2010

Automation of CAD model based assembly simulations using motion capture

Anup Madhav Vader

Department:

AUTOMATION OF CAD MODEL BASED ASSEMBLY SIMULATIONS USING MOTION CAPTURE

by

ANUP MADHAV VADER

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

2010

Approved by

Ming C. Leu, Advisor

Xiaoqing (Frank) Liu

K. Chandrashekhara

ABSTRACT

The manufacturing industry often uses text and video as a supplement for worker training procedures. Unfortunately, text is often difficult to follow and video lends itself to occlusion issues. CAD based animations of assembly operations can overcome these shortcomings while offering greater potential for operations analysis and simulation flexibility. However, creating such CAD based training simulations manually is a time intensive task and many a times fail to reveal the practical assembly issues faced in the real world. Thus, it is highly beneficial to be able to automate these simulations using motion capture from a physical environment.

This thesis research summary includes the development and demonstration of a low-cost versatile motion tracking system and its application for generating CAD based assembly simulations. The motion tracking system is practically attractive because it is inexpensive, wireless, and easily portable. The system development focus herein is on two important aspects. One is generation of model based simulation, and the other is motion capture. Multiple Wii Remotes (Wiimotes) are used to form vision systems to perform 3D motion tracking. All the 6 DOF of a part can be tracked with the help of four Infra-red (IR) LEDs mounted on the part to be tracked. The obtained data is fed in real-time to automatically generate an assembly simulation of object models represented by Siemens NX5 CAD software.

A Wiimote Vision System Setup Toolkit has been developed to help users in setting up a new vision system using Wiimotes given the required volume and floor area to be tracked. Implementation examples have been developed with different physical assemblies to demonstrate the capabilities of the system.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards my advisor, Dr. Ming C. Leu, for his advice, guidance, and encouragement throughout my graduate studies and particularly during this research. I would also like to express my deepest gratitude to Dr. Xiaoqing (Frank) Liu, for his invaluable contributions and suggestions during the entire tenure of this project.

I would also like to thank Dr. K. Chandrashekhara for his time and effort in serving as a committee member and reviewing this thesis. Special thanks go to my colleagues Dr. Wenjuan Zhu and Abhinav Chadda for their valuable advice and also my friends who have been of wonderful support.

Last, but not the least, I am very thankful to my parents for their continuous love and support.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGEMENTS iv

LIST OF FIGURES viii

LIST OF TABLES xi

SECTION

1. INTRODUCTION 1

    1.1 CALIBRATION TECHNIQUES FOR WIIMOTES: AN OVERVIEW 4

    1.2 6-DOF TRACKING WITH WIIMOTE BASED SYSTEM AND ITS INTEGRATION WITH CAD SOFTWARE 5

    1.3 WIIMOTE VISION SYSTEM SETUP TOOLKIT 6

2. PRINCIPLE OF STEREO VISION AND DISTRIBUTED VISION SYSTEM 8

    2.1 STEREO VISION PRINCIPLE 8

    2.2 COVERAGE OF A SINGLE STEREO VISION SYSTEM 9

    2.3 DISTRIBUTED VISION SYSTEM 10

3. CALIBRATION OF A CAMERA AND VISION SYSTEM 13

    3.1 CAMERA MODEL 13

    3.2 SINGLE STEREO VISION SYSTEM CALIBRATION 15

    3.3 CALIBRATION OF A MULTIPLE STEREO VISION SYSTEM 20

    3.4 DEFINING THE WORLD COORDINATE SYSTEM 26

4. PERFORMANCE EVALUATION OF STEREO VISION SYSTEMS 27

    4.1 EXPERIMENT WITH A SINGLE STEREO VISION SYSTEM 27

4.2 EXPERIMENT WITH AN INTEGRATED TWO STEREO VISION SYSTEM 29

4.3 DATA SMOOTHING 30

4.4 ERROR ANALYSIS AND MEASUREMENT 32

5. SVOBODA'S ALGORITHM FOR CALIBRATION OF A MULTI (>2) CAMERA SYSTEM 36

5.1 THE MULTI-CAMERA SELF-CALIBRATION TECHNIQUE 36

5.2 EVALUATION RESULTS 37

6. AN INTEGRATED CAMERA CALIBRATION TECHNIQUE 40

6.1 DECOMPOSITION OF CAMERA MATRIX INTO MATRICES OF INTRINSIC AND EXTRINSIC PARAMETERS 40

6.2 EXPERIMENT WITH A MULTI-CAMERA VISION SYSTEM WITH 4 WIIMOTES 42

6.3 EXPERIMENT WITH A MULTI-CAMERA VISION SYSTEM WITH 8 WIIMOTES 43

7. ORIENTATION TRACKING WITH 3 LEDS 45

7.1 ESTIMATION OF ROTATION MATRIX FOR KNOWN ABSOLUTE POSITIONS OF 3 POINTS ON A PLANE 46

7.2 SORTING AND SEQUENCING OF LEDS FOR DYNAMIC ORIENTATION TRACKING 47

7.3 ORIENTATION ACCURACY EVALUATION WITH WII MOTION PLUS 53

8. WIIMOTE VISION SYSTEM SETUP TOOLKIT 60

8.1 ESTIMATING THE MINIMUM NUMBER OF WIIMOTES REQUIRED FOR TRACKING 61

8.2 ESTIMATING MAXIMUM POSITION MEASUREMENT ERROR FOR A WIIMOTE VISION SYSTEM 65

8.3 ESTIMATING THE OPTIMUM TRANSFORMATIONS (R,T) FOR INDIVIDUAL DEPENDENT VISION SYSTEMS 69

8.4 AUTOMATION OF CAD ASSEMBLY GENERATION WITH SIEMENS NX5 71

9. APPLICATION TO ASSEMBLY SIMULATION 74

9.1 MOTION TRACKING WITH A WIIMOTE VISION SYSTEM 75

9.2 IMPLEMENTATION WITH AN INTEGRATED TWO VISION SYSTEM 77

10. CONCLUSIONS 80

BIBLIOGRAPHY 82

LIST OF PROJECT PUBLICATIONS 84

APPENDIX 85

VITA 88

LIST OF FIGURES

Figure Page

1.1. Wiimote motion tracking system architecture 3

2.1. Principle of the stereo vision 8

2.2. Stereo system coverage 10

2.3. Distributed vision system 11

3.1. Pinhole camera model 14

3.2. The 36 LED calibration plate 16

3.3. Data collected from the calibration plate by a Wiimote 16

3.4. Data collected by a single Wiimote stereo system 20

3.5. Calibration of an integrated system of two stereo vision system 21

3.6. Coordinates of a number of points measured in two different coordinate systems 22

3.7. Three points defining a triad 23

3.8. Defining the world coordinate system 26

4.1. Measurement accuracy with a single Wiimote stereo vision system 27

4.2. Wiimote motion tracking tested with a CNC machine 28

4.3. Measurement errors in the five experiments (refer Table 1) 29

4.4 Data before and after smoothing 31

4.5 Motion tracking error from theoretical analysis 33

4.6. Theoretic and actual relationships between Z_c and disparity d 34

4.7. Theoretical relationship between Zc and ΔZ 35

5.1. Svoboda's algorithm for self-calibration of 4 Wiimotes 38

5.2. Measurement error for an integrated two stereo vision system calibration using Zhang's and Horn's algorithm 39

5.3. Measurement accuracy for a camera vision system with 4 Wiimotes using Svoboda's calibration algorithm 39

6.1. Calibration of a 4 Wiimote vision system 42

6.2. Measurement accuracy for a multi-camera vision system with 4 Wiimotes using the integrated calibration technique 43

6.3. A multi-camera vision system with 8 Wiimotes 44

6.4. Measurement errors for the multi-camera system with 8 Wiimotes using the integrated calibration technique 44

7.1. Orientation tracking with 3 LED markers 45

7.2. Sequencing of LEDs for multi marker tracking 48

7.3. Wand with 4 LEDs for 6 DOF Motion Tracking 49

7.4. Finding distance of a point from line and position of its projection 50

7.5. Orientation tracking accuracy check with Nintendo Wii MotionPlus accessory 55

7.6. Orientation tracking evaluation with a 4-camera vision system 56

7.7. Estimating pitch, roll and yaw of a plane for known 3 points on the plane 57

7.8. Orientation tracking evaluation for a vision system with 4 Wiimotes 58

8.1. Wiimote Vision System Setup Toolkit 60

8.2. Typical Wiimote arrangement assumed by Wiimote Vision System Setup Toolkit 63

8.3.Principle of estimation of number of Wiimotes required for give tracking volume 64

8.4. Wiimote arrangement as suggested by Wiimote Vision System Setup Toolkit for tracking volume of 8m x 8m x 2m 66

8.5. MATLAB implementation of Dijkstra's algorithm to find camera vision system measurement error 68

8.6. Implementation of Dijkstra's algorithm to estimate system measurement error 69

8.7. Estimating resultant transformations (R,T) for intermediate vision systems 70

8.8. Automation of CAD simulation generation with Siemens NX5 Journals 73

9.1. Information flow for the Wiimote tracking system 74

9.2. The MSAT prototype frame..... 75

9.3. LED locations for several parts of the MSAT prototype 76

9.4. Motion capture and the generated CAD simulation in real-time 77

9.5. The bookshelf used for demonstration with the two stereo vision systems..... 77

9.6. The experimental setup for motion tracking for the bookshelf assembly..... 78

9.7. Locations of the LED marker for tracking different parts of the bookshelf 79

9.8. Snapshots of the bookshelf assembly simulation..... 79

LIST OF TABLES

Page

Table 4.1. Experimental conditions for the five experiments 30

Table 4.2. Average relative measurement errors for 1330 observations 30

Table 4.3. Moving average method adopted for dynamic data smoothing.....31

1. INTRODUCTION

Assembly of a product or its subsystems often involves fairly sophisticated operations and sequences performed by one or more operators with various tools and parts. It is necessary to provide effective training of the assembly process for new operators. The work of Baggett and Ehrenfeucht [1] suggests that dynamic presentation of an assembly process with a video is a superior training medium to static presentation with a text. But in the video format, crucial operations involving movements of objects could be obscured by other objects that are in the view of camera. Moreover, video information cannot be easily modified to allow investigating ways for improving existing assembly operations or planning new operations for variant assembly tasks. Thus, there is a great need for CAD model based assembly simulation.

A major problem in assembly simulation is the construction of assembly sequence including timing of part movement, which is often time-consuming, tedious, and costly [2]. One approach to address this issue is to track movements of parts, tools, and/or operators with sensors that can provide their positions and orientations in real time during an actual assembly process, and then use the obtained motion data to generate the assembly simulation with CAD models. With the use of CAD models to represent real objects in the simulation, the assembly sequence can be modified easily in the computer, thereby enabling effective and efficient investigation of alternative assembly sequences for purpose of shortening of cycle time, improvement of safety, etc. Furthermore, the simulation software can be reused for planning of variant assembly operations.

Tracking systems based on mechanical, magnetic, acoustic, inertial and optical technologies for motion capture for interactive computer graphic simulation and other

applications have been explored by researchers for many years and commercial products are continuously evolving. In particular, multi-camera systems have continued to evolve because of continuously decreasing prices of powerful computers and cameras. Among them infrared based optical motion capture systems (e.g., iotracker, PhaseSpace, Vicon, ART Gmbh, NaturalPoint) are becoming increasingly popular because of their higher precision and better flexibility. These systems are less susceptible to adverse shop floor conditions. However still most of the commercially available motion tracking systems cost in the range of tens of thousands of dollars. The lack of affordability of such has prohibited their wide applications.

Since the debut of Nintendo's Wii games, a vibrant Internet community has sprung up to take advantage of capabilities provide by the Wiimote, which includes an infrared camera inside that can detect the infrared light emitted from the IR LEDs mounted on the sensor bar and is extremely inexpensive as a result of mass production. Lee [3] created several excellent demonstrations and brought the potential of Wiimote to the attention of many researchers, such as head tracking for desktop display in virtual reality. This application used one Wiimote and two infrared LEDs, where the LEDs are fixed and apart at a known distance. It gives an impression of parallax in the display when the line between the two LEDs and the image plane of the Wiimote camera are approximately parallel. However, the use of only one camera cannot provide the six-degree-of-freedom pose of an object. Hay et al. [3] used two Wiimotes in a stereo vision to perform 3D tracking with good accuracy, but they did not investigate how to increase the tracking volume with multiple Wiimotes.

This thesis discusses the development and applications of the Wiimote based motion tracking system which costs less than $600 (including four Wiimotes, one Bluetooth card, software for the Bluetooth card, one PC, and infrared LEDs used as calibration and tracking markers).

This thesis is broadly divided into three parts. Sections 2 through 6 discuss the development of a new calibration technique which enables the position tracking of the infra-red markers with an accuracy in millimeters. Sections 7 and 8 describe the customization and configuration of the Wiimote based systems for different tracking volumes, and its integration with commercially available CAD software Siemens NX5. Section 9 describes the practical implementation of the Wiimote based tracking system for tracking different physical assemblies.

Figure 1.1 gives an overview of the Wiimote based motion tracking system and its integration with Siemens NX5 CAD software to automate the generation of CAD simulations.

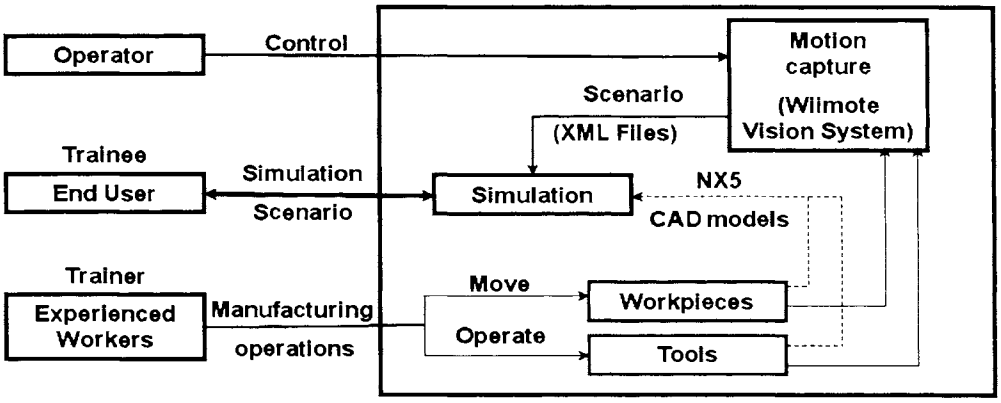


Figure 1.1 Wiimote motion tracking system architecture

1.1 CALIBRATION TECHNIQUES FOR WIIMOTES: AN OVERVIEW

Good calibration is the key to the efficient use of a multi-camera vision system. The calibration method largely depends on available resources. Kitahara et al. [4] calibrated their large-scale multi-camera system by using a classic method developed by Tsai [5]. The 3D points were collected by a combined use of a calibration board and a laser surveying instrument. The cost of the calibration hardware required for this method is much higher than the cost of the cameras. In comparison, the calibration technique discussed in the present paper requires minimum investment in the calibration hardware and thus is ideal for inexpensive IR cameras like Wii Remotes (Wiimotes) used in Nintendo Wii games.

Many researchers have successfully dealt with the problem of camera calibration by taking images from a 2D object consisting of a planar pattern [6, 7, 8]. Most of the other calibration techniques using planar objects are directly or indirectly derived from these techniques. Tsai's method of camera calibration is a classic one and is still widely used in computer vision, and there have been implementations of this method in C/C++ and other programming languages. The DLT-based calibration model presented by Heikkilla and Silven [7] uses the concepts and techniques of Melen [10] in photogrammetry, and its implementation is available as a MATLAB software module. Zhang's method [9] is a newer one and it makes use of advanced concepts in projective geometry, and the implementation of this method is also available in a MATLAB toolbox.

A comparative study of the above three methods has been carried out by Zollner and Sablatnig [11], whose experimental evaluations indicated that the overall error of the

DLT-based estimation is significantly smaller than Tsai's method in the mono-view case, but the DLT-based method generates larger errors than Zhang's method in the multi-view case.

Sections 2 through 6 of this thesis describe the development and evaluation of an integrated calibration technique in this thesis work to calibrate multiple Wiimotes together to form a low-cost motion capture system. This is a two-stage calibration technique: first the intrinsic parameters are determined with Zhang's method [9] and then the extrinsic parameters are determined with Svoboda's method [12]. Individual multi-camera systems can be further integrated using Horn's algorithm [13] to extend the range of motion capture. This integrated technique has been successfully implemented and demonstrated with good measurement accuracy with multiple Wiimotes.

1.2 6-DOF TRACKING WITH WIIMOTE BASED SYSTEM AND ITS INTEGRATION WITH CAD SOFTWARE

A Wiimote can detect up to four hotspots for a given instant. Thus, it is possible to track at least four LEDs for a given instant with a Wiimote vision system. To estimate all 6 DOF of a part (both position and orientation), a wand mounted with 4 LEDs can be used. The Wiimote vision system dynamically estimates the absolute positions of all 4 LEDs mounted on the wand. With four known markers lying in the plane of the wand, the orientation of the plane of the wand can be estimated. Thus, it is possible to estimate all 6 DOF a part with 4 markers mounted on a face of the part to be tracked.

The position and orientation data thus generated can be further dynamically passed to any commercially available CAD software like Siemens NX5 to automatically generate the CAD simulations in line with the physical part motion tracked. There exist

different ways to specify the orientation of an object in 3D space viz. quaternions, rotation matrices, etc. Typically Siemens NX5 APIs require the orientation to be specified in a standard orientation matrix form. For known positions of the four markers mounted on the part to be tracked, the orientation matrix can be calculated for any given instant.

The simulation data thus generated can be stored and played back many a times with different speeds. The CAD assembly can be viewed in different angles during the playback of the simulation data, thus allowing a better visualization of the complex assembly operations.

Sections 7 and 9 discuss in detail 6 DOF tracking with Wiimotes and its implementation for tracking different physical assemblies.

1.3 WIIMOTE VISION SYSTEM SETUP TOOLKIT

To assist the users in setting up a new Wiimote tracking system for a specified workspace, a software tool has been developed, viz. Wiimote Vision System Setup Toolkit. The code has been developed with Visual Basic language using NX Journals. The software estimates the number of Wiimotes required for given required tracking volume dimensions. To help the user visualize the shape of the actual tracking volume attained, the software also generates a CAD simulation clearly indicating the positions and orientations of the individual Wiimotes.

The toolkit assumes a particular linear arrangement of the Wiimtoes both along X and Y axis. This makes it possible to estimate the grouping of the Wiimotes to form the individual vision systems and subsequently estimate the maximum possible position measurement error that can incur in the system. With the Wiimote Navigation Toolbar

developed inside NX, it is possible to control the visibility, position and orientation of all individual Wiimotes inside CAD simulation. Based on the user modified positions of the individual Wiimotes, the system re-estimates the actual tracking volume achieved for the current Wiimote layout.

Dijkastra's algorithm [14] has been implemented to estimate the resultant transformation matrices for individual vision systems inside the entire Wiimote setup. The algorithm also lets the system estimate the measurement error incurred at any given point inside the tracking volume. Thus, the measurement errors can be easily estimated even for complex tracking volumes. Section 8 discusses different case studies for the Wiimote Vision System Setup Toolkit.

