposes of experiment is tracking markers in case no marker shade. So, when implement algorithm with hardware should design markers in case marker shade for test efficiency of algorithm.

CHAPTER VI
CONCLUSION

6.1 Background

Surgical navigation system helps surgeon to perform the operation accurately with more views. The important issue is to know the positions and orientations of the tools and the anatomy in three dimensions. To extract these factors, we chose the optical tracking to detect some markers that represent the medical devices.

6.2 Objective

The main objective of this research is to find algorithms that are suitable for the implementation of surgical navigation applications. The mission is to develop a high precision 3-D coordinate experimental platform that allows the measurement of position and orientation errors.

6.3 System of the experiment

This experiment is about to find the position of marker in 3D.By the way we will take the image from marker in right and left camera. As the marker image from same scene but position are different, to find position that we see the marker picture from right and left camera. In which position is the same position. This experiment that compute from algorithm for match marker of two image by use the principal of Fundamental Matrix to identify each markers. We have 12 markers but in this experiment we adjust to 5 cases. In each case divide to 6 patterns, total are 30 times.

6.4 Error Analysis

When we get the result already, use the result to find accurate value. By get it from Root Mean Square Error (RMSE). The results of RMSE of experiment divide RMSE of x-axis, RMSE of y-axis and RMSE z-axis that all of them have RMSE less than 0.5 mm. that our system had the accuracy.

REFERENCES

1. Abdel-Aziz YI and Karara HM, Direct linear transformation onto object space coordinates in close-range photogrammetry. In Proc. of the Symposium on Close-Range Photogrammetry, Urbana, Illinois, 1971, p.1-18.
2. Bakstein H, Camer calibration toolbox, March 1999.
3. Beyer H. Geometric and radiometric Analysis of a CCD-Camera Based Photogrammetric Close-Range System. PhD thesis, ETH, Zurich, 1992.
4. Champleboux G,Lavallee S,Szelinski R, and Brunie L. From accouate range imaging sensor calibration to accurate model-based 3D object localization. In Proceeding, 1992 IEEE Computer society Conference on Computer Vision and Pattern Recognition, 1992, p.83-88.
5. Davies ER. Machine Vision: Theory, Algorithms, Practicalities. Academic Press, 2nd edition, 1997.
6. Devernay F. A Non-Maxima Suppresion Method For Edge Detection with Sub-Pixel Accuracy. Technical Report 2724, INRIA, 1995.
7. Delherm C, Lavest JM, Dhome M, and Lapreste JT, Dense reconstruction by zooming. In B. Buxtion and R. Cipolla, editors, Proc. Of the 4th European Conference on Computer Vision(ECCV’96), Cambridge, England Springer-Verlag, 1996, p.427-438, Vol. 2.
8. Faugeras O. Three-dimensional computer vision – A geometric viewpoint. MIT Press, 1993.
9. Haralick RM and Shapiro LG. Computer and Robot Vision. Addison-Wesley, 1993.
10. Heikkila J and Silven O. Calibration procedure for short focal length off-the-shelf CCD cameras. In Proc. of the 13th International Conference on Pattern Recognition (ICPR’96), Vienna, Austria, 1996, p.166-170.
11. Heikkila J and Silven O. A four-step camera calibration procedure with implicit image correction. In Proc. of the IEEE Computer Society Conference on

Computer Vision and Pattern Recognition (CVPR’97), Dan Juan, Puerto Rico, 1997.

12. Kolb C, Hanrahan P, and Mitchell P. A realistic camera model for computer graphics. In R. Cook, editor, Proc. of the SIGGRAPH 95 Conference, p.317-324. ACM SIGGRAPH, Addison Wesley, August 1995.
13. Kuntsevitch A and Kappel F. Solvopt – the solver for local nonlinear optimization problems,1997.Available From: URL:http://bedvgm.kfunigraz.ac.at:8001/alex/solvopt/indes.html.
14. Lenz RK and Tsai RY. Techniques for calibration of the scale factor and image center for high accuracy 3d machine vision metrology. IEE Trans. On Pattern Analysis and Machine Intelligence, 1988, 10(10):713-720.
15. Lavest JM, Viala M, and Dhome M. Do We Really Need an Accurate Calibration Pattern to Achieve a Reliable Camera Calibration? In H. Burkhardt and B. Neumannn, editors, Proc. of the 5th Eurporan Conference on Computer Vision (ECCV’98), Freiburg, Germany, June 1998, Vol.1, 158-174.
16. Melen T. Extracting physical camera parameters from the 3 by 3 direct linear transformation matrix. In A. Gruen and H. Kahmen, editors, Optical measurements II, Zurich, Switzerland, 1993, p.355-365.
17. Melen T. Geometrical modeling and calibration of video cameras for underwater navigation. PhD thesis, Institut for teknisk kybernetikk, Norges tekniske hogslole, Trondheim, November 1994.
18. Nalwa VS, A guided tour of computer vision, Addison-Wesley, 1993.
19. Rektorys K et al. A survey of aoolied mathematics. SNTL, Orague, 1982.
20. Shirai Y. Three-Dimensional Computer Vision. Springer-Verlag, 1987.
21. Slama CC. Manual of Photogrammetry. American Society of Photogrammetry, Falls Chourch, Virginia, 4th edutuibm 1980.
22. The MathWorks Inc. MATLAB: A system for numerical computation and visualization. http://www.mathworks.com, 1984.
23. Triggs B. Autocalibration from Planar Scenes. In H. Burkhadt and B. Neumann, deitors, Proc of the 5th European Conference on Computer Vision (ECCV’98), Freibury, Germany, June 1998, Vol. 1, p.158-174.

24. Tsai RY. A versatile camera calibration technique for highaccuracy 3D machine metrology using off-the-shelf cameras and lenses. IEEE Journal of Robotics and Automation, 1987, 3(4):323-344.

25. Weng J, Cohen P, and Herniou M. Camera calibration with distortion models and accuracy evaluation. IEEE Trans. On Patern Analysis and Machine Intelligence, October 1992, 14(10):965-980.

26. Willson R. Tsai camera calibration software, 1995. [cited 2008 Mar 3]; Available from: URL: http://www.cs.cmu.edu/afs/cs.cmu.deu/afs/cs.cmu.due/user/rgw/www/TsaiCode.html.