2. PRINCIPLE OF STEREO VISION AND DISTRIBUTED VISION SYSTEM

2.1 STEREO VISION PRINCIPLE

A Wiimote camera contains a CMOS sensor with 128×96 pixels, but it can give a 1024×768 pixel resolution after sub-pixel image processing. With an IR filter, Wiimote can detect up to four IR hotspots and transmit their pixel coordinates via the wireless Bluetooth at a rate of 120 Hz. A Wiimote camera has a measurement range of 7 meters. It is very inexpensive and easily portable.

Two Wiimotes can be set up as a stereo vision to track the motion of infrared LEDs attached to the parts, tools and operators in an assembly. The basic principle of stereo vision is indicated in Figure 2.1. Assume the simplified configuration of two parallel cameras with identical intrinsic parameters as shown in Figure 2.1. Assume the straight line connecting the two optical centers of the two cameras being coincident with the x_c-axis.

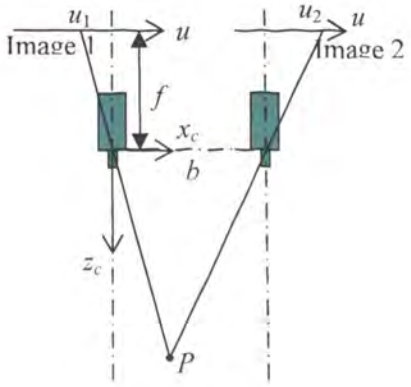


Figure 2.1. Principle of the stereo vision

In Figure 2.1, point P has the coordinates (X_C, Y_C, Z_C) and projects to both camera sensors. The pair of image points u_1 and u_2 of point P are referred to as conjugate points. The difference between the image locations of the two conjugate points is known as the disparity, $d=u_1-u_2$. The Z_C coordinate, which is the distance of point P from the stereo vision system, can be calculated as

$$Z_C = \frac{f.b}{d} \tag{1}$$

where f is the focal length of the Wiimote camera, and b is the basis length (distance between the two cameras). Thus the position of the point can be calculated from the image data from the two cameras using the triangulation principle. Note that it is necessary to determine the intrinsic parameters of both cameras and the transformation between the two coordinate frames. This requires calibration of the system.

2.2 COVERAGE OF A SINGLE STEREO VISION SYSTEM

An important parameter in the stereo system is the tracking volume which is the overlapping region of the two Wiimotes' coverage. As shown in Figure 2.1, two Wiimotes are placed such that their centers are connected by a horizontal line. The overlapping area at the distance of H from this line is represented by the length L and width W. The length L can be calculated as

$$L = 2H \tan\frac{\theta}{2} - b \tag{2}$$

For the Wiimote, $\theta = 41^\circ$ is the horizontal view angle, b=10 cm is the assumed basis length, and H=7 m is the height. Note that 7 m is the maximum distance that the Wiimote can sense in a direction perpendicular to the CCD chip. Substituting these

parameter values in Eq. (2), the horizontal overlapping length L between the two cameras can be determined as 5.1 m (at the distance of 7 m).

From Figure 2.2, the coverage width is

$$W = 2H\tan\frac{\alpha}{2} \tag{3}$$

where $\alpha = 30^{\circ}$ and H=7m. Substituting these values into Eq. (3), the coverage width can be found as 3.75 m. Therefore, for a single stereo system, the tracking area is 5.1m ×3.75m at the height of 7 m.

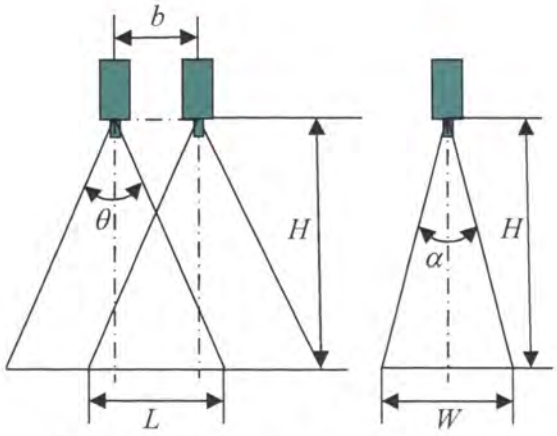
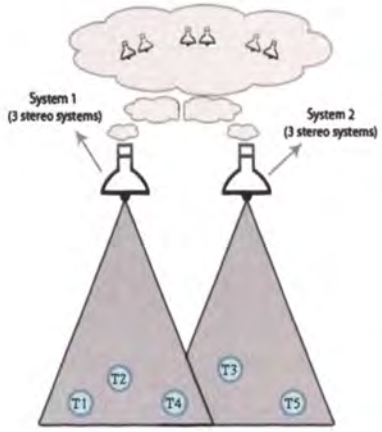


Figure 2.2. Stereo system coverage

2.3 DISTRIBUTED VISION SYSTEM

Because of the limited coverage of a single stereo system, multiple stereo vision systems can be integrated into one system so as to track a wide variety of assemblies on a typical assembly shop floor. Figure 2.3 illustrates the working principle of a distributed stereo system. As illustrated in Figure 2.3, system 1 and system 2 represent two computer systems each with a set of vision systems, with system 1 designated as the "Master". Consider the position of a LED at time instants T1 and T2 as shown in the Figure 2.3. The marker at these positions can be seen only by system 1, and thus its position

information is provided by system 1. For LED positions at T3 and T5, the marker can be seen only by system 2, and thus its position is provided by system 2. At T4, the marker lies in the overlapped region of system 1 and system 2 and can be tracked by both stereo systems. In this case, the master system (system 1) uses the data obtained by itself and the data it receives from the other system (system 2) to calculate the average, thus providing a smooth data transition.



System 1

Time	Position
10	(x11, y11, z11)
20	(x12, y12, z12)
30	-
40	(x14, y14, z14)
50	-

System 2

Time	Position
10	-
20	-
30	(x23, y23, z23)
40	(x24, y24, z24)
50	(x25, y25, z25)

Final Data

Time	Position
10	(x11, y11, z11)
20	(x12, y12, z12)
30	(x23, y23, z23)
40	((x14+x24)/2, (y14+y24)/2, (z14+z24)/2)
50	(x25, y25, z25)

Figure 2.3. Distributed vision system

Each Bluetooth adapter is capable of supporting a maximum of 7 devices in a Bluetooth stack, which implements the Bluetooth protocol to communicate with the PC. More than one Bluetooth stack can be deployed to increase the number of devices (Wiimotes) to 14, 21, etc. Given the size of a typical shop floor, it is necessary to increase the tracking volume in many applications. A distributed motion tracking system can be developed where multiple computers (each with a set of Wiimote vision systems) communicate with each other to form a distributed sensing network. The application is running with one of the systems taking up the role of the "Master". Each system sends

out the tracking data it receives to the "Master", which is responsible for the centralized storage of all the captured motion data and also provides the positional information of the part being tracked to a simulation system.

3. CALIBRATION OF A CAMERA AND VISION SYSTEM

The Wiimote provides information in terms of the 2D image coordinates (u, v) for every image impregnated on its CCD chip. A mathematical model can be developed to link the 2D data from two Wiimotes for an IR source in 3D space, so as to retrieve the 3D coordinate data of the marker. This requires determining the focal length and the principal point of each Wiimote. It is also useful to build a distortion model to allow image correction and accuracy improvement. Furthermore, the positions and orientations of the Wiimotes relative to each other need to be determined. Although there exist multiple-camera calibration methods that can be used to determine the intrinsic and extrinsic parameters simultaneously [7, 8, 9] the two problems were treated separately in this study because a camera's intrinsic parameters are determined far less frequently (ideally, only once in a camera's lifetime) than its extrinsic (position and orientation) parameters.

3.1 CAMERA MODEL

In the camera calibration, the transformation between 3D world coordinates and 2D image coordinates is determined by solving the unknown parameters of the camera model. Here the perspective projection (i.e., pinhole) camera model as illustrated in Figure 3.1 has been used. The center of projection is at the origin O of the camera coordinate system. The image coordinate system is parallel to the camera coordinate system, with a distance f (focal length) from O along the z_c axis. The z_c axis is the optical axis, or the principal axis. The intersection between the image plane and the optical axis is the principal point o. The u and v axes of the image plane coordinate system are

parallel to the x and y axes, respectively. The coordinates of the principal point in the image plane coordinate system are (u_0, v_0).

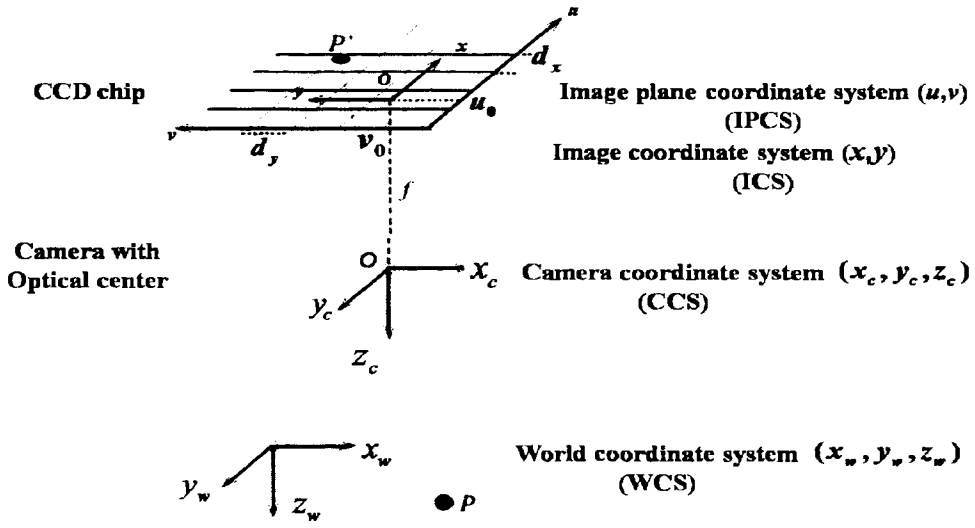


Figure 3.1. Pinhole camera model

As shown in Figure 3.1, let P be an arbitrary point located on the positive side of the z_c axis and p be its projection on the image plane. The coordinates of P in the camera coordinate system are (x_c, y_c, z_c) and in the world coordinates system is (X, Y, Z). The coordinates of p in the image plane coordinate system are (u, v), which are related to (X, Y, Z) by the following equation:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

Where

$$A = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the above equation, R and t are the rotation matrix and translation vector which relate the world coordinate system to the camera coordinate system, and A is the intrinsic parameter matrix. The parameter s represents the skewness of the image in terms of the two image axes, $\alpha_x=f/d_x$ and $\alpha_y=f/d_y$ are scaling factors in the image u and v axes, respectively, f is camera focal length, and d_x and d_y are the pixel dimensions in the x and y directions, respectively. The parameter λ is a scale factor.

For the stereo vision system, the relationship between the two camera coordinate systems also needs to be calibrated. For a poin t P in 3D space, its two coordinates P_L and P_R in the left and right camera coordinate systems have the following relationship:

$$P_R = R_s * P_L + T_s \quad (5)$$

where R_s is the rotation matrix and T_s is the translation vector between the two coordinate frames of the stereo system.

3.2 SINGLE STEREO VISION SYSTEM CALIBRATION

To calibrate a Wiimote camera, a calibration plate can be used. A plate that could hold 36 LEDs as shown in Figure 3.2 has been used. In the calibration process, the calibration plate was moved to different locations in the field of view of the Wiimote. Figure 3.3 indicates a pictorial representation of the 36 LEDs at two different locations as seen by a Wiimote in the camera coordinate system. The pixel coordinates of each LED at each fixed location were obtained by the Wiimote. Together with the known world coordinates of each LED's position, the intrinsic parameters of the Wiimote were

calculated using the Camera Calibration Toolbox of MATLAB, which was developed by Bouguet [15] based on a calibration algorithm developed by Zhang [9].

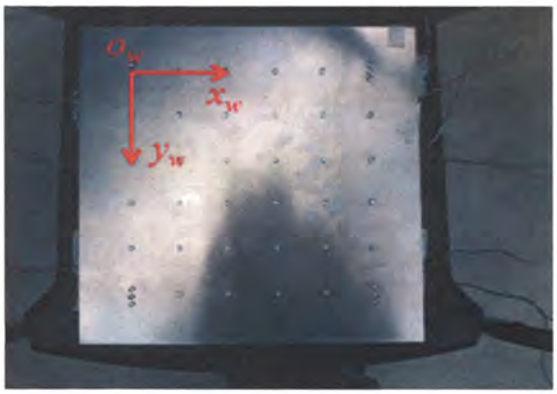


Figure 3.2. The 36 LED calibration plate

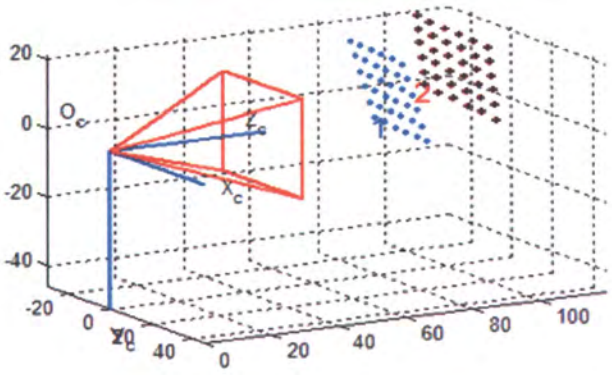


Figure 3.3. Data collected from the calibration plate by a Wiimote

Without loss of generality, it can be assumed that the 36 LEDs are on the XY-plane (Z = 0) of the world coordinate system, which has the origin O_w at the upper left corner of the calibration plate shown in Figure 3.2. Let r_i denote the i^{th} column of the rotation matrix R_s. From Eq. (4),

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Let P be a point on the calibration plate with coordinates (X, Y, 0), and its image p have coordinates (u, v). Also, let $\tilde{P} = [X\ Y\ 1]^T$ and $\tilde{p} = [u\ v\ 1]^T$, then the point P and its image point p are related by a homography H:

$$\lambda \tilde{p} = \mathrm{H}\tilde{\mathrm{P}}, \qquad \mathrm{H} = \mathrm{A}\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{6}$$

Given the image of a model plane representing the calibration plate, a homography can be estimated based on the maximum likelihood criterion. Let P_i and p_i be the model and image points, respectively. Ideally, together they should satisfy Eq. (6). However, In reality, they don't because of noise in the extracted image points. The maximum likelihood estimation of H is obtained by minimizing the following functional

$$\sum \| p_i - p_i \|^2 \tag{7}$$

The homography H will be more accurate if there are more points on one image.

Let's denote the homography by $H=[h_1\ h_2\ h_3]$. From Eq. (6),

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \rho \mathrm{A}\begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

where ρ is an arbitrary scalar. As r_1 and r_2 are orthonormal,

$$h_1^T A^{-T} A^{-1} h_2 = 0 \tag{8}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \tag{9}$$

Let B be

$$\mathrm{B}=\mathrm{A}^{-T}\mathrm{A}^{-1}=\begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}=\begin{bmatrix} \frac{1}{\alpha_x^2} & -\frac{s}{\alpha_x^2\alpha_y} & \frac{sv_0-u_0\alpha_y}{\alpha_x^2\alpha_y} \\ -\frac{s}{\alpha_x^2\alpha_y} & \frac{s^2}{\alpha_x^2\alpha_y^2}+\frac{1}{\alpha_y^2} & -\frac{s(sv_0-u_0\alpha_y)}{\alpha_x^2\alpha_y^2}-\frac{v_0}{\alpha_y^2} \\ \frac{sv_0-u_0\alpha_y}{\alpha_x^2\alpha_y} & -\frac{s(sv_0-u_0\alpha_y)}{\alpha_x^2\alpha_y^2}-\frac{v_0}{\alpha_y^2} & \frac{(sv_0-u_0\alpha_y)^2}{\alpha_x^2\alpha_y^2}+\frac{v_0^2}{\alpha_y^2}+1 \end{bmatrix}$$

and b be defined by a vector

$$\mathrm{b}=\left[B_{11},B_{12},B_{22},B_{13},B_{23},B_{33}\right]^T \tag{10}$$

Also, let the i^{th} column vector of H be $\mathrm{h}_i = [\mathrm{h}_{i1}, \mathrm{h}_{i2}, \mathrm{h}_{i3}]^T$. Thus

$$\mathrm{h}_i^T\,\mathrm{B}\mathrm{h}_j=\mathrm{v}_{ij}^T\mathrm{b} \tag{11}$$

where $\mathrm{v}_{ij}=\left[h_{i1}h_{j1},h_{i1}h_{j2}+h_{i2}h_{j1},h_{i2}h_{j2},h_{i3}h_{j1}+h_{i1}h_{j3},h_{i3}h_{j2}+h_{i2}h_{j3},h_{i3}h_{j3}\right]^T$.

Therefore, the two fundamental constraints (8) and (9) can be rewritten as two homogeneous equations in b:

$$\begin{bmatrix} \mathrm{v}_{12}^T \\ (\mathrm{v}_{12}-\mathrm{v}_{22})^T \end{bmatrix} b = 0 \tag{12}$$

If n images of the model plane are observed, stacking n such equations results in

$$\mathrm{Vb}=0 \tag{13}$$

where V is a 2n×6 matrix. Once b is estimated, the camera intrinsic parameters in the matrix A can be computed. The matrix B is estimated using $\mathrm{B}=\lambda\mathrm{A}^{-T}\mathrm{A}^{-1}$, where λ is a scale factor. The intrinsic parameters can be extracted from matrix B as given in Eq. (14)

$$
\begin{aligned}
\lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\
\alpha_x &= \sqrt{\lambda / B_{11}} \\
\alpha_y &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\
s &= -B_{12}\alpha_x^2 \alpha_y / \lambda \\
u_0 &= sv_0 / \alpha_x - B_{13}\alpha_x^2 / \lambda \\
v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2)
\end{aligned}
\tag{14}
$$

Once the intrinsic parameters in matrix A have been obtained, the extrinsic parameters for each image can be computed from Eq. (6) as follows:

$$
r_1 = \lambda A^{-1}h_1, \quad r_2 = \lambda A^{-1}h_2, \quad r_3 = r_1 \times r_2, \quad t = \lambda A^{-1}h_3 \tag{15}
$$

with $\lambda = 1 / \left\| A^{-1}h_1 \right\| = 1 / \left\| A^{-1}h_2 \right\|$.

It is assumed that the two Wiimotes (left and right) have been calibrated with the calibration plate with the same points on the calibration plate at the same 3D locations. Therefore, the same number of images is used to calibrate the two cameras individually. Figure 3.4 shows a pictorial representation of 36 LEDs at two different locations as seen by the two Wiimote cameras. Then the two camera's calibration results are used to calibrate the stereo system. There exists a relationship between the world coordinate system and the camera coordinate system through the extrinsic parameters, i.e., the rotation matrix R and translation vector T, as follows:

$$
P_{CL} = R_1 * P_1 + T_1, \qquad P_{CR} = R_2 * P_2 + T_2 \tag{16}
$$

where P_{CL} and P_{CR} represent the 3D coordinates of these two points in the left camera frame and in the right camera frame, respectively, P_1 and P_2 are two points expressed in

the world coordinate system, R_1 and T_1 are the extrinsic calibration results for the left camera, and R_2 and T_2 are the extrinsic calibration results for the right camera.

During the stereo system calibration, the same points with the same world coordinates are used by both Wiimotes, so P_1=P_2. Then from Eq. (16) the following result can be obtained

$$P_{CR} = R_2 * R_1^{-1} * P_{CL} + (T_2 - R_2 R_1^{-1} T_1)$$

That is

$$P_{CR} = R_s P_{CL} + T_s$$

where R_S is the rotation matrix and T_S is the translation vector between the two Wiimote cameras coordinate systems.