27. Wei GQ and Ma SD. A complete two-plane camera calibration method and experimental comparsion. In Proc. of the 4th International Conference on Computer Vision (ICCV'93), Berlin, Germany, 1993, p.439-446.

28. Zeller and Faugeras O. Camera self-calibration from video sequences: the Kruppa equation revisited. Technical Report 2793, INRIA, 1996.

29. Kusalanukhun T. Optical Three-Dimensional Coordination For Medical Applications. M.Eng. (Biomedical Engineering). Mahidol University, 2006.

30. Tsai RY, A Versatile Camera Calibration Technique For High Accuracy 3-D Machine Vision Metrology Using Off-The-Shelf TV Cameras And Lenses, IEEE J. Robotics Automat. , 1987,pp. 323-344, Vol. RA-3, No. 4.

31. Jean-Yves Bouguet, Camera Calibration Toolbox for Matlab. [cited 2008 Mar 3]; Available from: URL: http://www.vision.caltech.edu/bouguetj/calib_doc/

32. "Computer Vision: A Modern Approach". D. Forsyth and J. Ponce, Prentice Hall, 2003.

33. "Introductory Techniques for 3-D Computer Vision". E. Trucco and A. Verri, Prentice Hall, 2000.

34. "Geometric Frame Work for Vision – Lecture Notes". A. Zisserman, University of Oxford.

35. Hazan EJ and Joskowicz L, Computer-Assisted Image-Guided Intramedullary Nailing of Femoral Shaft Fractures, Techniques in Orthopaedics, 2003, Vol. 18(2).

36. Eyke JC et al., Computer-Assisted Virtual Fluoroscopy, The University of Pennsylvania Orthopaedic Journal, 2002, Vol. 15: 53–59.
37. Heikkila J, Geometric Camera Calibration Using Circular Control Points, IEEE Transactions on Pattern Analysis and Machine Intelligence, October 2000, Vol. 22, No. 10.
38. Heikkila J and SilvCn O, A Four-step Camera Calibration Procedure with Implicit Image Correction.
39. Abdel-Aziz YI and Karara HM, Direct linear transformation into object shape coordinates in close-range photogrammetry, In Proc. Symp. Close-Range Photogrammetry, Urbana, IL, 1971, pp 1-18.
40. Bacakoglu H and Kamel MS, A Three-Step Camera Calibration method, IEEE Transaction on Instrumentation and Measurement, 1997, Vol.46, No 5.
41. Hartley RI and Sturm P, Triangulation, in Proc. ARPA Image Understanding Workshop, 1994, pp. 957-966.
42. Iocchi L, Stereo Vision Triangulation. [cited 2008 Mar 3]; Available from: URL:http://www.dis.uniroma1.it/~iocchi/stereo/triang.html
43. Chuckpaiwong I, Motion Detection of Marker, Final Report: Human Analysis for Medical Applications, 2006.
44. Dorfmuller K, Robust Tracking for Augmented Reality using Retroreflective Markers, Computers & Graphics 23(6) , 1999, pp. 795-800.
45. Neatpisarnvanit C, Wireless Active Marker System, Final Report: Human Analysis for Medical Applications, 2006.
46. PIC12F509 Microcontroller, Datasheet, Microship Technology Inc. [cited 2008 Mar 3]; Available from: URL: http://www.microship.com.
47. Easy Link Wireless: TLP-434 and RLP-434 modules, Laipac Technology Inc. [cited 2008 Mar 3]; Available from: URL: http://www.laipac.com
48. West JB, Maurer CR, Designing optically tracked instruments for image-guided surgery, IEEE Transactions on Medical Imaging, 2004, Vol. 23, Issue 5, pp. 533 – 545.
49. Schwald B and Figueiredo PM, Learning of Rigid Point-Based Marker Models for Tracking with Stereo Camera Systems.

50. Fisher R, Perkins S,Walker A, Wolfart E, Connected Components Labeling: Image Processing Learning Resources, 2004, [cited 2008 Mar 3]; Available from: URL: http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm.
51. Getting Started, Image Acquisition Toolbox, MATLAB® Help Document.
52. Microsoft Window SDK, DirectShow. [cited 2008 Apr 11]; Available from: URL: http://windowssdk.msdn.microsoft.com/library/default.asp?url=/library/en-us/directshow/htm/introductiontodirectshowapplicationprogramming.asp
53. Open Source Computer Vision Library, Reference Manual, Intel Corporation, 2001.
54. Shreiner D, Woo M, Neider J,and Davis T, OpenGl Programming Guide : The Official Guide to Learning OpenGL, Version 1.1, Addisson Wesley Longman, Inc, 1997.
55. O'Sullivan C, Viewing Systems & OpenGL. [cited 2008 Apr 9]; Available from: URL: http://isg.cs.tcd.ie/cosulliv/4BA6/Viewing_6pp.pdf.
56. Sepulveda MA, OpenGL Programming: The 3D Scene, [cited 2008 Apr 11]; Available from: URL: http://www.linuxfocus.org/English/May1998/article46.html
57. Craig JJ, Introduction to Robotics Mechanics and Control, Pearson Prentice Hall, 3rd Edition, 2005.
58. Bouguet JY, Camera Calibration Toolbox for Matlab. [cited 2008 Apr 11]; Available from: URL: http://www.vision.caltech.edu/bouguetj/calib_doc/
59. Zhang Z, A Flexible New Technique for Camera Calibration Technical Report MSR-TR-98-71 Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA 98052.
60. Williams AV, Computer Vision Course CMSC 828D: Fundamentals of Computer Vision 1112. [cited 2008 May 9]; Available from URL: http://www.umiacs.umd.edu/~ramani/cmsc828.html
61. Naughter PJ, CSerialPort v1.03 - Serial Port Wrapper, A Freeware MFC Class for Win32 Serial Ports. cited 2008 May 9]; Available from: URL: http://www.codeproject.com/system/cserialport.asp