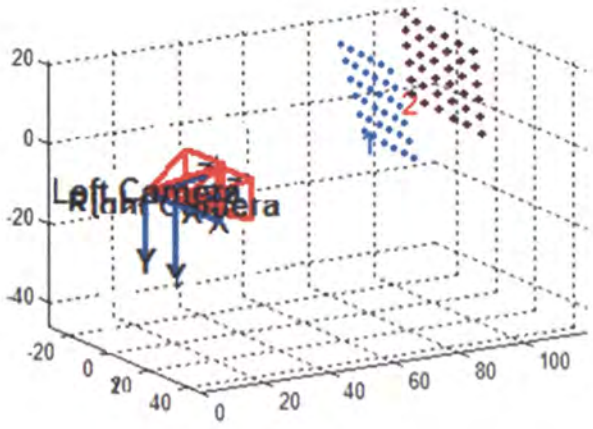


Figure 3.4. Data collected by a single Wiimote stereo system

3.3 CALIBRATION OF A MULTIPLE STEREO VISION SYSTEM

When more than one camera vision systems are used, each vision system generates 3D coordinates with respect to its own coordinate system. In order to integrate the multiple vision systems, it is necessary to determine the relative position and

orientation between the various camera vision systems so as to establish a common world coordinate system.

The photogrammetric problem of recovering the transformation between the two systems from these measurements has been addressed by several researchers [16, 17, 9]. The classical approach of finding the transformation parameters is to minimize the sum of squares of errors numerically. However, Horn [13] derived a closed-form solution to the least-square problem by use of unit Quaternions to represent rotation. Hereby this algorithm developed by Horn [13] has been implemented.

In order to integrate multiple stereo vision systems, it is necessary to determine the relative position and orientation between the various stereo vision systems. Consider a set of 3D points measured by two stereo systems, each having its local coordinate frame, as shown in Figure 3.5. These points represent the locations of a marker positioned randomly in 3D space with the requirement that they can be seen by both stereo systems at each time the data is recorded.

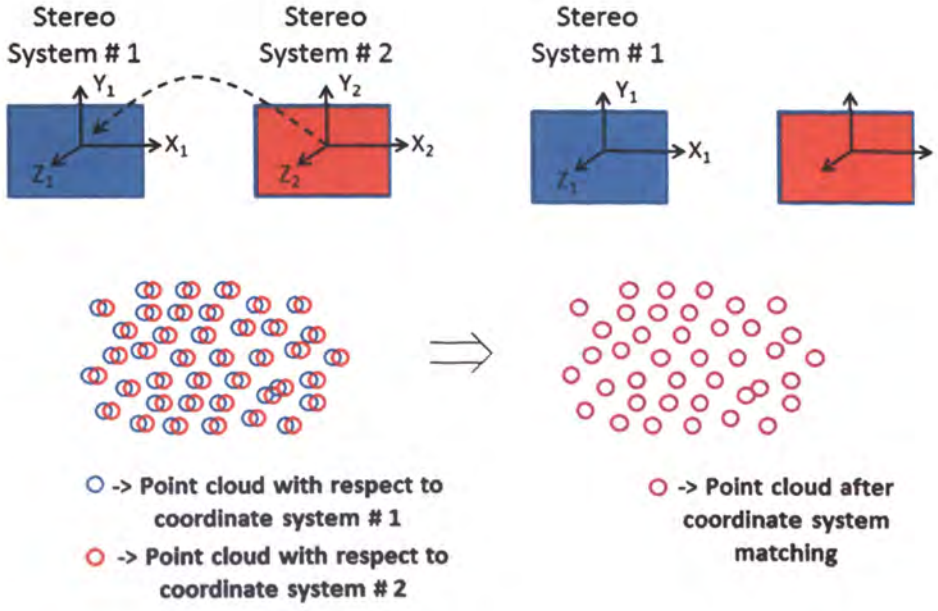


Figure 3.5 Calibration of an integrated system of two stereo vision system

In general, the transformation between two cartesian coordinate systems can be split into a rotation, a translation and a scaling factor (if applicable). Considering this particular application, the individual Wiimote stereo systems are calibrated independently. Thus, here, it is also necessary to account for the scaling factor between different stereo systems. Numerous experiments with multiple Wiimote systems have indicated that the scaling factor is almost unity in most of the cases. However it is always recommended to consider the scaling factor as one of the unknowns. In general, the transformation between two coordinate systems can be split into a rotation, a translation and a scaling factor. Consider three points in 3D space as show in Figure 3.6 and 3.7. Let the coordinates of the three points in each of the two coordinate systems be $r_{l,1}$ $r_{l,2}$, $r_{l,3}$ and $r_{r,1}$ $r_{r,2}$, $r_{r,3}$, respectively.

Let x_l be $x_l = r_{l,2} - r_{l,1}$

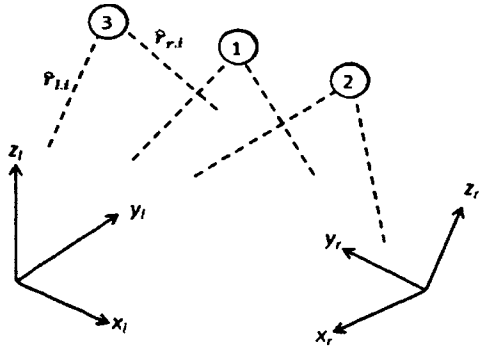


Figure 3.6 Coordinates of a number of points measured in two different coordinate systems

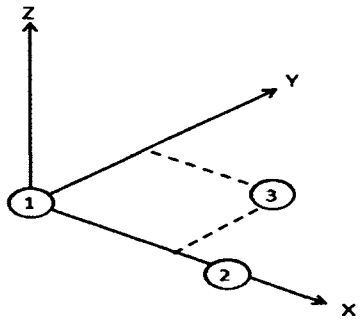


Figure 3.7 Three points defining a triad

The unit vector along x-axis in the left coordinate system can be obtained as

$$\hat{x}_l = \frac{x_l}{||x_l||} \tag{17}$$

Also let y_l be a component of $(r_{l,3} - r_{l,1})$ perpendicular to $\hat{x}_l$,

$$y_l = (r_{l,3} - r_{l,1}) - [(r_{l,3} - r_{l,1}).\hat{x}_l]\,\hat{x}_l .$$

The unit vector along y-axis in the left coordinate system can be obtained as

$$\hat{y}_l = \frac{y_l}{||y_l||} \tag{18}$$

To complete the triad, take the cross product to define the z-axis as

$$\hat{z}_l = \hat{x}_l \times \hat{y}_l$$

The same construction is then repeated in the right-hand coordinate system to obtain, $\hat{x}_r$, $\hat{y}_r$ and $\hat{z}_r$. Thus, hereby one needs to estimate the rotation that takes $\hat{x}_l$ to $\hat{x}_r$, $\hat{y}_l$ to $\hat{y}_r$, and $\hat{z}_l$ to $\hat{z}_r$. The column vectors can be used to form the matrices M_l and M_r as follows:

$$M_l = [\,\hat{x}_l \;\; \hat{y}_l \;\; \hat{z}_l\,], \quad M_r = [\hat{x}_r \;\; \hat{y}_r \;\; \hat{z}_r\,]$$

Given a vector r_l in the left coordinate system, $(M_l^T r_l)$ gives the components of the vector r_l along the axes of the constructed triad. Multiplication by M_r then maps these into the right-hand coordinate system as

$$r_r = M_r M_l^T r_l$$

Thus, the sought after rotation can be given as

$$R = M_r M_l^T \tag{19}$$

The above constitutes a closed-form solution for the rotation.

To calculate the scaling factor between the two coordinate systems, the centroids $\bar{r}_l$ and $\bar{r}_r$ of the two sets of points can be calculated in the left and right coordinate systems, respectively, as follows:

$$\bar{r}_l = \frac{1}{n}\sum_{i=1}^{n}(r_{l,i}) \quad , \quad \bar{r}_r = \frac{1}{n}\sum_{i=1}^{n}(r_{r,i}) \tag{20}$$

Subtract the centroids from all measured data to get the resultant coordinates as

$$r'_{l,i} = r_{l,i} - \bar{r}_l \quad \text{and} \quad r'_{r,i} = r_{r,i} - \bar{r}_r \tag{21}$$

Now the scaling factor can be computed as

$$s = \sum_{i=1}^{n}(r'_{r,i} * R(r'_{l,i})) \Big/ \sum_{i=1}^{n} \|r'_{l,i}\|^2 \tag{22}$$

To find the translation factor, let there be n points (representing n different positions of the LED in 3D space) where the calibration data is collected. The measured coordinates in the left and right coordinate systems can be denoted by $\{r_l,i\}$ and $\{r_r,i\}$ respectively, where i ranges from 1 to n. Here, one needs to estimate the transformation from the left to the right coordinate system of the form

$$r_r = sR\,(r_l) + T_o$$

Here s is a scale factor, T_o is the translation, and $R(r_l)$ denotes the rotated vector r_l. The translation T_0 can be computed as follows:

$$T_0 = \bar{r}_i - s * R(\bar{r}_i) \tag{23}$$

It is practically impossible to find a transformation that maps the measured coordinates of a set of points in one coordinate system exactly into the measured coordinates of these points in the other coordinate system. The residual error can be written as

$$e_i = r_{r,i} - sR\,(r_{r,i}) - T_o$$

The root mean square (RMS) of these errors can be determined as

$$e_{RMS} = \sqrt{\sum_{i=1}^{n} ||e_i||^2} \tag{24}$$

The above solution routine has been coded into a software routine by Wengert and Bianchi [18]. This routine has been used hereby to calculate the transformation parameters in calibrating two Wiimote stereo systems with respect to each other.

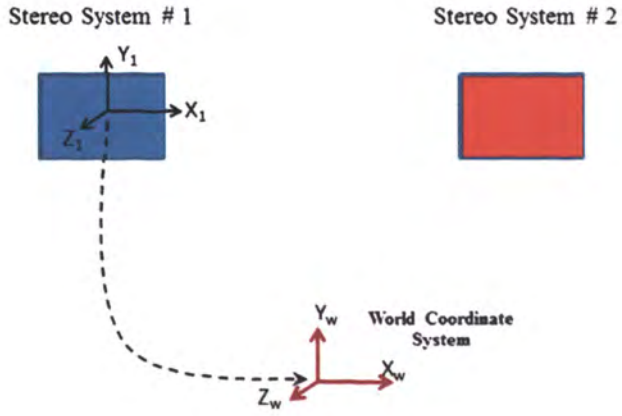
Experiments with more than 1000 measured points distributed in the overlapped region between two Wiimote stereo systems have shown that a good estimation of the transformation parameters can be provided using the above technique for two Wiimote stereo systems.

3.4 DEFINING THE WORLD COORDINATE SYSTEM

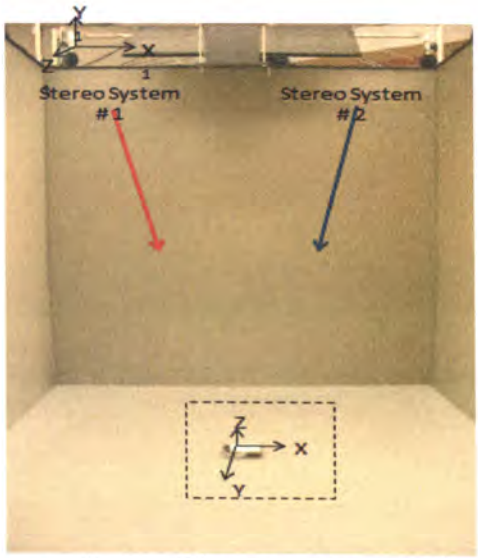
From the application point of view, it is convenient to locate the world coordinate system at a known position. This allows the user to define a convenient point of origin before making measurements in 3D space. The procedure involves positioning an L-shaped wand at an appropriate position within the measurement volume as shown in Figure 3.8. The system calculates both the rotation and transformation parameters that relate the local coordinate system of each of the stereo systems to the world coordinate system defined by the wand. The transformation between the local coordinate system and the world coordinate system is given as

$$r_w = R^{-1}\left[(r_1) - T\right] \tag{25}$$

where R is the Rotation, and T is the Translation.



(a) Coordinate transformation between two stereo vision systems



(b) A wand placed in a workspace where tracking is done

Figure 3.8 Defining the world coordinate system

4. PERFORMANCE EVALUATION OF STEREO VISION SYSTEMS

Based on the principles and techniques discussed above, vision systems have been developed with Wiimotes for 3D motion tracking and assessed their measurement accuracy. The following discusses the experiments with these systems.

4.1 EXPERIMENT WITH A SINGLE STEREO VISION SYSTEM

Figure 4.1 indicates the measurement accuracy of a system consisting of two Wiimotes using Zhang's algorithm. Two infrared LEDs are placed 19.5 cm apart on a wand. The wand is moved randomly in 3D space at the distance of 2-2.5 m from the two Wiimotes and the distance between the two markers is calculated from the obtained image data for many locations of the wand. Absolute position of each LED is measured independently and the distance between the two markers is further estimated for these different positions of the wand. The RMS distance error observed was 2.7 mm. This result is comparable to that obtained by a much more expensive system [16]. Figure 4.1 shows the measurement accuracy for a typical single stereo system.

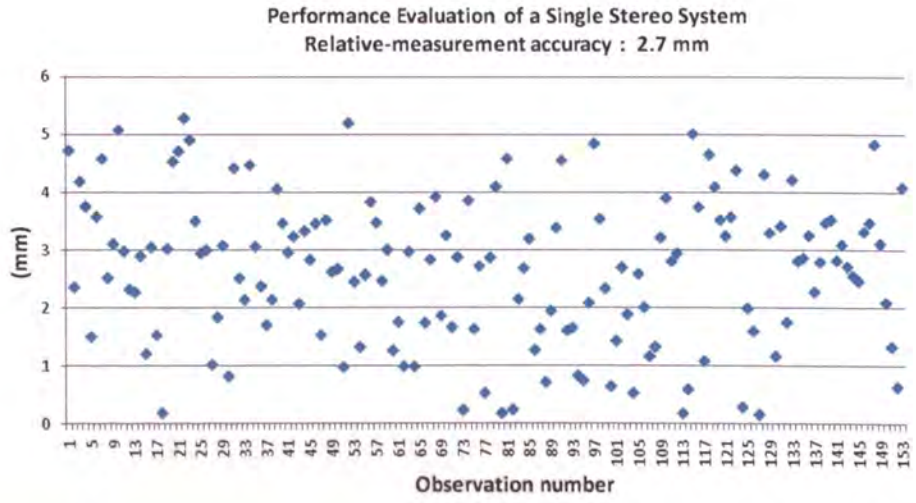
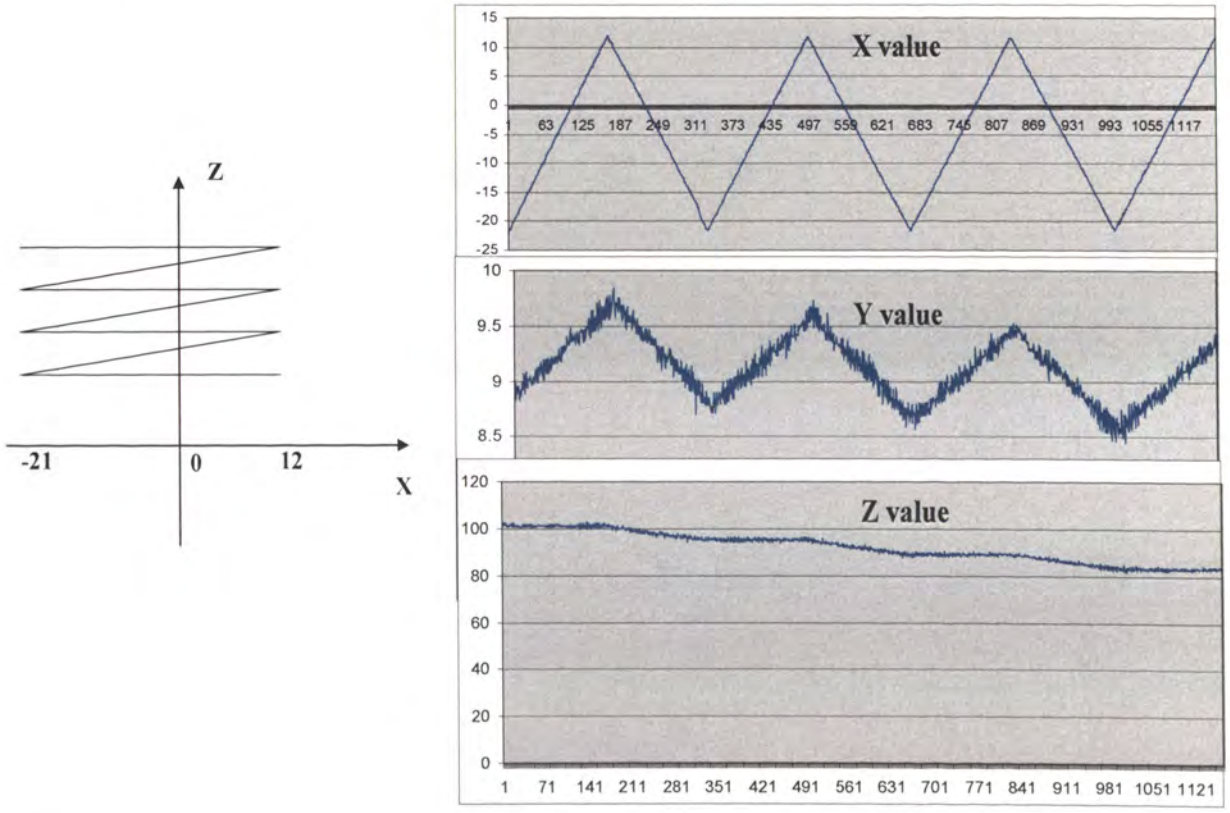


Figure 4.1 Measurement accuracy with a single Wiimote stereo vision system

An experiment was carried out with the Wiimote stereo vision system for motion tracking was done with the help of a CNC machine used to generate a prescribed trajectory. An LED was mounted on a Bridgeport CNC machine (X-travel 14', Y-travel 12'), and measurements were taken for travels of the LED marker in the X, Y and Z directions. NC programs were written for the CNC machine for tracing different profiles on a 2D plane as shown in Figure 4.2(a), so as to test the continuous motion capture capability by the Wiimote stereo system. Figure 4.2(b) shows the experimental results obtained. Accuracy was tested for the total known travel of the CNC tool for a given path. Measurements were taken at a distance of 1.5m from the Wiimotes. The repeated measurement accuracy as noted in this case was 0.5mm against a constant tool travel of 250 mm.



(a) Programmed motion (b) Motion as tracked by the system

Figure 4.2 Wiimote motion tracking tested with a CNC machine

4.2 EXPERIMENT WITH AN INTEGRATED TWO STEREO VISION SYSTEM

Experiments were carried out to determine the measurement accuracy with an integrated two stereo vision system. A marker was exposed to two Wiimote stereo vision systems such that it can be seen by both systems at the same position. Let (x_1,y_1,z_1) and (x_2,y_2,z_2) be the coordinates obtained from the first and second stereo vision systems, respectively, for the same marker position in 3D space. The absolute error at any point P can be calculated as

$$\text{Error, } e = \sqrt{(x_i - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{26}$$

Figure 4.3 indicates the measurement errors under five different experimental conditions. The number of observations taken was 1330 under each experimental condition. However, for the ease in visualization, only first 550 observations for each experiment are indicated in Figure 4.3. Table 4.1 summarizes the experimental conditions for the experiments, and Table 4.2 summarizes the average measurement errors obtained in these experiments.