APPENDIX

Case 1	X-axis		Y-axis		Z-axis	
(A→B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4750	-86.4966	3.8380	3.8086	53.4390	53.4044
Marker 2	-54.3370	-54.4203	55.3520	55.4612	28.3460	28.3889
Marker 3	-34.0400	-34.0337	52.7160	52.7092	85.8310	85.8937
Marker 4	-31.1070	-31.0926	35.5910	35.5967	38.9290	38.8493
Marker 5	-21.7140	-21.7713	-21.7150	-21.6617	33.3330	33.2610
Marker 6	-20.6570	-20.5975	20.8700	20.8730	59.8560	59.8846
Marker 7	8.8850	8.9445	-8.9820	-8.9868	36.5630	36.5430
Marker 8	23.2620	23.2601	23.1620	23.1204	46.2900	46.3245
Marker 9	65.0790	65.0954	53.4050	53.4197	30.8790	30.9198
Marker 10	79.5360	79.5447	-12.5650	-12.6318	42.7280	42.7636
Marker 11	108.5380	108.5287	-2.8390	-2.8033	75.1610	75.2255
Marker 12	119.9820	120.0183	48.2500	48.3312	34.8000	34.8334
RMSE		**0.0400**		**0.0317**		**0.0464**

Case 1	X-axis		Y-axis		Z-axis	
(A→B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5697	-86.5970	3.8509	3.8469	53.3400	53.3467
Marker 2	-54.3744	-54.4168	55.4176	55.4944	28.4800	28.5596
Marker 3	-34.1586	-34.1709	52.5992	52.5689	85.8600	85.9109
Marker 4	-31.2126	-31.1794	35.5449	35.4776	39.0769	38.9979
Marker 5	-21.5668	-21.6095	-21.7412	-21.7178	33.4468	33.4429
Marker 6	-20.6514	-20.7115	20.7487	20.7035	59.7876	59.7535
Marker 7	8.7633	8.7573	-9.1139	-9.1121	36.4338	36.3826
Marker 8	23.2579	23.2546	23.2551	23.2237	46.2827	46.2210
Marker 9	64.9662	64.9904	53.4061	53.4329	30.8459	30.8604
Marker 10	79.4011	79.3713	-12.6295	-12.6019	42.6436	42.6222
Marker 11	108.5119	108.5044	-2.7584	-2.7686	75.2108	75.2136
Marker 12	120.0773	120.0556	48.2732	48.1704	34.9488	34.9305
RMSE		**0.0306**		**0.0475**		**0.0446**

Case 1	**X-axis**		**Y-axis**		**Z-axis**	
(A→B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5680	-86.5208	4.2295	4.2588	53.4562	53.4805
Marker 2	-54.2628	-54.3688	55.4529	55.4403	27.9451	27.9448
Marker 3	-33.8943	-33.9266	53.0889	53.1129	85.7324	85.7186
Marker 4	-30.6846	-30.7198	35.5230	35.5564	39.0214	39.0852
Marker 5	-21.9855	-22.0364	-21.9430	-21.9469	33.2688	33.3620
Marker 6	-20.8615	-20.8706	20.8278	20.8722	60.1033	60.0772
Marker 7	9.0926	9.1686	-8.7440	-8.6285	36.4367	36.4419
Marker 8	23.1840	23.1821	22.9388	22.9650	45.8250	45.7846
Marker 9	64.8027	64.8641	53.5321	53.5315	30.6327	30.6667
Marker 10	79.5991	79.5643	-12.6853	-12.6396	42.9391	42.8209
Marker 11	108.8486	108.8490	-2.7288	-2.7260	75.1384	75.1879
Marker 12	120.1236	120.0844	48.0300	47.9747	34.8758	34.8868
RMSE		**0.0506**		**0.0446**		**0.0529**

Case 1	**X-axis**		**Y-axis**		**Z-axis**	
(A→B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4358	-86.4631	3.8363	3.7915	53.2858	53.2721
Marker 2	-54.1550	-54.1419	55.3519	55.3586	28.2857	28.3964
Marker 3	-34.0812	-34.0819	52.6786	52.6716	85.8570	85.9325
Marker 4	-31.1270	-31.1560	35.6505	35.5923	38.9116	38.8143
Marker 5	-21.9046	-21.7978	-21.7546	-21.6954	33.4926	33.4086
Marker 6	-20.9065	-20.9194	20.6204	20.6196	59.8192	59.7905
Marker 7	8.7795	8.7090	-9.1363	-9.1095	36.3354	36.3261
Marker 8	23.3041	23.3926	23.1985	23.1626	46.2915	46.2920
Marker 9	64.9978	65.0141	53.2165	53.1837	30.8897	30.9316
Marker 10	79.3360	79.2800	-12.6171	-12.6014	42.7755	42.7394
Marker 11	108.6989	108.7299	-2.9802	-2.9749	75.2360	75.1999
Marker 12	119.8752	119.9387	48.0738	48.1662	34.9917	34.9817
RMSE		**0.0536**		**0.0417**		**0.0577**

Case 1	X-axis		Y-axis		Z-axis	
(A→B→C)	Ref.	Actual	Ref.	Actual	Ref.	Actual
Marker 1	-86.4801	-86.5292	3.2257	3.3075	53.1997	53.2348
Marker 2	-54.2673	-54.3017	55.4703	55.4075	28.6691	28.6448
Marker 3	-33.7754	-33.7085	52.7452	52.7346	85.9412	86.0344
Marker 4	-30.9516	-30.9970	35.4432	35.4333	39.2492	39.3046
Marker 5	-22.1517	-22.1723	-21.8787	-21.8633	33.2086	33.1472
Marker 6	-20.4827	-20.5080	20.5998	20.5712	59.5763	59.5428
Marker 7	9.0879	9.1689	-8.9939	-9.0428	36.7649	36.8320
Marker 8	23.4211	23.4251	23.2568	23.2345	46.3003	46.3197
Marker 9	65.4065	65.3525	53.3224	53.3765	30.6899	30.7096
Marker 10	79.6178	79.5615	-12.6900	-12.5713	42.7057	42.6204
Marker 11	108.3698	108.4565	-2.8480	-2.8365	74.6588	74.6702
Marker 12	119.9447	120.0415	48.2063	48.1930	35.0710	35.1053
RMSE		**0.0583**		**0.0517**		**0.0519**

Case 1	**X-axis**		**Y-axis**		**Z-axis**	
(A→B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.6660	-86.6765	4.0395	3.9913	53.8532	53.8647
Marker 2	-54.6378	-54.6000	55.1996	55.0806	28.7420	28.6940
Marker 3	-34.0957	-34.0769	52.9729	52.9310	85.5582	85.5509
Marker 4	-31.4232	-31.4905	35.6716	35.6844	38.2373	38.2745
Marker 5	-21.7355	-21.6614	-21.5275	-21.5367	33.8696	33.8251
Marker 6	-20.5732	-20.5716	20.5558	20.5474	59.9732	59.9802
Marker 7	9.2970	9.3905	-8.5213	-8.5271	36.5691	36.5573
Marker 8	23.3160	23.2555	23.2923	23.3008	46.1682	46.1644
Marker 9	64.9164	64.8773	52.8299	52.8048	30.4185	30.4006
Marker 10	80.0263	79.9879	-12.4240	-12.4593	42.7944	42.6905
Marker 11	108.7856	108.7802	-2.4567	-2.4313	74.7487	74.7415
Marker 12	120.0512	120.0024	48.4416	48.4205	34.5482	34.6179
RMSE		**0.0498**		**0.0424**		**0.04289**

Case 2	X-axis		Y-axis		Z-axis	
(A&B→C)	Ref.	Actual	Ref.	Actual	Ref.	Actual
Marker 1	-86.4750	-86.5320	3.8380	3.7079	53.4390	53.4555
Marker 2	-54.3370	-54.4869	55.3520	55.2915	28.3460	28.3759
Marker 3	-34.0400	-34.0450	52.7160	52.5671	85.8310	85.8384
Marker 4	-31.1070	-31.0517	35.5910	35.6469	38.9290	38.9239
Marker 5	-21.7140	-21.7057	-21.7150	-21.7427	33.3330	33.2013
Marker 6	-20.6570	-20.4992	20.8700	20.7406	59.8560	59.8574
Marker 7	8.8850	8.8519	-8.9820	-9.0708	36.5630	36.5192
Marker 8	23.2620	23.3415	23.1620	23.0633	46.2900	46.2767
Marker 9	65.0790	65.0005	53.4050	53.3978	30.8790	30.8626
Marker 10	79.5360	79.4097	-12.5650	-12.8065	42.7280	42.6701
Marker 11	108.5380	108.6047	-2.8390	-2.9084	75.1610	75.1900
Marker 12	119.9820	119.8427	48.2500	48.1109	34.8000	34.8120
RMSE		**0.0945**		**0.1167**		**0.0459**

Case 2	**X-axis**		**Y-axis**		**Z-axis**	
(A&B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4098	-86.4244	3.5956	3.7707	53.2527	53.3157
Marker 2	-54.3747	-54.3499	55.3296	55.4049	28.3006	28.3028
Marker 3	-34.1061	-34.1138	52.7218	52.7283	85.7658	85.7501
Marker 4	-31.0821	-30.9083	35.5485	35.5193	38.9393	38.9507
Marker 5	-21.7524	-21.5902	-21.7353	-21.7270	33.3109	33.3608
Marker 6	-20.7098	-20.6472	20.7187	20.7953	59.8281	59.8889
Marker 7	8.8905	8.8997	-9.0946	-8.8710	36.4896	36.4625
Marker 8	23.3874	23.3066	23.0805	23.1132	46.2835	46.3292
Marker 9	64.8270	64.7809	53.4417	53.5280	30.7346	30.7260
Marker 10	79.5945	79.4539	-12.6236	-12.5557	42.7892	42.7724
Marker 11	108.4372	108.3997	-2.6853	-2.6298	75.0286	75.0557
Marker 12	120.0764	120.0293	48.2640	48.3642	34.7338	34.7804
RMSE		**0.0882**		**0.0998**		**0.0372**

Case 2	X-axis		Y-axis		Z-axis	
(A&B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5276	-86.5283	3.8007	3.9689	53.6103	53.6227
Marker 2	-54.2032	-54.1507	55.3295	55.3889	28.5788	28.5839
Marker 3	-33.8033	-33.6668	52.5272	52.6063	86.0385	86.0365
Marker 4	-31.2732	-31.2250	35.6379	35.6484	38.8153	38.7029
Marker 5	-21.7179	-21.7966	-21.3115	-21.3273	33.3994	33.3739
Marker 6	-20.8236	-20.7484	20.9135	21.0005	59.9927	60.0051
Marker 7	8.9976	8.9809	-9.1954	-9.2149	36.4019	36.4203
Marker 8	23.3370	23.2554	23.1990	23.2066	46.3203	46.3292
Marker 9	65.0014	65.2108	53.1896	53.1370	30.9934	30.9916
Marker 10	79.4521	79.4601	-12.5427	-12.6113	42.5348	42.4546
Marker 11	108.4250	108.3313	-3.0930	-3.1198	75.0181	75.0350
Marker 12	120.1209	120.1844	48.3579	48.2390	34.9167	34.9102
RMSE		**0.0909**		**0.0757**		**0.0416**

Case 2	**X-axis**		**Y-axis**		**Z-axis**	
(A&B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.3780	-86.4490	3.9826	4.0976	53.4427	53.4305
Marker 2	-54.2172	-54.3342	55.3599	55.2991	28.5096	28.5183
Marker 3	-34.0572	-33.9507	53.0243	53.1049	85.9715	86.0176
Marker 4	-31.0419	-31.1100	35.2508	35.2725	38.8827	38.8738
Marker 5	-21.7810	-21.9536	-21.9217	-21.9591	33.3103	33.2842
Marker 6	-20.7215	-20.6402	20.7173	20.6341	59.8816	59.9532
Marker 7	8.8085	8.9527	-8.5467	-8.5180	36.4031	36.3596
Marker 8	23.0713	23.1386	23.2483	23.0664	46.2423	46.2827
Marker 9	65.1257	65.1396	53.3162	53.1589	30.8611	30.8356
Marker 10	79.7830	79.6971	-12.5590	-12.3574	42.5233	42.5605
Marker 11	108.4223	108.3471	-2.9021	-2.9093	75.3485	75.3909
Marker 12	119.8817	120.0047	48.4456	48.7085	34.5737	34.5322
RMSE		**0.1020**		**0.1293**		**0.0380**