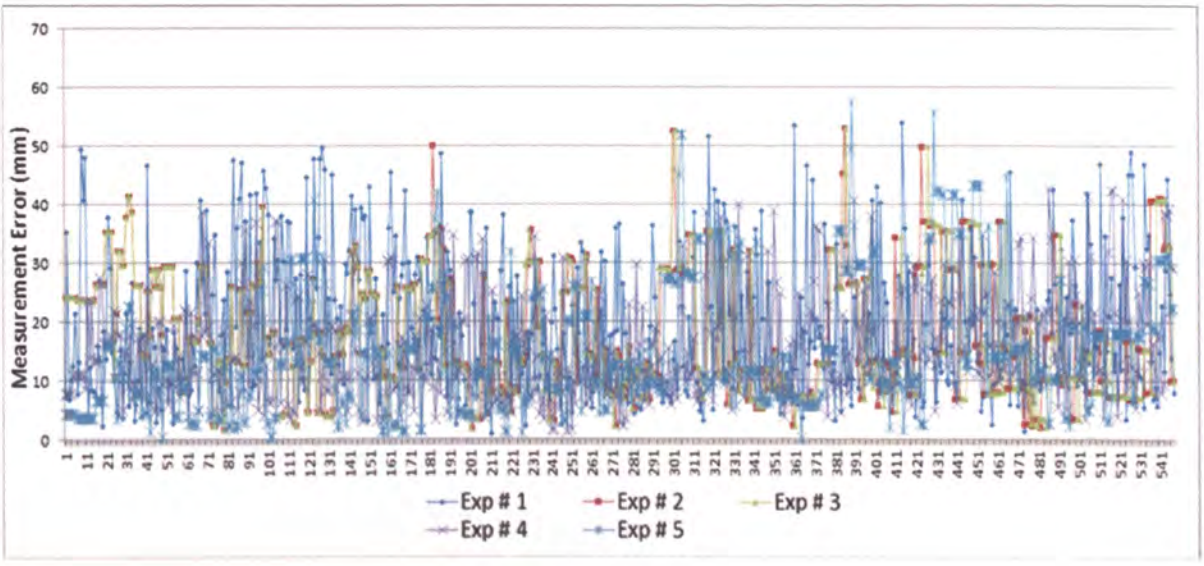


Figure 4.3. Measurement errors in the five experiments (refer Table 4.1)

Table 4.1 Measurement conditions for the five experiments

	Distance from stereo system in Z-direction	Movement of the marker	Location of origin in 3D space	Number of observations
Experiment # 1	1.6 m	Parallel to X-axis	Near the right edge of the total tracked area	1330
Experiment # 2	1.6 m	Parallel to Z-axis	Near the right edge of the total tracked area	1330
Experiment # 3	1.4 m	Parallel to Z-axis	Near the left edge of the total tracked area	1330
Experiment # 4	2.5 m	Parallel to X-axis	Approximately at the center of the total tracked area	1330
Experiment # 5	2.5 m	Parallel to Z-axis	Approximately at the center of the total tracked area	1330

Table 4.2 Average relative measurement errors for 1330 observations

	Experiment Number				
	1	2	3	4	5
Average measurement error (mm)	18.6	17.9	17.9	16.5	15.9

4.3 DATA SMOOTHING

Because the measured data fluctuates during the motion capture, different techniques including moving average, B-spline curve, and Bezier curve methods have been considered for smoothing the captured data. With consideration of both the accuracy

level sought and the need for real-time animation in the assembly simulation application, it was decided to implement the moving average method. The implementation of this method in the computer code revealed a considerable improvement in the dynamic simulation. Table 4.3 depicts the five-point moving average method adopted. Figure 4.4 displays the data before and after mathematical smoothing. It can be seen clearly that the data obtained after the moving average is substantially smoother than the original data.

Table 4.3 Moving average method adopted for dynamic data smoothing

original points	filtered points
a1	a1
a2	(a1+a2+a3)/3
a3	(a1+a2+a3+a4+a5)/5
a4	(a2+a3+a4+a5+a6)/5
a5	(a3+a4+a5+a6+a7)/5
a6	(a4+a5+a6+a7+a8)/5
a7	(a5+a6+a7+a8+a9)/5
a8	(a6+a7+a8+a9+a10)/5
a9	(a8+a9+a10)/3
a10	a10

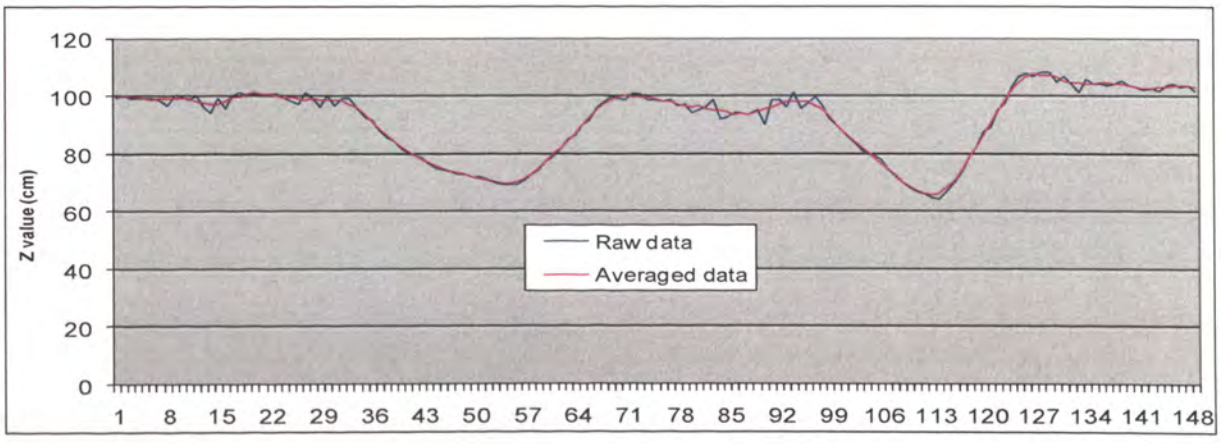


Figure 4.4 Data before and after smoothing

4.4 ERROR ANALYSIS AND MEASUREMENT

According to the stereo vision principle described in Section 2, let the coordinates of point P in 3D space in the camera frame be (X_C, Y_C, Z_C) and the image plane coordinates of the point's projection be (u, v). Thus from Eq. (1),

$$Z_C = \frac{f \cdot b}{d} = \frac{\alpha . b}{u_1 - u_2} = \frac{\alpha . b}{d}$$

where α is the focal length in pixel unit, d is disparity between the projections of point P on the two image planes in pixel unit.

Assuming that the two cameras have the same intrinsic parameters and are parallel to each other, α = 1360. For b=10 cm and b=20 cm, Z_C and ΔZ_C have the following relations:

$$\begin{aligned} b = 10\ cm: \quad & Z_C = \frac{13600}{d} && \Delta Z_C = \frac{13600}{d^2}\Delta d \\ b = 20\ cm: \quad & Z_C = \frac{27200}{d} && \Delta Z_C = \frac{27200}{d^2}\Delta d \end{aligned} \tag{27}$$

From above equation it can be seen that Z_C is inversely proportional to d, and ΔZ_C is inversely proportional to the square of d. Consider Z_C changing from 100 cm to 700 cm. When b = 10 cm, d will change from 136 pixels to 19.4 pixels, and ΔZ_C will be from 0.7 cm to 36 cm, setting Δd as 1 pixel. When b = 20 cm, d will change from 272 pixels to 38.8 pixels, and ΔZ_C will be from 0.36 cm to 18 cm. The calculated tracking errors at b = 10 cm and b = 20 cm are shown in Figure 4.5. Apparently, ΔZ_C with b=20cm is 50% of ΔZ_C with b=10cm. Therefore, to increase the measurement accuracy, the measured object can be set closer to the Wiimote cameras and the two cameras can be set farther away

from each other. However, this will reduce the tracking volume, thus one should consider the tradeoff between measurement accuracy and tracking volume.

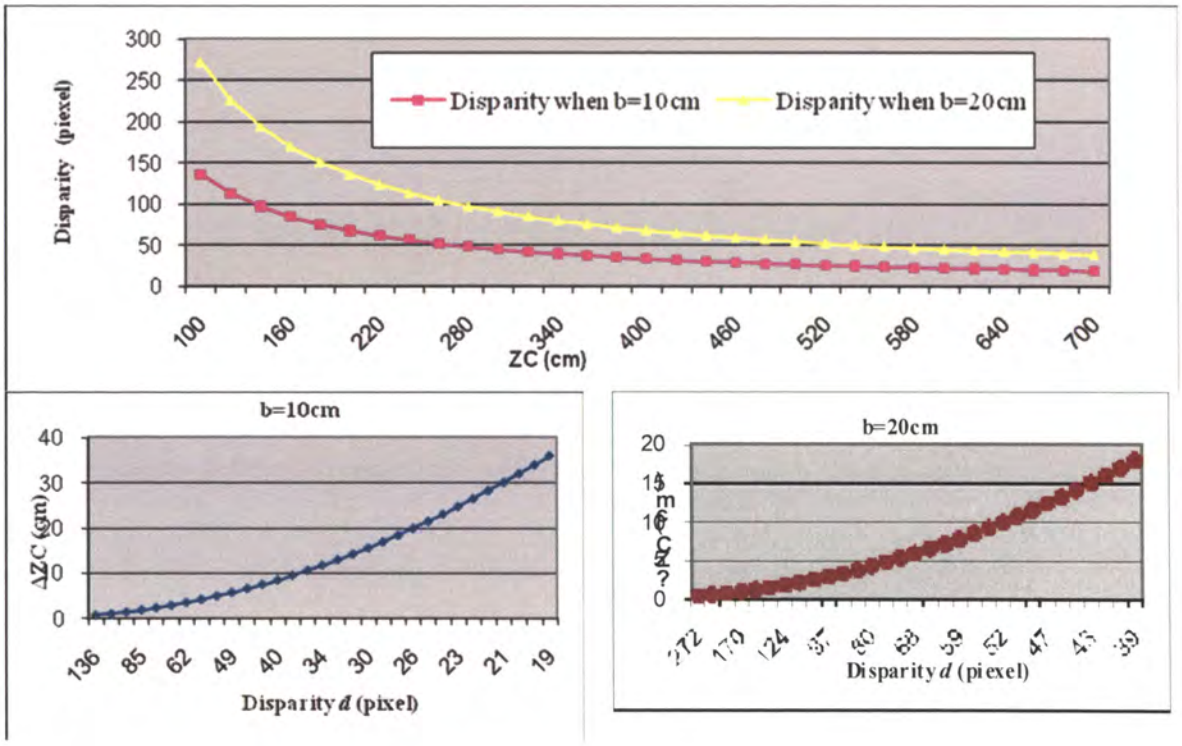


Figure 4.5 Motion tracking error from theoretical analysis

To assess the analysis result, the theoretic and actual relationships between Z_c and disparity d are compared as shown in Figure 4.6. As it can be seen from Figure 4.6, the two relationships agree well each other and trends in the relationship are exactly the same.

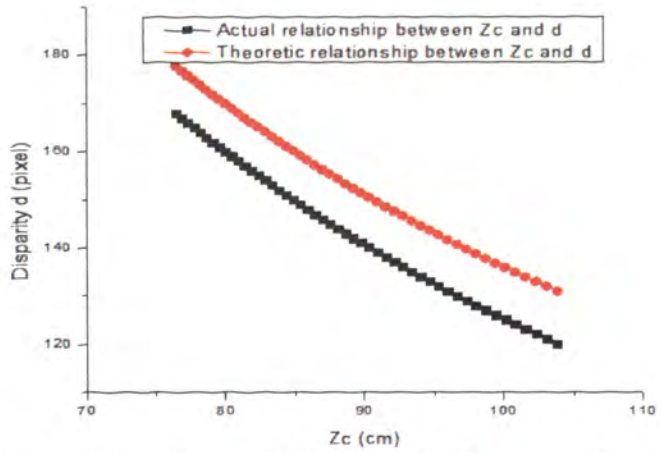


Figure 4.6. Theoretic and actual relationships between Z_c and disparity d

From Eq. (27) $d = \frac{13600}{Zc}$ and $d^2 = \frac{13600}{\Delta Zc}\Delta d$.

Simplification gives $\Delta Z_c = \frac{Zc^2}{13600}\Delta d$.

This implies that $Z_c^2 \alpha \Delta Z_c$.

Thus as the tracking object moves away from the Wiimotes along Zc, the error incurred in the position measurement increases in proportion of the square of Zc.

Figure 4.7 indicates the experimental data collected for a stereo vision system with 2 Wiimotes. Measurement errors were estimated with a wand mounted with 2 LEDs placed 153 mm apart. The measurement errors were recorded in the plane X_c-Z_c (see Figure 2.1) for different locations of the IR marker. The numbers associated with the point cloud as given in the Figure 4.7, indicate the measurement error noted at each concerned marker location. It can be seen clearly that the measurement error increases as the marker is moved away from the Wiimotes along Z_c.

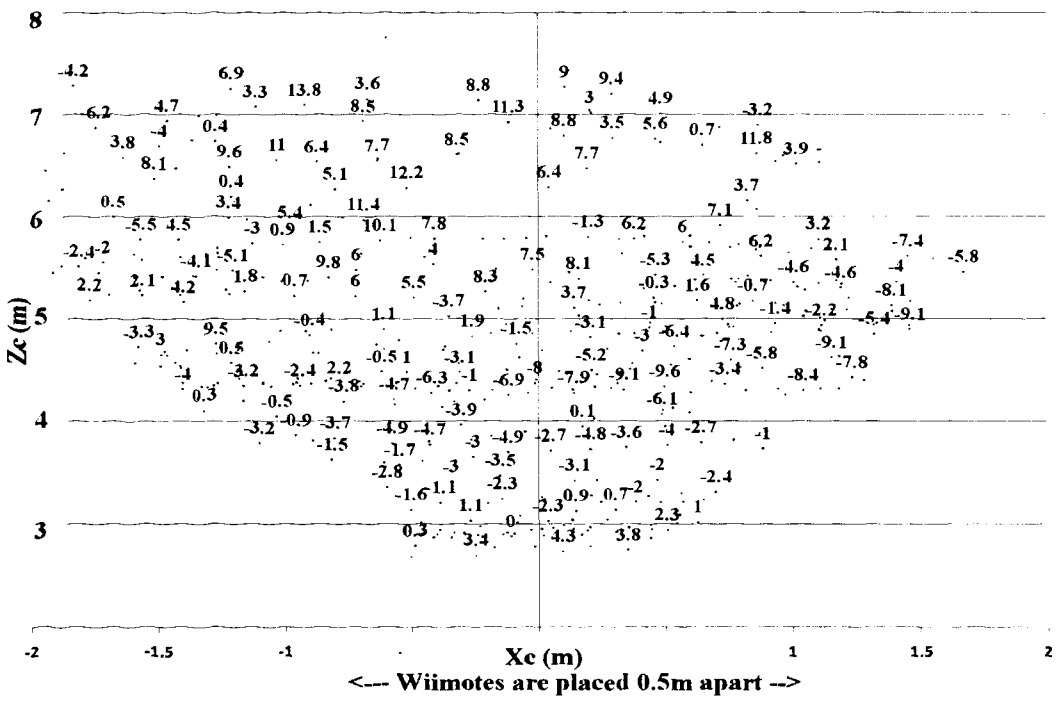


Figure 4.7. Measurement error plot (errors in mm)

5. SVOBODA'S ALGORITHM FOR CALIBRATION OF A MULTI (>2) CAMERA SYSTEM

Svoboda et al. [12] proposed a method for calibrating multiple cameras together. The number of cameras that can be calibrated together using this method is 3 minimum, and there is no upper limit. The only calibration object required is an IR point source. Several IR LEDs can be mounted closely together on a wand such that together they form a uniform IR light source which can be seen practically by all cameras in all directions. The calibration can be achieved by moving an IR LED through the work volume, and the cameras being calibrated do not have to see all of the points where the data is recorded, i.e., only sufficient overlap among the cameras being calibrated is necessary. This property helps in terms of obtaining more coverage volume per Wiimote in the stereo vision system. The sections below summarize the practical implementation and performance evaluation of Svoboda's algorithm to develop Wiimote vision systems.

5.1 THE MULTI-CAMERA SELF-CALIBRATION TECHNIQUE

Svoboda's algorithm is briefly as follows. Let m be the number of cameras to be calibrated together, and n be the number of calibration points recorded. Also, let X_j be a calibration point with homogeneous coordinates $[x_j, y_j, z_j, 1]^T$ in the world coordinates, where $j = 1,\ldots,n$. The pinhole camera model in Eq. (4) can be written for m cameras and n calibration points as:

$$\begin{bmatrix} \lambda_1^1 \begin{bmatrix} u_1^1 \\ v_1^1 \\ 1 \end{bmatrix} \cdots\cdots \lambda_n^1 \begin{bmatrix} u_n^1 \\ v_n^1 \\ 1 \end{bmatrix} \\ \cdots\cdots \\ \lambda_1^m \begin{bmatrix} u_1^m \\ v_1^m \\ 1 \end{bmatrix} \cdots\cdots \lambda_n^m \begin{bmatrix} u_n^m \\ v_n^m \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P^1 \\ \vdots \\ P^m \end{bmatrix}_{3mX4} [X_1 \quad \cdots \quad X_n]_{4Xn} \qquad (28)$$

where P_i is a 3x4 matrix for camera i. It contains all 11 camera parameters (5 intrinsic and 6 extrinsic). Thus the calibration here involves finding the camera projection matrices P^i and the scaling factors λ_j^i. In the calibration process, the 2D projection coordinates (u,v) of the image of a 3D marker can be first obtained. Then the wrongly recorded data points (outliers) can be detected using the RANSAC analysis [19]. Then, the scaling factors λ_j^i can be estimated and the missing points estimated by the method described by Martinec and Pajdla [20]. The projective structures can be further optimized using the bundle adjustment [21], and the overall matrix in Eq. (28) further factorized to get matrices P and X [22].

5.2 EVALUATION RESULTS

Figure 5.1 shows the experimental setup with 4 Wiimotes mounted on a CAVE frame for evaluation of Svoboda's calibration technique. Two IR LEDs with a known distance apart were waved through the work volume randomly and their image data was collected. The intrinsic and extrinsic parameters were then determined and used to compute the position of each LED at every calibration point. Points were recorded when the LED was detected by at least three of the four Wiimotes. Then the scaling factor was determined with a sufficient number of observations.

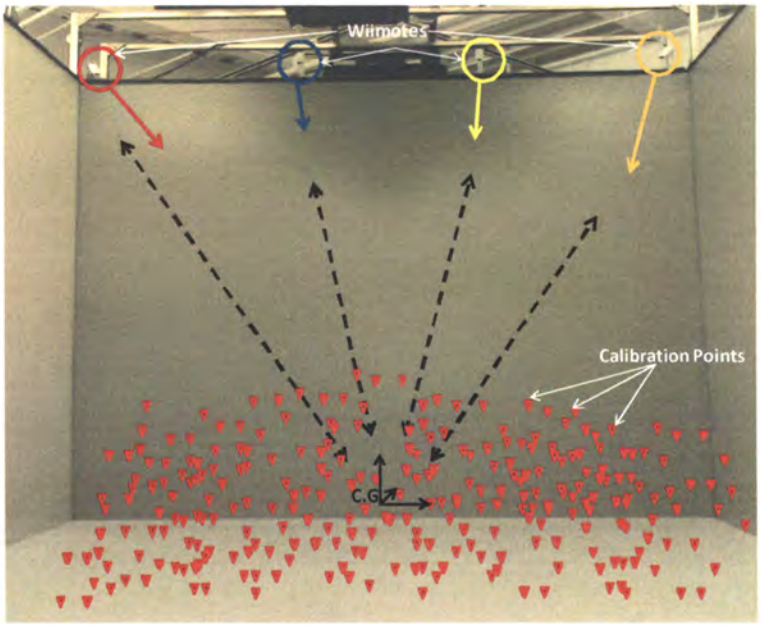


Figure 5.1. Svoboda's algorithm for self-calibration of 4 Wiimotes

To test the measurement accuracy, two IR LEDs were placed a known distance apart. The measurement accuracy is shown in Figure 5.2. It shows the difference between the actual distance between the two LEDs and that estimated by measuring the absolute positions of the 2 LEDs with the motion capture system employing Svoboda's algorithm. As indicated in Figure 5.2, the average magnitude of measurement error was 2.37 mm, which is significantly smaller than the 18.9 mm resulted from using the combination of Zhang's and Horn's algorithms as indicated in Figure 5.3.

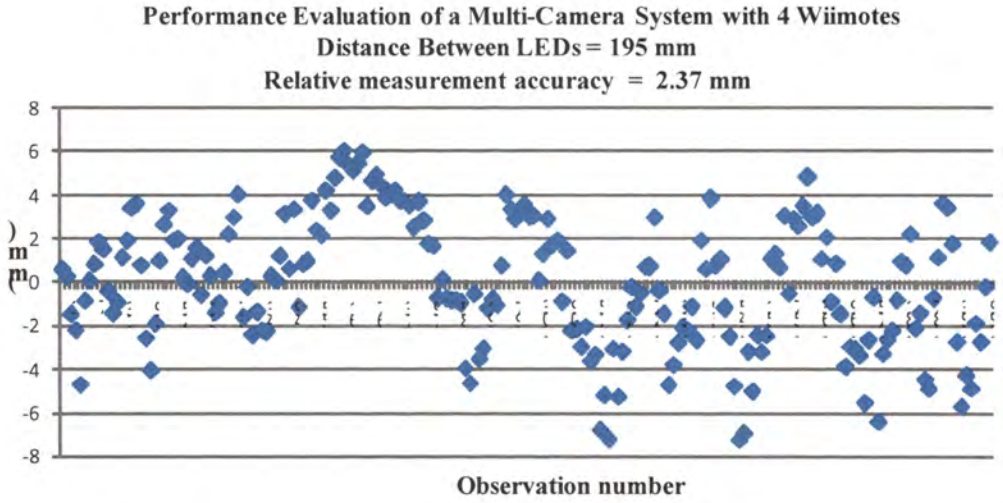


Figure 5.2 Measurement accuracy for a camera vision system with 4 Wiimotes using Svoboda's calibration algorithms

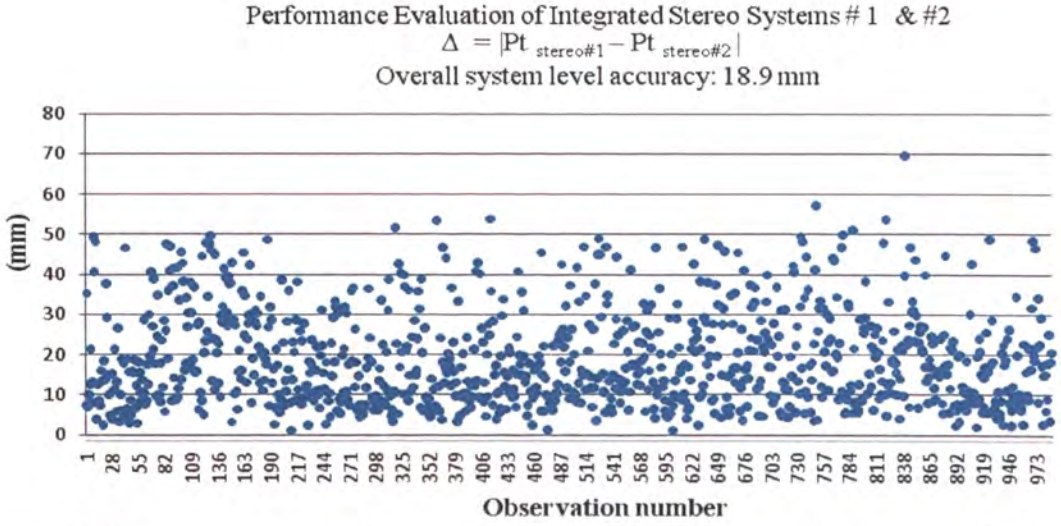


Figure 5.3 Measurement error for an integrated two stereo vision system calibration using Zhang's and Horn's algorithm

6. AN INTEGRATED CAMERA CALIBRATION TECHNIQUE

A new integrated calibration technique was developed and evaluated to calibrate multiple cameras together to form a low-cost motion capture system. This is a two-stage calibration technique: first the intrinsic parameters are determined with Zhang's method [9] and then the extrinsic parameters are determined with Svoboda's calibration method [12]. The individual camera vision systems were further integrated using Horn's algorithm [13] to extend the range of motion capture. This technique has been successfully implemented and demonstrated with good measurement accuracy using multiple Wiimotes.

6.1 DECOMPOSITION OF CAMERA MATRIX INTO MATRICES OF INTRINSIC AND EXTRINSIC PARAMETERS

As discussed above, Svoboda's algorithm can be used to obtain intrinsic and extrinsic parameters simultaneously for the cameras in a multi-camera vision system. This algorithm has been shown in the above experimental evaluation to be more accurate than the combination of Zhang's and Horn's algorithms for a multi-camera vision system with four Wiimotes. However, the experimental evaluations have also indicated that the values of intrinsic parameters obtained from Svoboda's algorithm deviate more from the known values of the Wiimotes' intrinsic parameters compared with Zhang's algorithm, especially when the overlapping coverage volume among the 4 Wiimotes is becoming small. Thus, the basic idea behind the development of an integrated camera calibration technique is decomposing the camera calibration matrix P^i in Eq. (9) into a matrix of intrinsic parameters and a matrix of extrinsic parameters, and then using Zhang's

algorithm to determine the intrinsic parameters and using Svoboda's algorithm to determine the extrinsic parameters.

The matrix P^i in Eq. (28) can be decomposed into separate matrices of intrinsic and extrinsic parameters as follows [22]:

$$P^i = A\,[R^i\ t^i] \tag{29}$$

where the two matrices on the right hand side of Eq. (29) represent the intrinsic camera parameters and the extrinsic camera parameters, respectively. Thus, the intrinsic camera parameters can be determined using Zhang's algorithm and then by using these values of intrinsic parameters in Eq. (28), the extrinsic parameters of a system of cameras can be determined together using Svoboda's algorithm [12].

To determine the minimum number of cameras required for implementing the integrated camera calibration technique, it can be noted that there are 11 unknowns in the matrix P in Eq. (29). After the five intrinsic parameters have been calibrated using Zhang's algorithm, there are six independent extrinsic parameters left that need to be calibrated using Svoboda's algorithm. Thus, it is necessary to have a set of six equations to solve for the extrinsic parameters associated with the P^i matrix. Since each point correspondence between the two images leads to two equations, a minimum of three image correspondences is required to solve for P^i. Thus, the marker such as an IR LED must be visible to at least 3 cameras for any given instance in order for its position to be captured.

Computer software has been written in C# using Wiimote Library functions to identify dynamically the correspondences between the images for any given marker point. The multi-camera stereo vision system identifies a set of three or more Wiimotes

that can detect the marker for any given instance. The obtained image data is then used to determine the extrinsic parameters and then to determine the position of each marker point using Eq. (28).

6.2 EXPERIMENT WITH A MULTI-CAMERA VISION SYSTEM WITH 4 WIIMOTES

In this experiment, 4 Wiimotes are calibrated together using the integrated camera calibration technique described above. The camera arrangement is as shown in Figure 3.8(b). The calibration steps adopted for such a 4 Wiimote vision systems are summarized in Figure 6.1.

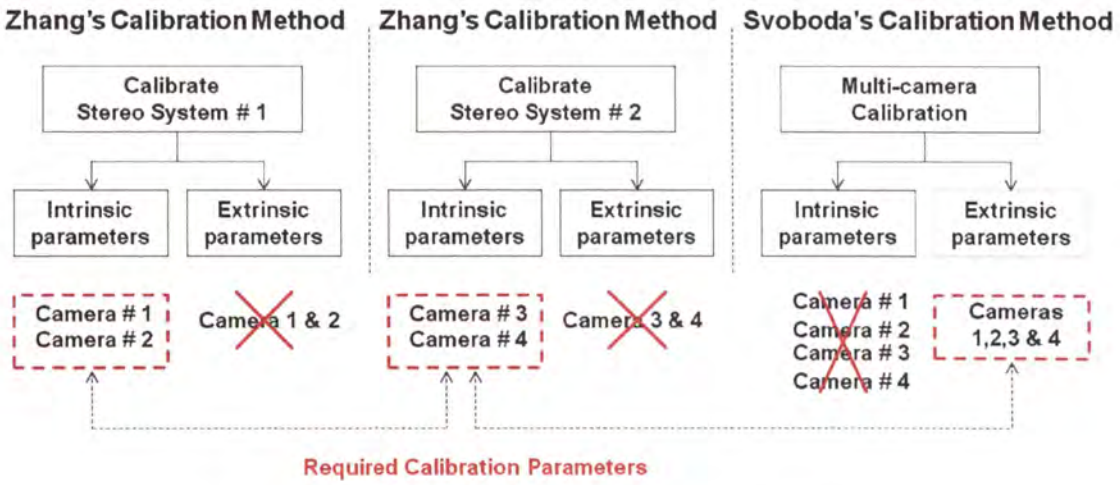


Figure 6.1. Calibration of a 4 Wiimote vision system

To estimate the measurement accuracy of the vision system, a wand with two IR LEDs mounted at a known distance apart is moved randomly in the space. By using the integrated camera calibration technique described, the positions of the two markers and the distance between them are tracked in real time. The measurement accuracy is shown

in Figure 6.2, wherein the image data used is the same as that corresponding to Figure 5.2.

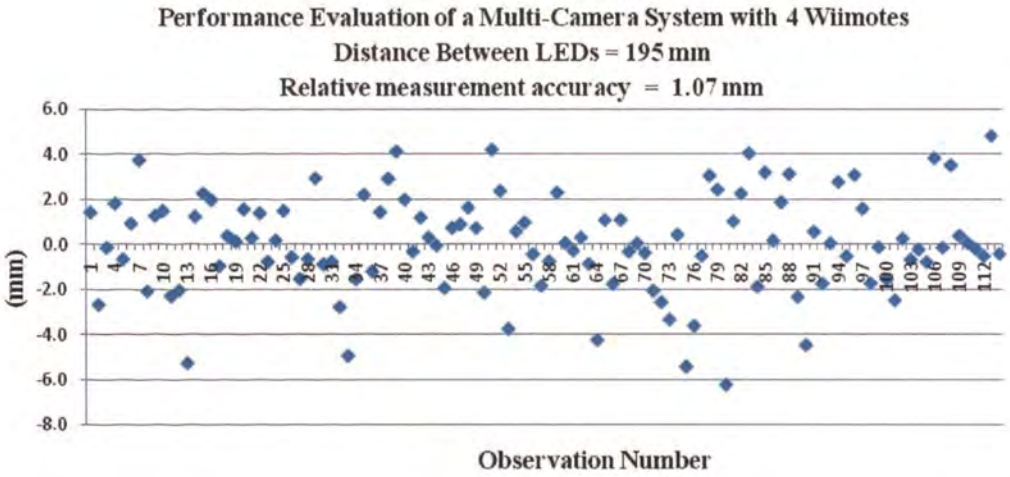


Figure 6.2 Measurement accuracy for a multi-camera vision system with 4 Wiimotes using the integrated calibration technique

By first determining the intrinsic parameters for each of the 4 Wiimotes using Zhang's method, the measurement accuracy has improved from an average error of 2.37 mm (Figure 5.2) to 1.07 mm (Figure 6.2).The measurement error is much smaller than the average error of 18.9 mm (Figure 5.3) resulted from using the combination of Zhang's and Horn's algorithms.

6.3 EXPERIMENT WITH A MULTI-CAMERA VISION SYSTEM WITH 8 WIIMOTES

In order to cover a larger measurement volume, the integrated calibration technique is also evaluated for 8 Wiimotes arranged such that each set of 4 Wiimotes forms a multi-camera vision system and the two systems are integrated into a single tracking system using Horn's algorithm [13] to determine the relative position and orientation of the two systems. Figure 6.3 shows the 8 Wiimotes mounted in the CAVE, with two multi-camera subsystems each consisting of four Wiimotes. Figure 6.4 shows the measurement

accuracy for the integrated multi-camera vision system with 8 Wiimotes. The average magnitude of measurement error for the data in this Figure 6.4 is 8.7 mm. Note that all 8 Wiimotes hereby cannot be calibrated together as a Single Vision system, as all 8 Wiimotes together have no common overlapping area. Thus, two sets of 4 Wiimotes each have been identified which offer sufficient overlap between them to be calibrated as a Single Vision system.

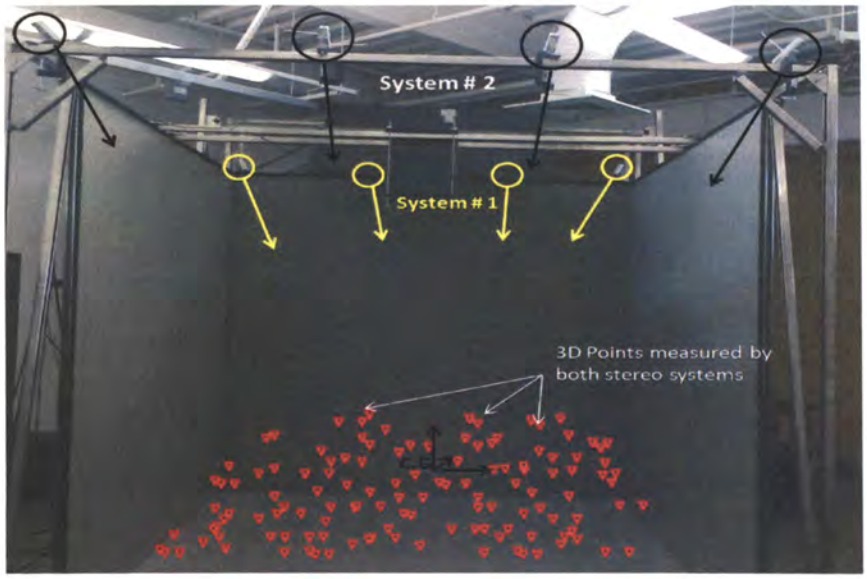


Figure 6.3 A multi-camera vision system with 8 Wiimotes

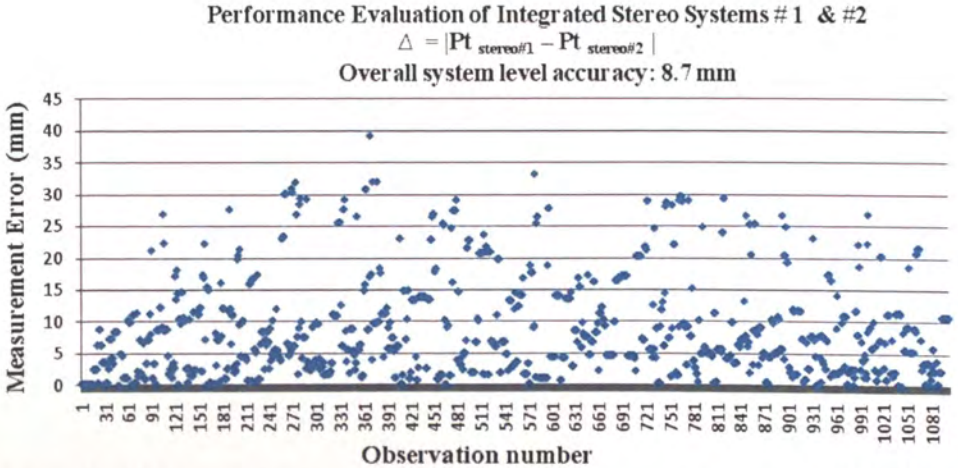
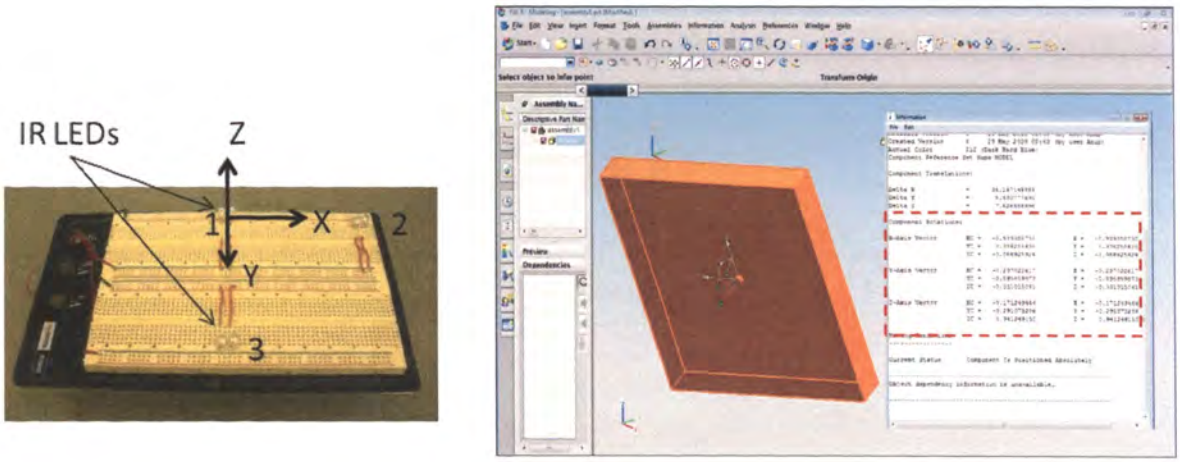


Figure 6.4 Measurement errors for the multi-camera system with 8 Wiimotes using the integrated calibration technique

7. ORIENTATION TRACKING WITH 3 LEDS

All 6 DOF of a part can be tracked with the help of a wand mounted with 3 LEDs as shown in Figure 7.1 below. Three points define a plane in 3D space. The absolute positions of all the 3 LEDs mounted on the wand can be estimated dynamically by the Wiimote vision system as discussed in the earlier sections. The plane passing through the LED markers should be parallel to the plane of the wand to be tracked.



(a) Wand with IR LEDs (b) Orientation of Wand as indicated in Siemens NX5

Figure 7.1 Orientation tracking with 3 LED markers

There are different ways in which the absolute orientation of a plane in 3D space can be defined, viz. Quaternions, Euler angles, Rotation Matrices, etc. In particular, Siemens NX5 APIs require the orientation of the object to be defined in a standard 3x3 rotation matrix so as to be simulated at the CAD level. Thus the orientation of the wand in the real world needs to be estimated from the known absolute positions of 3 LEDs mounted on it and later express the same in a standard 3x3 matrix form.