Case 2 (A&B→C)	X-axis		Y-axis		Z-axis	
	Ref.	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.3311	-86.2900	3.3727	3.4012	53.6429	53.6126
Marker 2	-54.0581	-54.1888	55.6930	55.7756	28.4832	28.5175
Marker 3	-34.3240	-34.2857	52.5539	52.5531	85.8666	85.8676
Marker 4	-31.0092	-30.9593	35.0337	35.1196	39.0046	39.0578
Marker 5	-21.7239	-21.7750	-21.6851	-21.6076	33.0676	33.0006
Marker 6	-20.9885	-20.9650	21.2717	21.4023	59.6011	59.6251
Marker 7	8.8107	8.7509	-8.9677	-8.8445	36.5595	36.4778
Marker 8	23.2187	23.2208	23.2057	23.3016	46.3857	46.3135
Marker 9	64.7864	64.8284	53.3977	53.2323	30.6374	30.6521
Marker 10	79.0084	79.1275	-12.5181	-12.6172	42.9473	42.9403
Marker 11	108.2924	108.3696	-2.6008	-2.5323	75.1906	75.1341
Marker 12	119.9805	119.7161	48.2992	48.2017	35.5375	35.5229
RMSE		**0.1005**		**0.0974**		**0.0461**

Case 2	**X-axis**		**Y-axis**		**Z-axis**	
(A&B→C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.6789	-86.6383	4.1529	4.2289	53.5785	53.5384
Marker 2	-54.6507	-54.5291	54.6089	54.4376	28.7367	28.6535
Marker 3	-33.9530	-33.8081	52.8156	52.9694	86.0482	86.0031
Marker 4	-31.6284	-31.7309	35.3343	35.1734	38.8283	38.8577
Marker 5	-21.6043	-21.5837	-21.9857	-21.8747	32.8529	32.8806
Marker 6	-21.3658	-21.3069	20.9231	20.8122	59.6159	59.5952
Marker 7	9.0701	9.0437	-9.0999	-9.0613	36.6791	36.6822
Marker 8	23.3822	23.6317	23.5018	23.5983	45.9408	45.9637
Marker 9	65.3444	65.4300	53.3675	53.4493	30.9810	30.9909
Marker 10	79.2938	79.2087	-12.2103	-12.2066	43.1155	43.1284
Marker 11	108.7761	108.8573	-3.0054	-3.0980	75.2467	75.3508
Marker 12	119.6066	119.6766	48.2741	48.2629	34.8577	34.7439
RMSE		**0.1083**		**0.1063**		**0.0557**

Case 3	X-axis		Y-axis		Z-axis	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4750	-86.4309	3.8380	3.7700	53.4390	53.2037
Marker 2	-54.3370	-54.2805	55.3520	55.5619	28.3460	28.4242
Marker 3	-34.0400	-34.1094	52.7160	52.6467	85.8310	85.9380
Marker 4	-31.1070	-31.0236	35.5910	35.5959	38.9290	38.9936
Marker 5	-21.7140	-21.9377	-21.7150	-21.5952	33.3330	33.1607
Marker 6	-20.6570	-20.5472	20.8700	21.0045	59.8560	59.7970
Marker 7	8.8850	8.8848	-8.9820	-8.9613	36.5630	36.6381
Marker 8	23.2620	23.1005	23.1620	22.9191	46.2900	46.01967
Marker 9	65.0790	64.9561	53.4050	53.1580	30.8790	30.7598
Marker 10	79.5360	79.5567	-12.5650	-12.5007	42.7280	42.5448
Marker 11	108.5380	108.5610	-2.8390	-2.9496	75.1610	75.3314
Marker 12	119.9820	119.8814	48.2500	48.3347	34.8000	35.0940
RMSE		**0.1046**		**0.1384**		**0.1713**

Case 3	**X-axis**		**Y-axis**		**Z-axis**	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4041	-86.2782	3.7782	3.5662	53.33819	53.3492
Marker 2	-54.4118	-54.3259	55.3038	55.2462	28.2795	27.9807
Marker 3	-34.0171	-34.2276	52.8143	52.7456	85.8868	86.1736
Marker 4	-31.1293	-31.1654	35.7672	35.7235	38.8101	38.8658
Marker 5	-21.7993	-21.7440	-21.5723	-21.6174	33.2555	33.3772
Marker 6	-20.6224	-20.7781	20.9612	20.7229	59.8831	60.1549
Marker 7	8.8960	8.8753	-8.9493	-8.7852	36.7165	36.3655
Marker 8	23.1487	23.1061	23.1690	23.3676	46.1848	46.0584
Marker 9	65.0107	65.0601	53.2550	53.2361	30.9416	30.9715
Marker 10	79.5082	79.4211	-12.6068	-12.7174	42.6482	42.7380
Marker 11	108.6035	108.6115	-2.8411	-2.8090	75.1296	75.0356
Marker 12	119.8572	119.8050	48.2728	48.2128	34.7398	34.7905
RMSE		**0.0963**		**0.1282**		**0.1877**

Case 3	X-axis		Y-axis		Z-axis	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.6958	-86.7866	3.8108	3.7053	53.2751	53.4464
Marker 2	-54.3711	-54.4755	55.4088	55.4934	28.0524	28.0793
Marker 3	-34.0943	-34.0570	52.9579	52.9503	85.7664	85.6666
Marker 4	-30.9347	-30.8446	35.1882	35.3627	38.8574	39.1250
Marker 5	-21.8100	-21.6822	-21.4823	-21.3835	33.1240	33.0361
Marker 6	-20.8192	-20.8320	20.8019	20.5694	59.6562	59.8650
Marker 7	8.7243	8.7856	-8.9815	-9.4359	36.6153	36.8846
Marker 8	23.3122	23.5078	23.3212	23.4023	45.9462	45.5910
Marker 9	64.9446	65.1714	53.5098	53.3585	31.1157	31.6171
Marker 10	79.8246	79.7872	-12.8556	-12.7194	42.8478	43.0783
Marker 11	108.6414	108.8652	-3.1097	-2.8723	75.1115	75.2900
Marker 12	120.0397	120.0238	48.3532	48.2064	34.7959	34.6654
RMSE		**0.1259**		**0.1926**		**0.2441**