7.1 ESTIMATION OF ROTATION MATRIX FOR KNOWN ABSOLUTE POSITIONS OF 3 POINTS ON A PLANE

For any matrix to be accepted by NX5 as a valid rotation matrix, it needs to bear all of the following set of properties:

- R is an orthogonal matrix. i.e.

  $RR^T = I$ where R^T is the transpose of R and I is the identity matrix.

- R is a unit matrix i.e. $|R| = 1$.

- R is a normalized matrix.

In the Euclidean space, the rows of the rotation matrix R represent the unit vectors along the coordinate axes of the rotated space. As shown in Figure 7.1(a), let us assume that the wand is aligned with the part to be tracked such that at the CAD model level of the part, positive X-axis lies along the line connecting LEDs 1 and 2, whereas the positive Y-axis lies along the line connecting LEDs 1 & 3.

Let the absolute positions of LEDs 1,2 and 3 as measured by the Wiimote vision system be (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3), respectively. Thus, a unit vector along positive X axis can be estimated as,

$$\hat{x} = \frac{(x_2 - x_1)\,\hat{i} + (y_2 - y_1)\,\hat{i} + (z_2 - z_1)\,\hat{k}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \tag{30}$$

Similarly, a unit vector along positive Y axis can be estimated as,

$$\hat{y} = \frac{(x_3 - x_1)\,\hat{i} + (y_3 - y_1)\,\hat{i} + (z_3 - z_1)\,\hat{k}}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}}$$

Thus, a unit vector normal to the plane containing vectors X and Y can be given as

$$\hat{z} = \hat{x} \times \hat{y} \tag{31}$$

However, in practical conditions, the vectors 1-2 and 1-3 may not be exactly perpendicular. For the sake of convenience let us assume that positive X axis perfectly coincides with the positive X axis of the part as defined at the CAD level. Vector $\hat{z}$ is normal to the plane of the wand. Thus, unit vector along positive Y-axis can be re-estimated as

$$\hat{y} = \hat{z} \times \hat{x} \tag{32}$$

Thus, the required orientation matrix can be expressed as

$$R = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix}$$

where (r_{x1}, r_{x2}, r_{x3}) are the coefficients of $\hat{i}$, $\hat{j}$ and $\hat{k}$ for unit vector $\hat{x}$ from Eq. (30), (r_{y1}, r_{y2}, r_{y3}) are the coefficients of $\hat{i}$, $\hat{j}$ and $\hat{k}$ for unit vector $\hat{y}$ from Eq. (32), and (r_{z1}, r_{z2}, r_{z3}) are the coefficients of $\hat{i}$, $\hat{j}$ and $\hat{k}$ for unit vector $\hat{z}$ from Eq. (31).

Thus, for all 6 D.O.F. tracking, care needs to be taken at the user's end to ensure that the assumed positive x-axis lies along the positive X-axis as defined for the part at the CAD level.

7.2 SORTING AND SEQUENCING OF LEDS FOR DYNAMIC ORIENTATION TRACKING

As discussed in Section 7.1, 3 LEDs are sufficient for the wand to estimate its absolute position and orientation in the real world. However, while implementing Wiimote based system for multiple LED marker tracking one needs to address the issue of identification and sorting of the multiple markers for a multi-Wiimote system. The problem is defined in Figure 7.2 below.

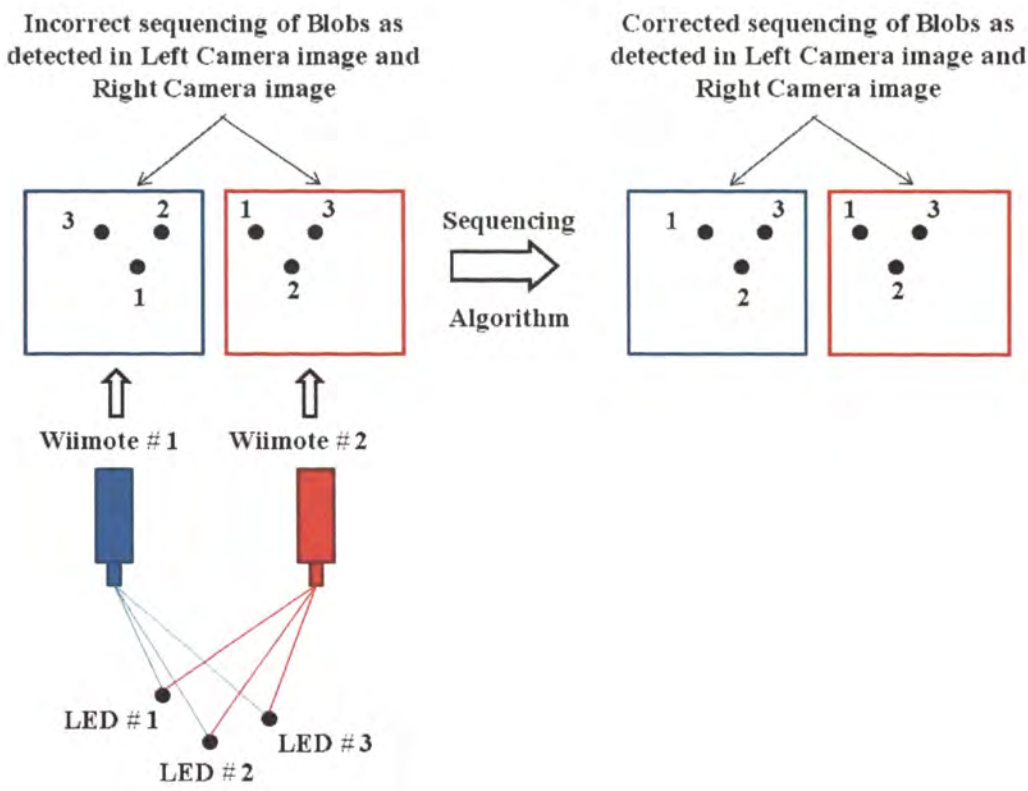
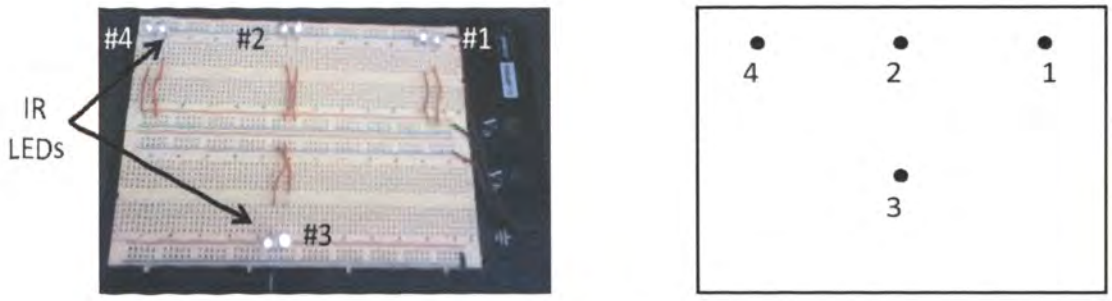


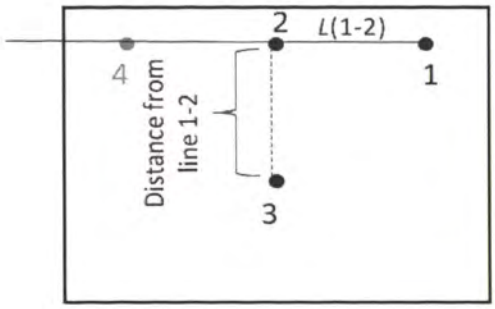
Figure 7.2 Sequencing of LEDs for multi marker tracking

A Wiimote can typically detect up to 4 IR hot spots in real world. Consider an assembly of 2 Wiimotes exposed to 3 different IR markers in the real world as shown in Figure 7.2. To be able to estimate the absolute positions of individual LEDs one needs to identify the correspondence between the blobs as detected by the different Wiimotes constituting the vision system. To achieve this auto-sequencing of blobs as detected by Wiimotes, an algorithm has been implemented as described below.

The sequencing needs to be done for the IR LEDs as discussed above. This is done by the geometric pattern recognition for individual camera images. The idea here is to have a geometric pattern that is invariant to the projection from 3D to 2D. As can be seen in Figure 7.3, the 3 LEDs of the wand are arranged in more or less a line and the 4^{th} LED is distinctly located above this line.



(a) Wand with 4 LEDs (b) Blobs as detected by Wiimote Sensor Chip



(c) Distance of a point from a line

Figure 7.3 Wand with 4 LEDs for 6 DOF Motion Tracking

In the 2D image data of the Wiimote the 3 LEDs also form more or less a line. Therefore, the first step in the sorting algorithm is to find the 3 image points which are most close to form a line. The algorithm can be given as :

```
Initialize lineStartPoint, lineEndPoint, onLinePoint, freePoint
Dist = onLinePoint.DistanceToLine(lineStartPoint, lineEndPoint, out lambda);
// check if projected point is between line end and start point
//lambda between 0 and 1 if point lies between start and end points
if ((lambda > 0) && (lambda < 1))
{
        // if distance is short, make this combination the result
```

if (dist < m_pointLineDist)

Intechange m_lineStartPoint <-> lineStartPoint;

}

Let us consider the randomly generated order of 4 points in an image as shown in Figure 7.3(b). By default, the rank assigned to the blobs depends on the order in which individual markers get exposed to the Wiimote.

To begin with, assume the first point (#1) to be the line start point and the second point (#2) to be the line end point. Now the distance of the third point (#3) from the line as defined by points #1 and #2 above can be easily estimated as indicated in Figure 7.4.

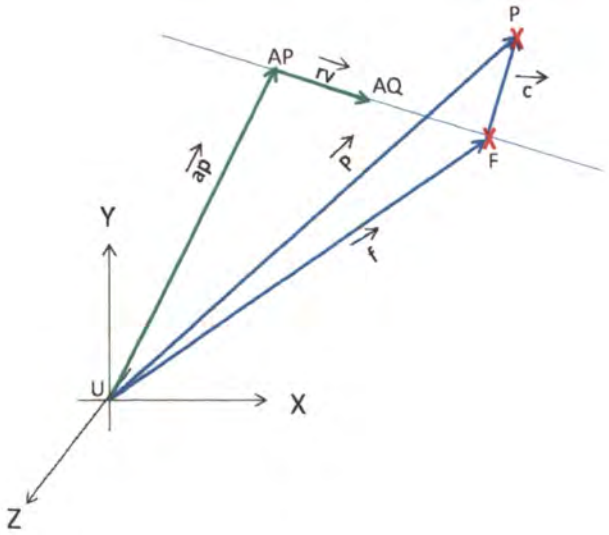


Figure 7.4 Finding distance of a point from line and position of its projection

Let us assume point AP = *lineStartPoin (*or point #1*)* , *AQ = lineEndPoin (*or point #2*), P = onLinePoint (*or points #3*)*, F = projection of point P on line connecting AP and AQ, $\bar{c}$ = projection vector for point P. Thus the len($\bar{c}$) is nothing but distance of point P from the required line as indicated. Now as indicated,

$$\bar{c} = \bar{p} - \bar{f}$$

But $\bar{f}$ can be given as $\bar{f} = \overline{ap} + \overline{rv} * \lambda$, where λ is a scalar.

Substituting, $\bar{c} = \bar{p} - (\overline{ap} + \overline{rv} * \lambda)$

As $\bar{c}$ is perpendicular to $\overline{rv}$,

$$\bar{c} \cdot \overline{rv} = 0 \quad \Rightarrow \quad \bar{p} \cdot \overline{rv} - \overline{rv} \cdot (\overline{ap} + \lambda * \overline{rv}) = 0$$

Simplification gives, $\lambda = (\bar{p} - \overline{ap}) \cdot \frac{\overline{rv}}{rv^2}$

Thus, vector $\bar{c}$ can now be estimated as, $\bar{c} = \bar{p} - (\overline{ap} + \overline{rv} * \lambda)$

And thus, length of projection = |c|

Thus, it can be seen that the position of the projection of point #3 on line 1-2 can also be estimated. If the projection point is outside the line start and line end point the λ value will be below 0 or greater than 1.

The known length of projection of point #3 on line 1-2 is compared to the earlier calculated smallest projection length. If it is smaller, then this order of image points will be stored as the best solution.

Thus, in order to get the right solution, this function has to be called with all possible combinations of the 4 LED image points. A typical call to the function thus consists of following combinations of inputs:

TestPoints(irList[0], irList[1], irList[2], irList[3]);

TestPoints(irList[0], irList[1], irList[3], irList[2]);

TestPoints(irList[0], irList[2], irList[1], irList[3]);

TestPoints(irList[0], irList[2], irList[3], irList[1]);

TestPoints(irList[0], irList[3], irList[1], irList[2]);

TestPoints(irList[0], irList[3], irList[2], irList[1]);

TestPoints(irList[1], irList[2], irList[0], irList[3]);

TestPoints(irList[1], irList[2], irList[3], irList[0]);

TestPoints(irList[1], irList[3], irList[0], irList[2]);

TestPoints(irList[1], irList[3], irList[2], irList[0]);

TestPoints(irList[2], irList[3], irList[0], irList[1]);

TestPoints(irList[2], irList[3], irList[1], irList[0]);

The above procedure for sorting of the 4 LEDs is, thus, to be repeated for all the 2D images of all the cameras in the system.

Now that the right order of the three points has been obtained from the line and the 4th point, it is still necessary to determine the right direction of the line. In the wand as shown in Figure 7.3, one of the LEDs is below the line. If the start and end point of the line would be interchanged then the 4th LED would be above the line. Mathematically one needs to check if the order of the points is clockwise or counterclockwise. The algorithm for the same can be given as

// Only remaining test is to check if line start and end point are in right order

// Check start and end line point with the free point

// If clockwise direction then ok, if counterclockwise then exchange start and end points

Vector E1 = m_lineStartPoint - m_freePoint; // P1-P2

Vector E2 = m_lineEndPoint - m_freePoint; // P3-P2

*if ((E1.X * E2.Y - E1.Y * E2.X) >= 0) then clockwise = TRUE;*

else clockwise = FALSE

if (clockwise=FALSE) then

Interchange m_lineEndPoint and m_lineStartPoint

Once the right order of the points is obtained, the position estimation algorithm can be run to estimate the absolute position of individual LEDs from Eq. (28).

7.3 ORIENTATION ACCURACY EVALUATION WITH WII MOTION PLUS

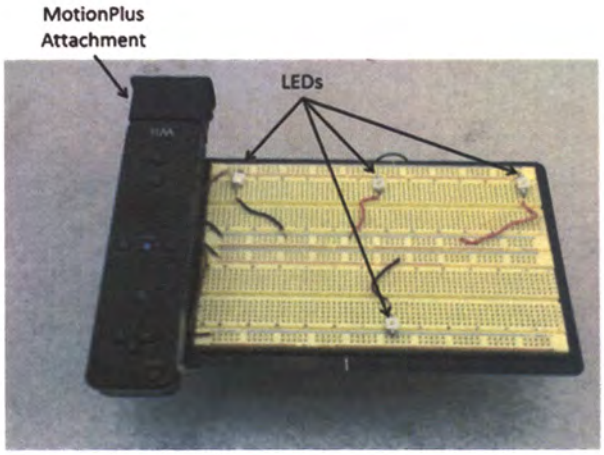
A new class of Inertia Navigation Systems (INS) has evolved in last few years, using the MEMS inertia sensing technology. These tracking devices typically need a a computer, accelerometers (motion sensors) and gyroscopes (rotation sensors) to continuously estimate via dead reckoning the orientation, and velocity (direction and speed of movement) of a moving object without the need for external references. Such MEMS based 3 DOF (pitch, roll and yaw) tracking systems typically offer a measurement accuracy of 0.5° along pitch, roll and yaw.

To improve the orientation measurement accuracy of the Nintendo Wiimote device, Nintendo recommends the use of its Wii MotionPlus accessory. It is an expansion device for the Wiimote game controller that allows it to more accurately capture complex motion. Wii MotionPlus typically offers a frame rate of 100Hz along all three Euler rotations. Typically in *slow* mode, 20 units of change indicates a turning speed of 1°/sec whereas in the *fast* mode 20 units of change indicates a turning speed of 5°/sec. Thus time track can be kept so as to estimate the change in angular position of the controller for any instance.

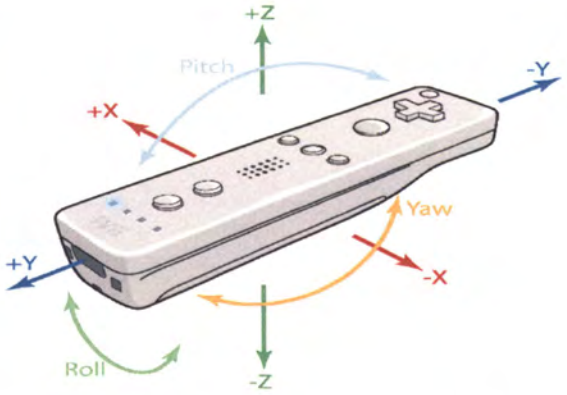
Nintendo Wiimote with MotionPlus accessory is used hereby as a benchmark to compare the performance of Wiimote Tracking System for orientation tracking (3 DOF) (pitch, roll and yaw). Orientation accuracy of the wand was checked for rotations along all three axes (X,Y,Z). As the total time of the experiment is short (<20 mins) and the surrounding conditions are also unchanged during the entire experiment, the likely drifting error involved in the measurement can be safely neglected.

A camera vision system with 4 Wiimotes mounted on one side of the CAVE was setup as shown in Figure 7.6. The system was calibrated as a single vision system in a similar way as discussed in Section 6. A Wiimote with MotionPlus attachment was mounted on a wand with 4 LEDs as shown in Figure 7.5(b). Figure 7.5(c) indicates the coordinate system convention followed by Wiimote accelerometer and MotionPlus gyroscopes. It is necessary to ensure before commencement of the experiment that the coordinate system as followed by the Wiimote tracking system and that followed by Wiimote and MotionPlus are the same throughout the experiment. The absolute orientation value can be directly read from the Wiimote accelerometer and displayed on the screen dynamically. Thus, to begin with, Wiimote was adjusted such that it reads zero along all pitch, roll and yaw values. The wand's position and orientation were recorded for the same position. This transformation can now be used to shift the Wiimote tracking system's coordinate system for all the readings hereon, to ensure that MotionPlus and Wiimote tracking systems read yaw, pitch and roll in the same Global Coordinate System.

(a) Wii MotionPlus Accessory (ω_1, ω_2, ω_3 are angular velocities along three major axes of rotations)



(a) Wand with 4 LEDs with Wii MotionPlus attachment



(c) Wiimote Accelerometer and MotionPlus coordinate system

Figure 7.5 Orientation tracking accuracy check with Nintendo Wii MotionPlus accessory

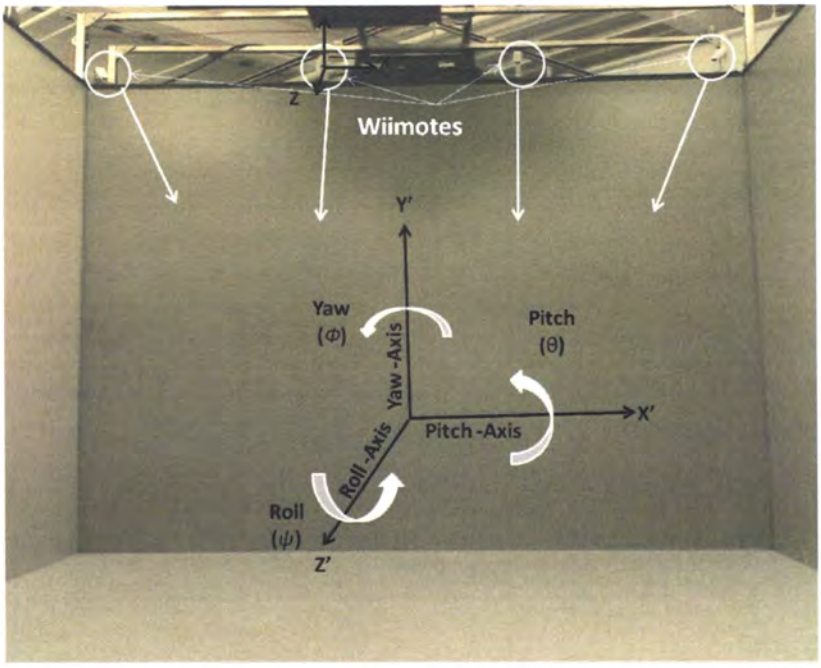
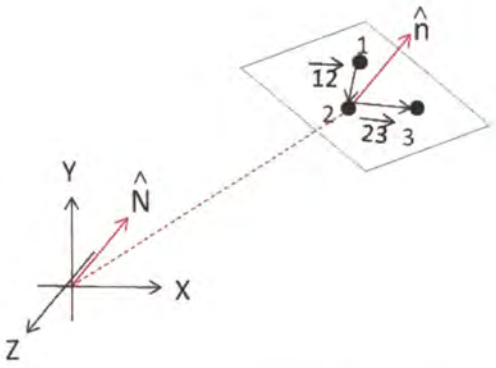


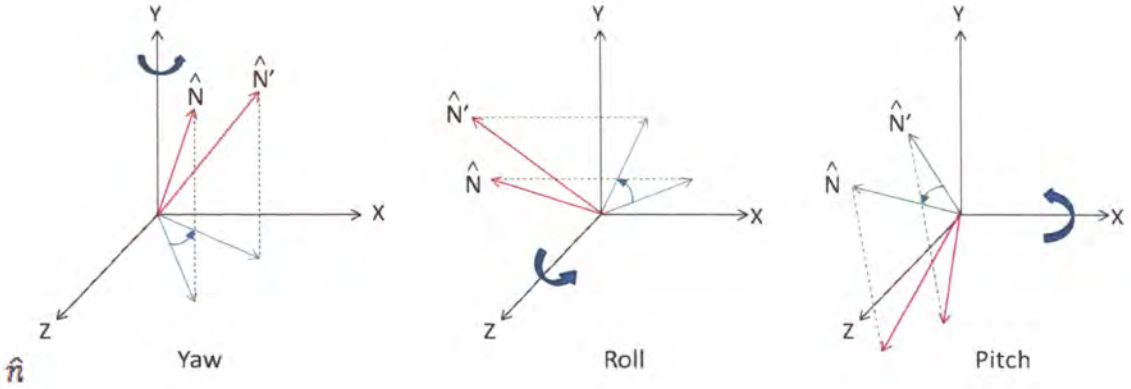
Figure 7.6 Orientation tracking evaluation with a 4-camera vision system

As discussed before, Wii MotionPlus measures the change in the velocity along all pitch, roll and yaw for the Wiimote with an accuracy of 0.5^{o}. Thus keeping track of time elapsed along with known initial position, one can easily measure the absolute change in pitch, roll and yaw of the Wiimote assembly for a given time instance.

One also needs to express the wand orientation in terms of pitch, roll and yaw to have a clear comparison with MotionPlus measurements. Thus, the mathematical problem here is to estimate the absolute Euler angles of a plane given the absolute positions of 3 points lying on the same plane. Consider a plane as shown in Figure 7.7. The plane represents the plane of the wand and the 3 points indicate the three nonlinear LEDs mounted on the wand.



(a) Plane representing a wand with 3 LEDs with unit surface normal



(b) Estimating Euler angles of a plane with 2D projections of normal to the plane

Figure 7.7 Estimating pitch, roll and yaw of a plane for known 3 points on the plane

With the known absolute positions of points 1, 2 and 3, one can easily estimate the surface normal to the plane. Let $\widehat{12}$ be the unit vector along vector joining LEDs #1 and #2 in Euclidean space. Similarly, let $\widehat{23}$ be the unit vector along vector joining LEDs #2 and #3 in Euclidean space. Thus, a normal unit vector to the plane $\hat{n}$ can be given as

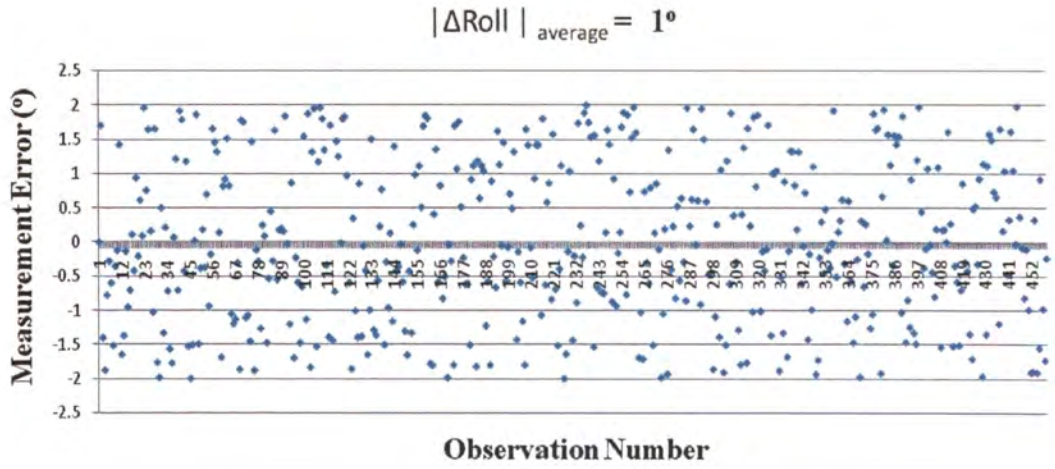
$$\hat{n} = \widehat{12} \times \widehat{23}$$

Shifting the vector to the origin of the coordinate system, let $\hat{N}$ be the required unit normal vector. Thus, following the same conventions as that of Wiimote

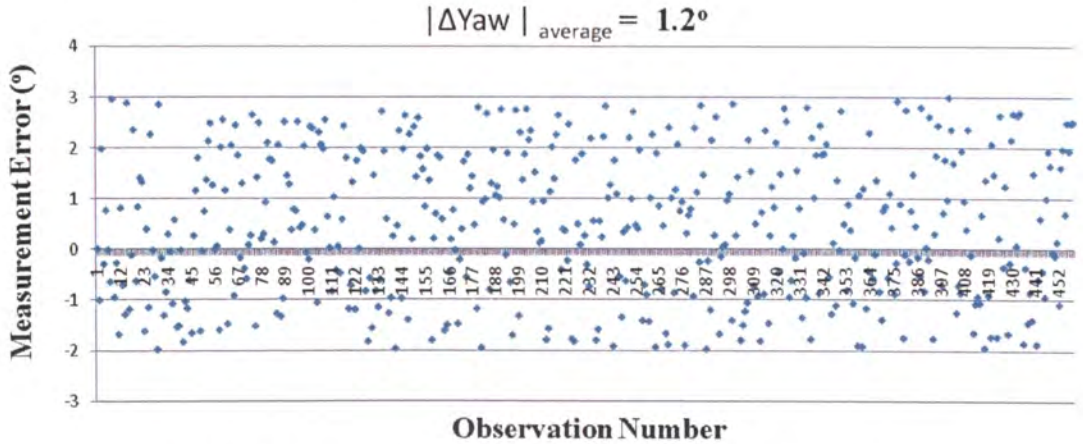
Accelerometer and MotionPlus as indicated in Figure 7.7, the 2D projections of vector $\tilde{N}$ in the required planes can be found and the concerned angle of rotation can be calculated.

Readings were taken for a variation of $>170^{o}$ in all 3 DOF movements i.e. pitch, roll and yaw, independently.

Figure 7.8 indicates the orientation tracking accuracy of the Wiimote vision system with 4 Wiimotes.

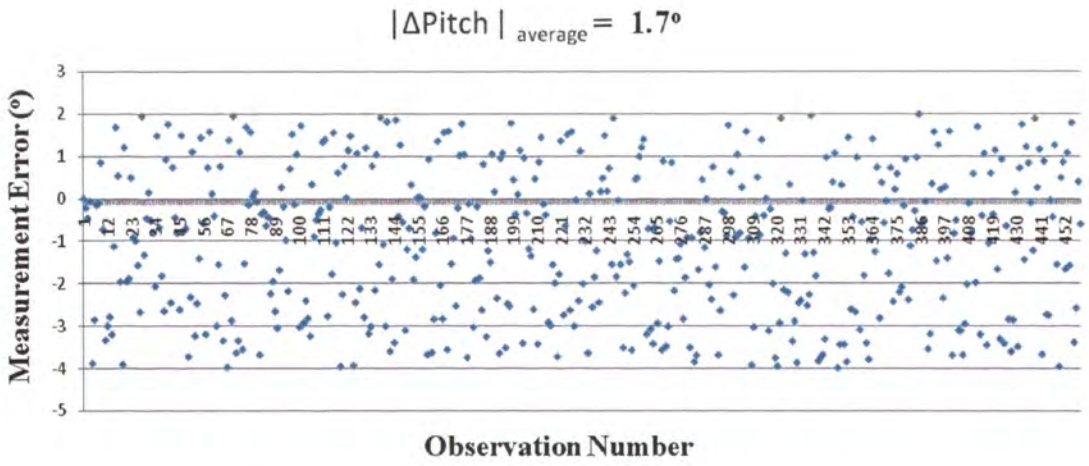


(a) Average absolute error in Roll measurement



(b) Average absolute error in Yaw measurement

Figure 7.8 Orientation tracking evaluation for a vision system with 4 Wiimotes



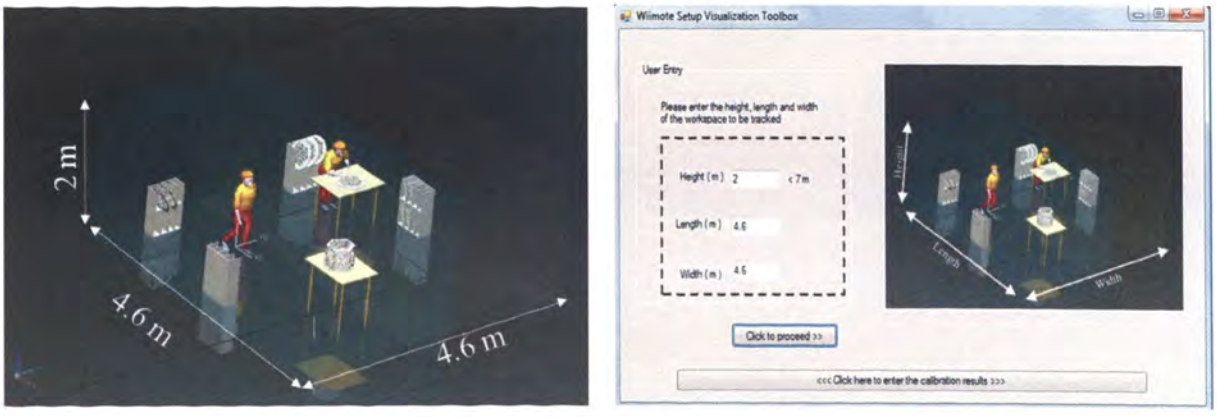
(c) Average absolute error in Pitch measurement

Figure7.8 Orientation tracking evaluation for a vision system with 4 Wiimotes (cont.)

8. WIIMOTE VISION SYSTEM SETUP TOOLKIT

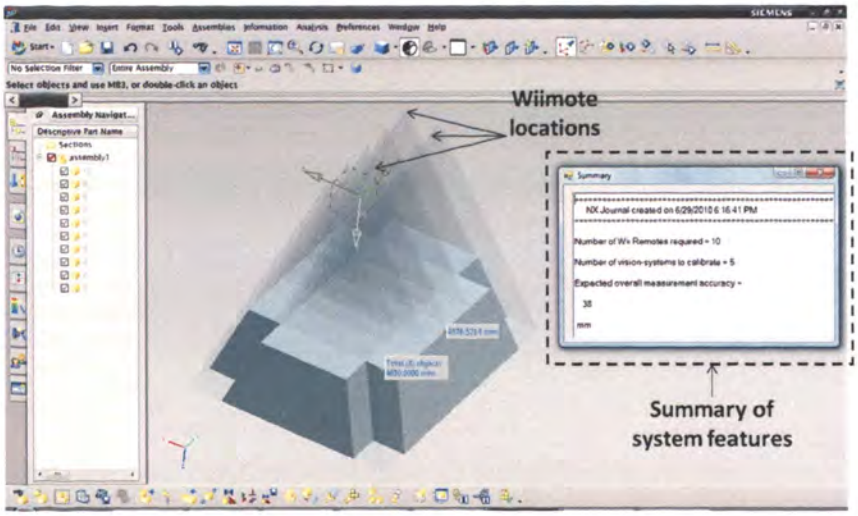
A software tool has been developed to assist the users in setting up a new Wiimote vision system. Given the dimensions of a space to be tracked (length x breadth x height), the software estimates the minimum number of Wiimotes required for tracking the markers inside the specified volume. The software also estimates the number of vision systems that need to be calibrated to setup the entire tracking system.

Based on the estimated number of vision systems required, the software also estimates the maximum possible measurement error that can incur over the entire given tracking volume, as shown in Figure 8.1. Estimating the total number of Wiimotes required, the system generates an NX5 CAD assembly so as to let the user visualize the actual shape of the tracking volume achieved as shown in Figure 8.1(c).

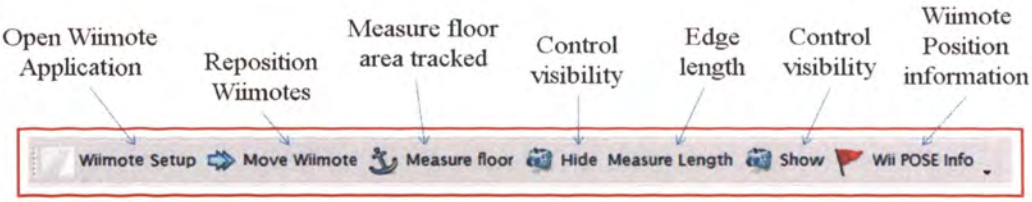


(a) Dimensions of volume to be tracked (b) Software's user interface

Figure 8.1. Wiimote Vision System Setup Toolkit



(b) Solution as generated by the Toolkit (Number of Wiimotes required = 10, maximum system measurement error = 20 mm)



(d) Wiimote Navigation Toolbar inside NX to aid in solution modification

Figure 8.1. Wiimote Vision System Setup Toolkit (cont.)

The software also lets the user adjust the positions and orientations of the individual Wiimotes depending on the space constraints for mounting the Wiimotes in the real world with the help of the Wiimote Navigation Toolbar inside NX, as shown in Figure 8.1(d). The software re-estimates the shape and volume of the actual tracked volume for the new positions of the Wiimotes.

8.1 ESTIMATING THE MINIMUM NUMBER OF WIIMOTES REQUIRED FOR TRACKING

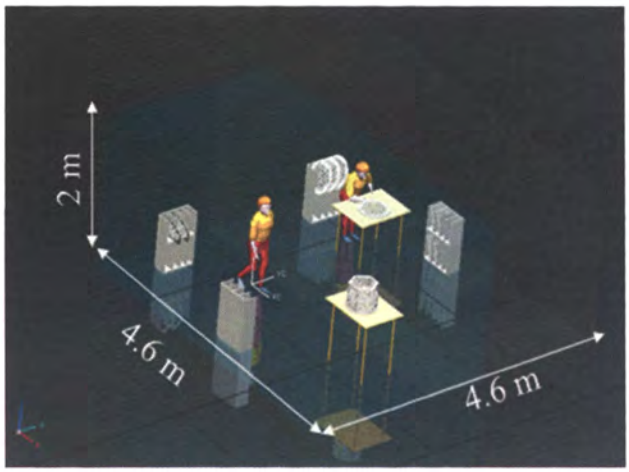
As discussed in Section 6, to form a vision system with multiple Wiimotes it is necessary to have sufficient overlap between all the Wiimotes contributing to the vision

system. This overlap is necessary for calibration of the extrinsic parameters of the Wiimotes with Svoboda's algorithm as discussed in Section 5. Similarly, to form an integrated vision system using multiple vision systems calibrated independently, Horn's method is implemented as discussed in Section 3. These constraints need to be taken into consideration while adopting a feasible camera arrangement for a given tracking volume.

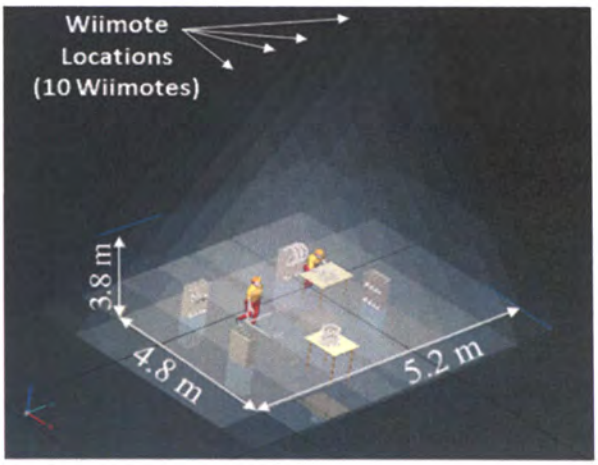
Wiimote Vision System Setup Toolkit assumes an arrangement of the cameras wherein all Wiimotes are placed 1m apart both along X and Y directions and parallel to each other as shown in Figure 8.2(c). Such kind of arrangement ensures sufficient overlap between the individual camera Field of Views (FOV) which is necessary for the calibration technique to consistently generate good calibration results. Such kind of arrangement is practically tested for consistency and repeatability of results. Such an arrangement of 3 Wiimotes offers an individual Wiimote measurement accuracy of approximately 1 mm, whereas every new integrated vision system introduces an absolute position measurement error of about 8.5mm both along X or Y axes. Thus integration of a new vision system with existing vision system (both along X and Y directions) adds an absolute measurement error of about 9.5mm.

Figure 8.2 indicates a typical example case wherein a set of 10 Wiimotes are required to have a coverage volume of 4.6 m x 4.6 m x 2 m. In order to keep the total number of vision systems to minimum, vision systems are formed with every 3 neighboring Wiimotes along X axis as shown in Figure 8.2 (c), with one Wiimote being common between the two adjacent vision systems located along X axis. This overlap between two adjacent vision systems along X-axis is required to calibrate two neighboring vision systems with respect to each other. As discussed in Section 2, the

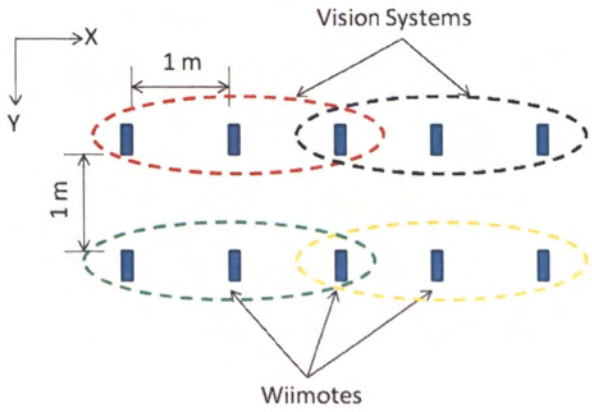
FOV of the Wiimotes is not symmetrical ($\alpha < \theta$), Vision systems cannot be formed with 3 Wiimotes along Y-axis. Thus Horn's calibration algorithm is implemented to integrate the vision systems along Y-axis.