Case 3	**X-axis**		**Y-axis**		**Z-axis**	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5523	-86.6243	3.9724	3.8922	53.3372	53.3807
Marker 2	-54.7486	-54.6728	55.5264	56.0067	28.5650	28.5326
Marker 3	-34.4203	-34.5846	52.9730	53.0389	85.9192	85.9058
Marker 4	-31.3755	-31.4812	35.5111	35.3386	38.6743	38.7932
Marker 5	-21.5373	-21.5624	-21.8402	-21.7072	32.5728	32.4374
Marker 6	-20.1042	-20.2341	20.8083	20.7658	59.1423	59.0061
Marker 7	9.2894	9.4127	-9.0343	-8.8790	36.4591	36.4578
Marker 8	23.1146	23.2641	23.2273	23.1725	46.1067	45.9832
Marker 9	64.4257	64.4493	53.9103	54.1116	30.7565	30.5031
Marker 10	79.6071	79.4667	-12.5291	-12.3778	42.3033	42.4513
Marker 11	108.3174	108.3833	-2.6438	-2.6117	75.2291	75.1840
Marker 12	119.4482	119.1925	48.8740	48.8291	34.8624	34.5961
RMSE		**0.1272**		**0.1790**		**0.1380**

Case 3	X-axis		Y-axis		Z-axis	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.6329	-86.6978	3.3048	3.2939	54.2222	54.2600
Marker 2	-54.3945	-54.3323	55.0624	54.9133	28.3169	28.6350
Marker 3	-34.9737	-35.1072	52.4506	52.3385	86.2065	86.2252
Marker 4	-31.6500	-31.5452	35.6704	35.6658	38.8972	38.8272
Marker 5	-22.4403	-22.3539	-22.4329	-22.2847	33.0126	33.0995
Marker 6	-20.2136	-20.2779	20.3149	20.2251	59.9798	59.7759
Marker 7	8.8282	8.8942	-8.9029	-8.6814	36.9837	36.9673
Marker 8	23.7071	23.8365	23.7497	23.6277	45.6243	45.7001
Marker 9	65.3027	65.3342	53.5516	53.6484	30.4427	30.3924
Marker 10	79.7273	79.8133	-12.7421	-12.9386	42.6513	42.6227
Marker 11	108.2662	108.2791	-2.8584	-2.9885	75.3464	75.3769
Marker 12	120.0960	120.0976	48.2811	48.2099	34.4303	34.5019
RMSE		**0.0807**		**0.1288**		**0.1199**

Case 3	**X-axis**		**Y-axis**		**Z-axis**	
(A→B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-85.6362	-85.7034	2.9179	2.9081	53.1763	53.1099
Marker 2	-53.3385	-53.4288	54.5379	54.6387	28.7048	28.5613
Marker 3	-33.6915	-33.7070	52.1291	52.1600	86.3752	86.0798
Marker 4	-31.7902	-31.6955	35.3833	35.3821	39.1793	39.1029
Marker 5	-21.5325	-21.3774	-21.6275	-21.6244	34.7189	34.4557
Marker 6	-20.9405	-20.8976	20.9847	20.9009	59.7758	59.7596
Marker 7	8.3629	8.3068	-9.6024	-9.3195	36.7777	36.9321
Marker 8	23.6106	23.6285	23.5120	23.4790	45.3066	45.7094
Marker 9	65.3210	65.2439	53.6184	53.4063	30.6060	30.6944
Marker 10	79.4391	79.3448	-11.8376	-11.8830	41.7838	41.8536
Marker 11	108.3490	108.2082	-3.0941	-3.1795	75.1070	75.2860
Marker 12	119.5388	119.3482	48.2467	48.2284	34.1419	34.2148
RMSE		**0.1009**		**0.1133**		**0.1879**

Case 4	X-axis		Y-axis		Z-axis	
(A&C→B)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4750	-86.2807	3.8380	3.8005	53.4390	53.5502
Marker 2	-54.3370	-54.3649	55.3520	55.3728	28.3460	28.4678
Marker 3	-34.0400	-34.0204	52.7160	52.6394	85.8310	85.8096
Marker 4	-31.1070	-31.2056	35.5910	35.5804	38.9290	38.9140
Marker 5	-21.7140	-21.8279	-21.7150	-21.6811	33.3330	33.2130
Marker 6	-20.6570	-20.7463	20.8700	20.9733	59.8560	59.9300
Marker 7	8.8850	9.0069	-8.9820	-9.1225	36.5630	36.7486
Marker 8	23.2620	23.2724	23.1620	23.0589	46.2900	46.4844
Marker 9	65.0790	64.8039	53.4050	53.3408	30.8790	30.8373
Marker 10	79.5360	79.8101	-12.5650	-12.5479	42.7280	42.7606
Marker 11	108.5380	108.6361	-2.8390	-2.7045	75.1610	75.2556
Marker 12	119.9820	119.7503	48.2500	48.4436	34.8000	34.7177
RMSE		**0.1578**		**0.0957**		**0.1074**

Case 4	**X-axis**		**Y-axis**		**Z-axis**	
(A&C→B)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4520	-86.5534	3.9026	3.9071	53.4781	53.6294
Marker 2	-54.3015	-53.8834	55.2391	55.4807	28.5533	28.8607
Marker 3	-33.9879	-33.9904	52.7357	52.7047	85.7987	85.8890
Marker 4	-31.1686	-31.1278	35.7607	35.7794	39.0758	39.0785
Marker 5	-21.5794	-21.7165	-21.6424	-21.5476	33.2828	33.0412
Marker 6	-20.5595	-20.8522	20.9493	20.8967	59.8770	60.0628
Marker 7	8.6472	8.5996	-8.9217	-9.0332	36.6385	36.7410
Marker 8	23.1528	23.2410	23.1562	22.9969	46.1952	46.0781
Marker 9	65.0464	65.1708	53.2941	53.4116	30.9403	31.0200
Marker 10	79.3348	79.0835	-12.3506	-12.3021	42.9041	43.2242
Marker 11	108.6948	108.4064	-2.9743	-2.8097	75.1699	75.2230
Marker 12	120.0053	119.9398	48.2957	48.2503	35.0596	35.0943
RMSE		**0.1973**		**0.1135**		**0.1721**

Case 4	X-axis		Y-axis		Z-axis	
(A&C→B)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5755	-86.5662	3.6250	3.6930	53.5968	53.5934
Marker 2	-54.1085	-54.2188	55.5158	55.5391	28.3668	28.3834
Marker 3	-34.0740	-34.1009	53.0835	53.2075	86.3660	86.4879
Marker 4	-30.7785	-30.6158	35.3884	35.4010	39.2449	39.0935
Marker 5	-21.6334	-21.6129	-22.0340	-22.0161	32.8953	32.74467
Marker 6	-20.7761	-20.7785	21.3006	21.2401	59.8524	59.8948
Marker 7	8.9040	8.7633	-8.9565	-9.0602	37.1800	37.2235
Marker 8	23.2357	23.0140	22.9615	22.9320	46.1227	46.0856
Marker 9	65.4333	65.4875	53.0923	53.2379	30.9440	30.9112
Marker 10	79.7130	79.7846	-12.2556	-12.0753	42.6349	42.7697
Marker 11	108.6300	108.6782	-2.4570	-2.5903	75.4906	75.3943
Marker 12	119.8313	119.5497	48.6942	48.7330	34.6367	34.6097
RMSE		**0.1286**		**0.0954**		**0.0890**

Case 4	**X-axis**		**Y-axis**		**Z-axis**	
(A&C→B)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.6371	-86.5662	4.1464	4.1618	53.4046	53.5888
Marker 2	-54.0507	-53.7833	55.7574	55.7372	28.3882	28.3787
Marker 3	-33.9209	-33.8819	52.6337	52.7825	85.8516	85.9484
Marker 4	-31.1335	-30.9069	35.3575	35.2953	39.0516	38.7859
Marker 5	-21.4667	-21.4192	-21.9110	-21.8300	33.2987	33.3076
Marker 6	-21.2483	-21.1278	20.7834	20.9763	59.6261	59.5120
Marker 7	9.2854	9.3725	-8.5719	-8.5323	36.4353	36.1817
Marker 8	23.2420	23.5088	22.6509	22.5647	46.2953	46.4609
Marker 9	64.9495	64.8088	53.2774	53.5206	31.3308	31.5502
Marker 10	79.7372	79.5996	-12.4742	-12.5583	42.8656	42.9010
Marker 11	108.7504	108.8068	-2.5664	-2.5383	75.1837	75.0190
Marker 12	120.3184	120.4549	47.9268	48.0089	34.8971	35.2594
RMSE		**0.1542**		**0.1130**		**0.1885**

Case 4	**X-axis**		**Y-axis**		**Z-axis**	
(A&C→B)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.7859	-86.7798	4.5268	4.4609	52.5975	52.9621
Marker 2	-53.7169	-53.6046	55.1871	55.1756	28.0801	28.3832
Marker 3	-33.6180	-33.4334	52.9421	52.9721	86.4110	86.5306
Marker 4	-31.1737	-31.1279	35.8869	35.8285	38.7971	38.8017
Marker 5	-21.5882	-21.4265	-21.6269	-21.9343	34.4137	34.4949
Marker 6	-20.0892	-19.9744	21.3951	21.5502	59.2020	59.3046
Marker 7	9.0159	8.8180	-8.7303	-8.7711	36.3485	36.2600
Marker 8	23.4523	23.3759	22.7188	22.8616	46.5089	46.4697
Marker 9	65.2385	65.3218	53.2262	53.0909	31.4760	31.7036
Marker 10	79.5069	79.2132	-12.8554	-12.7650	42.5459	42.6970
Marker 11	109.0639	108.9498	-2.6974	-2.6432	74.9624	75.0079
Marker 12	120.3733	120.0067	48.0473	48.0008	35.2941	35.1715
RMSE		**0.1765**		**0.1228**		**0.1725**

Case 4	X-axis		Y-axis		Z-axis	
(A&C→B)	Ref.	Actual	Ref.	Actual	Ref.	Actual
Marker 1	-85.8310	-85.9514	4.0727	4.0650	52.8958	52.6801
Marker 2	-54.3437	-54.3450	56.0018	55.8460	27.5668	27.3367
Marker 3	-34.7067	-34.6125	53.5334	53.7037	86.1497	86.1385
Marker 4	-31.3852	-31.3620	35.2396	35.1927	38.7267	38.7389
Marker 5	-21.3362	-20.9491	-21.3114	-21.3019	33.1313	33.0049
Marker 6	-21.1129	-21.3089	20.3563	20.3850	59.8980	59.8133
Marker 7	9.5708	9.7244	-8.3348	-8.2428	36.3453	36.3411
Marker 8	23.3848	23.5015	23.1695	23.2205	46.0087	45.8221
Marker 9	65.1384	65.0133	53.5144	53.5389	31.3180	31.4280
Marker 10	79.7283	79.6403	-11.7084	-11.8485	42.3207	42.3296
Marker 11	108.5029	108.5128	-3.8784	-3.7814	75.0318	75.0542
Marker 12	119.6928	119.6910	48.3065	48.4658	35.0466	35.2860
RMSE		**0.1508**		**0.1010**		**0.1378**

Case 5	**X-axis**		**Y-axis**		**Z-axis**	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4750	-86.3314	3.8380	3.9586	53.4390	53.4943
Marker 2	-54.3370	-54.2767	55.3520	55.5406	28.3460	28.4249
Marker 3	-34.0400	-33.7302	52.7160	52.5112	85.8310	85.8113
Marker 4	-31.1070	-31.1158	35.5910	35.5774	38.9290	38.9643
Marker 5	-21.7140	-21.7199	-21.7150	-21.6986	33.3330	32.9654
Marker 6	-20.6570	-20.7334	20.8700	20.5166	59.8560	59.5555
Marker 7	8.8850	8.7742	-8.9820	-9.3383	36.5630	36.7268
Marker 8	23.2620	23.4485	23.1620	23.0300	46.2900	46.2431
Marker 9	65.0790	64.8158	53.4050	53.6753	30.8790	30.5527
Marker 10	79.5360	79.4757	-12.5650	-12.1377	42.7280	42.6644
Marker 11	108.5380	108.0181	-2.8390	-2.8056	75.1610	75.0017
Marker 12	119.9820	120.1380	48.2500	47.9090	34.8000	34.9382
RMSE		**0.2123**		**0.2475**		**0.1872**

Case 5	**X-axis**		**Y-axis**		**Z-axis**	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.1045	-86.0878	3.8900	3.6894	53.4896	53.5881
Marker 2	-54.3445	-54.2578	55.5532	55.3456	28.0578	27.9931
Marker 3	-33.9738	-34.2185	52.7804	52.9061	85.5045	85.6690
Marker 4	-31.3982	-31.9454	35.6835	35.8571	38.8771	38.5942
Marker 5	-21.8263	-21.9334	-21.5576	-21.1432	33.5130	33.7417
Marker 6	-20.7210	-20.2792	21.0071	20.8882	59.9340	60.1298
Marker 7	8.9121	8.7910	-9.1663	-9.0490	36.5378	36.6363
Marker 8	23.3865	23.6736	23.1588	23.4639	46.2850	46.5366
Marker 9	64.8562	65.0951	53.4238	53.8525	31.1498	31.2686
Marker 10	79.4909	79.6396	-12.2279	-12.3771	42.9108	43.0017
Marker 11	108.4395	108.4152	-2.7128	-3.0191	75.1112	74.8276
Marker 12	119.8778	119.8716	48.2607	48.2180	34.7541	34.5701
RMSE		**0.2501**		**0.2456**		**0.1874**

Case 5	X-axis		Y-axis		Z-axis	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4343	-86.8968	3.9939	3.8818	53.2194	53.6725
Marker 2	-54.4134	-54.1373	55.5553	55.8971	28.2005	28.6607
Marker 3	-33.5945	-33.3163	52.6709	52.5420	86.1841	86.1301
Marker 4	-30.9271	-31.0179	35.4339	35.1494	38.7987	38.8993
Marker 5	-21.8075	-21.9332	-21.6162	-21.7674	32.8598	32.8360
Marker 6	-20.7283	-20.5236	20.7218	20.7533	59.6058	59.6054
Marker 7	8.7110	8.6132	-8.8257	-8.7501	36.8121	36.7255
Marker 8	23.0175	22.9370	22.6339	22.6697	46.3011	46.2621
Marker 9	65.2725	65.5524	53.5459	53.4254	30.9907	31.1878
Marker 10	79.5340	79.5901	-12.1327	-12.3314	42.8893	42.9830
Marker 11	107.7780	108.1542	-2.6412	-2.4034	75.6806	75.4076
Marker 12	119.9416	119.7817	48.4001	48.8777	34.9928	35.0475
RMSE		**0.2412**		**0.2229**		**0.2170**

Case 5	X-axis		Y-axis		Z-axis	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.2734	-86.0964	4.2200	4.3097	52.9445	52.9000
Marker 2	-54.7365	-54.5083	54.9865	55.1643	28.3046	27.9303
Marker 3	-34.0186	-33.9380	51.7210	51.9357	86.0844	86.0624
Marker 4	-31.1027	-31.0645	35.9399	35.9189	38.8636	38.9459
Marker 5	-21.7728	-21.9116	-21.6882	-21.3787	33.5878	33.4856
Marker 6	-20.8976	-20.8954	20.2166	20.2506	60.6567	60.8965
Marker 7	8.5383	8.3172	-9.1968	-9.0360	36.5458	36.5650
Marker 8	23.2446	23.6262	23.5713	23.2265	46.3752	46.2860
Marker 9	65.3308	65.2977	53.2037	53.2386	31.3405	31.3996
Marker 10	79.4369	79.2905	-12.6577	-12.7545	42.9842	43.0178
Marker 11	108.0393	107.8411	-2.8373	-2.9837	75.4667	75.4307
Marker 12	120.1435	120.3224	48.2362	47.8098	35.8621	35.7779
RMSE		**0.1823**		**0.2125**		**0.2136**

Case 5	X-axis		Y-axis		Z-axis	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.5784	-86.6953	4.0991	4.1163	53.1636	53.3372
Marker 2	-54.3982	-54.5941	54.7911	54.9294	27.8662	27.7052
Marker 3	-34.5310	-34.5080	52.6643	53.0910	86.2296	86.0791
Marker 4	-31.1162	-31.1026	35.1278	35.1272	38.7591	38.6099
Marker 5	-21.4543	-21.5603	-22.8713	-22.8892	33.1716	33.1097
Marker 6	-19.8733	-19.7651	20.9758	20.9249	60.0584	59.7541
Marker 7	8.9194	9.0557	-9.3424	-9.5173	36.7494	36.9147
Marker 8	23.9374	24.0451	23.6389	23.7234	46.4501	46.3275
Marker 9	64.9191	64.8171	53.2799	53.2532	31.3420	31.5339
Marker 10	78.9902	78.7258	-12.0802	-11.9723	43.0878	43.4823
Marker 11	108.2339	108.1118	-2.3996	-2.2246	74.6661	74.7250
Marker 12	119.7447	119.6316	49.1433	48.8931	34.9688	34.8903
RMSE		**0.1335**		**0.1702**		**0.1919**

Case 5	X-axis		Y-axis		Z-axis	
(A&B&C)	**Ref.**	**Actual**	**Ref.**	**Actual**	**Ref.**	**Actual**
Marker 1	-86.4960	-86.7786	2.9724	3.1301	54.0386	54.1255
Marker 2	-54.1750	-54.2933	55.5914	55.4760	27.0573	27.0522
Marker 3	-33.7868	-33.6830	52.4921	52.5976	84.7879	84.7119
Marker 4	-31.6213	-31.9199	35.7844	36.1187	39.1221	39.0736
Marker 5	-21.6643	-21.6816	-21.6885	-21.5285	32.9025	32.6695
Marker 6	-20.7152	-20.7177	20.6270	20.8037	59.2406	59.3919
Marker 7	9.2296	9.1604	-8.8598	-8.9046	37.8838	38.1166
Marker 8	24.2036	24.4009	23.5212	23.5806	45.8378	45.6331
Marker 9	65.2417	65.3704	53.4817	53.3775	30.2673	30.6076
Marker 10	79.4884	80.0724	-12.4981	-12.5292	42.8796	42.7808
Marker 11	108.5536	108.3039	-3.3421	-3.3618	74.7960	74.8305
Marker 12	119.6751	119.7065	48.9032	49.1027	34.2282	34.2990
RMSE		**0.2343**		**0.1515**		**0.1635**

BIOGRAPHY

NAME	Miss Dareena Darachai
DATE OF BIRTH	30 August 1982
PLACE OF BIRTH	Bangkok, Thailand
INSTUTIONS ATTENDED	University of The Thai Chamber of Commerce, 2000-2003:
	B.Eng. (Computer Engineering)
	Mahidol University, 2004-2010:
	M.Sc. (Technology of Information System Management)
HOME ADDRESS	264, Sathupradit Rd. Bangpongpang, Bangkok.
	Tel. 081-647-2347
	E-mail: mono_boo_boo@hotmail.com