(a) Tracking area required (4.6m x 4.6m x 2m)



(b) System suggested arrangement of Wiimotes



(c) Grouping and calibration of Wiimotes (4 Vision systems)

Figure 8.2 Typical Wiimote arrangement assumed by Wiimote Vision System Setup Toolkit

Consider a Wiimote vision system with 3 Wiimotes placed 1m apart as shown in Figure 8.3. Such a vision system offers a tracking area of 5.1 m x 3.75 m at a distance of 7m away from the Wiimotes.

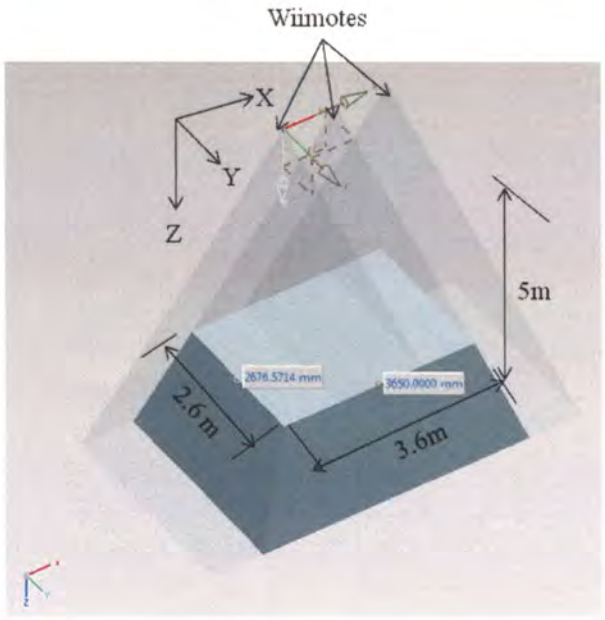


Figure 8.3 Principle of estimation of number of Wiimotes required for give tracking volume

As most of the mechanical assembly operations on shop floors are carried out within a height of 2-3m from the floor, the Toolkit considers only the tracking area offered at a distance of 2m from the shop floor (that is, at a distance of 5m from the Wiimote vision system). As shown in Figure 8.3, the tracking area offered by a Wiimote vision system with 3 Wiimotes, at a distance of 5m from the Wiimotes is (3.6m x 2.6m). As discussed in Section 6, after the calibration any 2 Wiimotes dynamically form the vision system for tracking. With this principle of working of the vision system, addition of every Wiimote along X-axis increases the tracking area of the vision system by 1m along X-axis. Similarly every Wiimote added along Y-axis to the existing Wiimote vision system increases the tracking area of the vision system by 1m along Y-axis. Considering this

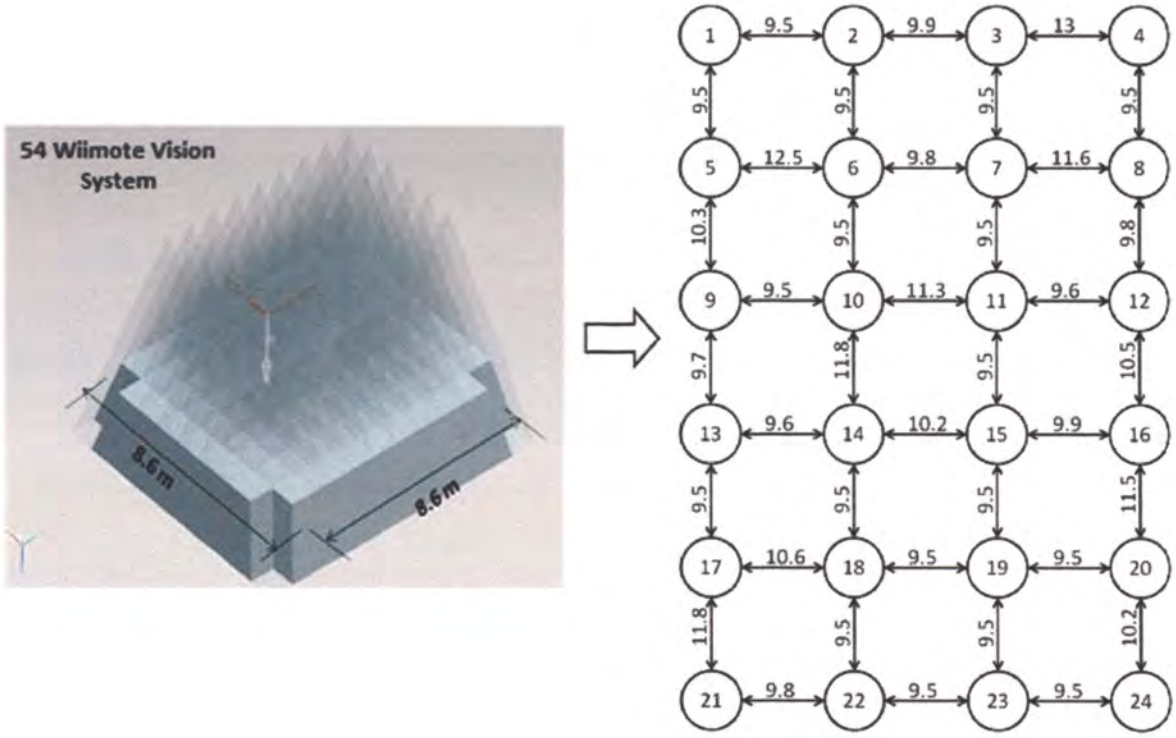
method of expanding the tracking area along X and Y-axes individually, and also the known tracking area dimensions at a distance of 5m away from the Wiimotes, number of Wiimotes required for a given tracking area can be estimated.

Code has been written in VB .NET to consider this algorithm of estimation of the number of vision systems, and the same has been

8.2 ESTIMATING MAXIMUM POSITION MEASUREMENT ERROR FOR A WIIMOTE VISION SYSTEM

In order to let the user have a fair idea about the likely attainable system level measurement accuracy, it is required to find the maximum system level measurement error incurred.

Consider a tracking volume of 8m x 8m x 2m. System estimates the number of Wiimotes required for tracking the specified volume with the assumption stated above. As shown in Figure 8.4, minimum 54 Wiimotes are required to track this volume. The total number of vision systems that need to be calibrated in this case is 24. Such a vision system can be represented by a 2D array as shown in Figure 8.4(b). In Figure 8.4(b) every node in the matrix represents an individual Wiimote vision system. Every vision system is calibrated with its neighboring vision system along X and Y direction (and not diagonally). This integration error can be expressed as the weight assigned to the edge representing the existence of known relationship between the two neighboring integrated vision systems as shown in Figure 8.4(b)



(a) Software generated solution

(b) Vision system configuration (Directed Graph)

Figure 8.4 Wiimote arrangement as suggested by Wiimote Vision System Setup Toolkit for tracking volume of 8m x 8m x 2m

It can be easily seen from Figure 8.4(b) that the maximum error occurs when the tracking is being done by the farthest vision systems, i.e. systems 1 and 24 in this case. Now as discussed earlier, there is no direct relationship between these two vision systems. Thus the required relationship i.e. coordinate transformation ($R_{resultant}$, $T_{resultant}$) between the farthest vision system and the "Master" system needs to be separately calculated. It can be seen that the less the number of intermediate vision systems between the active vision system and the master system, the less is the transformation error incurred in the measurement. Thus the problem becomes a typical shortest path finding problem for a 2D directed graph for the known weights as indicated in Figure 8.4(b).

There are different standard mathematical algorithms to solve the typical single source shortest path problem [14], viz. Bellman-Ford algorithm, Dijkstra's algorithm, Floyd-Marshall algorithm, etc. Dijkstra's algorithm [14] has been implemented hereby.

Let V be the set of vertices in the array and E be the set of edges in the array. Dijkstra's algorithm [14] solves the single-source shortest paths problem on a weighted, directed graph G = (V,E) for the case in which all edge weights w(u, v) $\geq$ 0 are defined for each edge (u,v) ϵ E.

Dijkstra's algorithm maintains a set of vertices whose final shortest-path weights from the source *s* have already been determined. The algorithm repeatedly selects the vertex u ϵ (V-S) with the minimum shortest-path estimate, adds u to S, and relaxes all edges leaving u. Dijkstra's algorithm [14] can be expressed as follows:

DIJKASTRA (G,w,s)

INITIALIZE-SINLGE-SOURCE (G,s)

$S \leftarrow \square$

$Q \leftarrow V[G]$

while $Q \neq \square$

do $u \leftarrow$*EXTRACT-MIN (Q)*

$S \leftarrow S \cup \{u\}$

for *each vertex* $v \in Adj[u]$

do RELAX(u,v,w)

MATLAB code has been generated for above algorithm. The input for the program is the 2D coordinates of the nodes of the 2D array indicating the camera vision systems along X and Y axes as indicated in Figure 8.5.

Figure 8.4(b) indicates the sequential numbering on the vision systems assumed. The direction of the edge and the weight associated to it can be defined in an input matrix E3 as indicated in Figure 8.5 [14].

Call to function : *[costs, paths] = dijkstra(V,E3,SID,FID,iswaitbar)*

Where [V] –Nx2 (or Nx3) matrix of x,y,(z) coordinates

[E3] - Px3 matrix containing a list of edge connections in the first two columns and edge weights in the third column

[SID] - 1xL vector of starting points

[FID] - 1xM vector of finish points

[costs] - LxM matrix of minimum cost values for the minimal paths

[paths] - LxM cell array containing the shortest path arrays

Figure 8.5 MATLAB implementation of Dijkstra's algorithm to find camera vision system measurement error

Figure 8.6 indicates a sample output of the program. The output includes the minimum error incurred in estimating the absolute position of the marker by the farthest located vision system. The path in the digraph to be adopted to minimize the maximum measurement error is indicated in red.

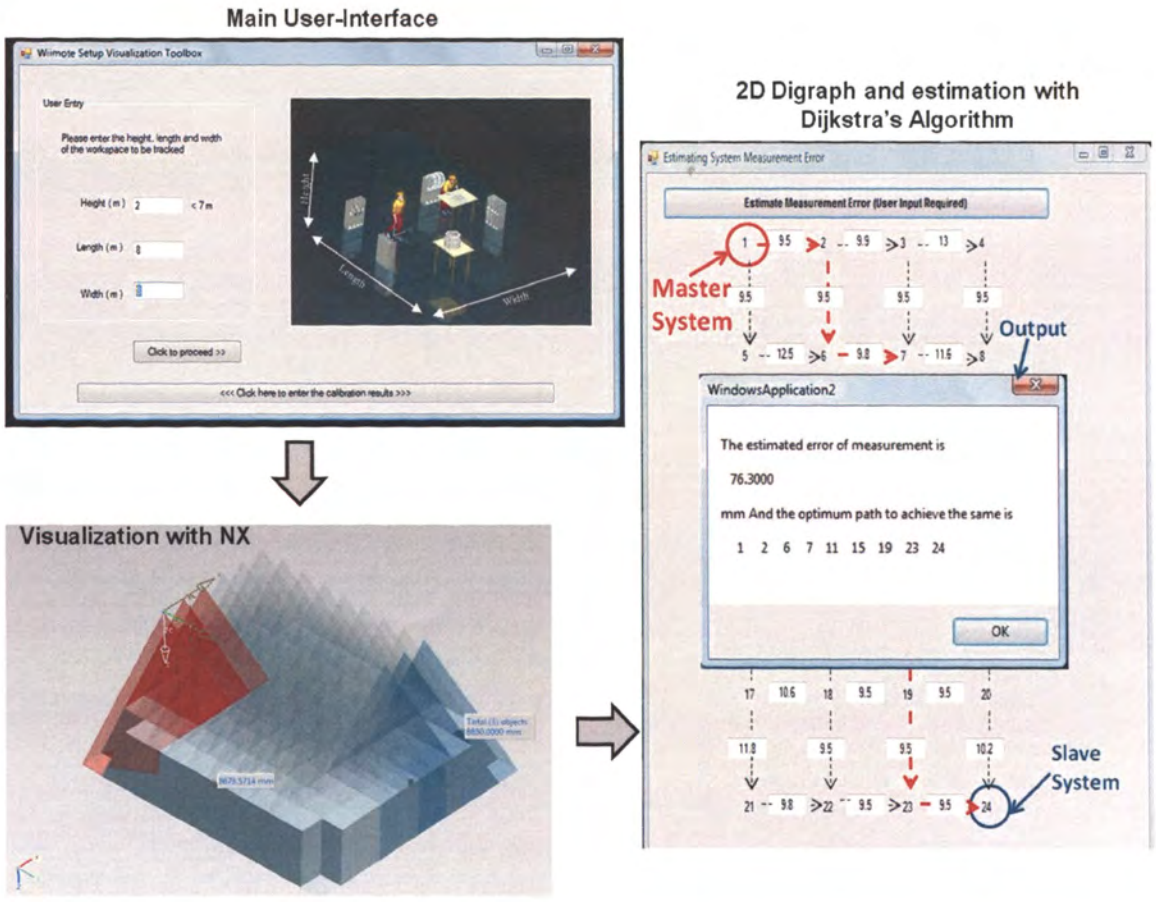


Figure 8.6 Implementation of Dijkstra's algorithm to estimate system measurement error

The above algorithm has also been extended to estimate the measurement error incurred at all intermediate individual vision systems in the entire network of vision systems.

8.3 ESTIMATING THE OPTIMUM TRANSFORMATIONS (R,T) FOR INDIVIDUAL DEPENDENT VISION SYSTEMS

As discussed above, in a 2D array of vision systems, every vision system is calibrated with respect to its immediate neighboring vision system. Thus to estimate the positions of the marker consistently with respect to the "Master" vision system's coordinate system throughout the entire tracking volume, it is necessary to estimate the

position and orientation of all intermediate vision systems with respect to the master vision system.

Consider an intermediate vision system #15 as shown in Figure 8.7.

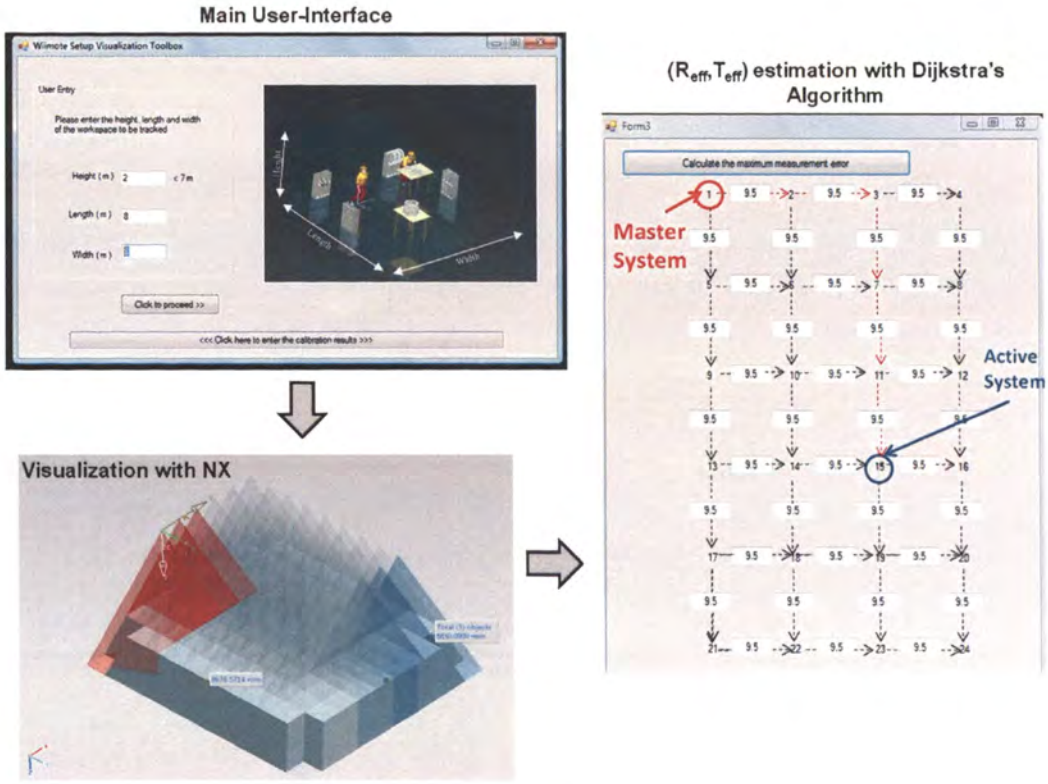


Figure 8.7 Estimating resultant transformations (R,T) for intermediate vision systems

Dijkstra's algorithm, as explained earlier, has also been implemented for all the intermediate vision systems inside the network. The system lets the user enter the physically measured errors between different vision systems as a result of the calibration process. The algorithm gives the most optimum path for the vision system under consideration. As indicated in Figure 8.6, the optimum path in this case happens to be 1-2-3-7-11-15. Thus, a chain of transformations gives the relationship between the "Slave"

system #15 with the "Master" vision system #1. The required transformation can be expressed as,

$$(R_{eff}, T_{eff}) = ((R_{11\text{-}15}^{-1} * R_{7\text{-}11}^{-1} * R_{3\text{-}7}^{-1} * R_{2\text{-}3}^{-1} * R_{1\text{-}2}^{-1}), (-R_{1\text{-}2}^{-1} * T_{1\text{-}2} - R_{2\text{-}3}^{-1} * T_{2\text{-}3} - R_{3\text{-}7}^{-1} * T_{3\text{-}7} - R_{7\text{-}11}^{-1} * T_{7\text{-}11} - R_{11\text{-}15}^{-1} * T_{11\text{-}15}))$$

Thus,

$$X_{global} = R_{eff} * X_{local} + T_{eff}$$

where $R_{i\text{-}j}$ indicates the rotation transformation between the coordinate systems of vision systems i and j, whereas $T_{i\text{-}j}$ indicates the translational relationship between the coordinate systems of vision systems i and j, and X_{local} and X_{global} being the 3D coordinates of the marker w.r.t. the global coordinate system of the tracking system and local coordinate system of the current active vision system, respectively.

8.4 AUTOMATION OF CAD ASSEMBLY GENERATION WITH SIEMENS NX5

Once the measurement error over the entire volume is estimated and is within acceptable limits for the user, the software generates a CAD display indicating the estimated positions of the Wiimotes in a virtual world.

Siemens NX offers inbuilt facilities of recording the events inside NX. These events can be simple modeling commands or certain measurement commands inside NX. For example, NX Journals allow the users to record a particular set of events inside NX, which can be played back later to generate the same set of commands or instances. NX Macros also offer the same capability, however Macros many a times end up requiring the manual user selection of particular features to carry out certain operations inside NX and thus cannot be run completely independently and almost always need some form of manual input to proceed at intermediate stages. Programming with NX Knowledge

Fusion (KF) also poses problems in terms of exchanging data with other application software, and thus has many constraints in terms of flexibility in programming.

Thus, NX Journals offer better flexibility in terms of programming with NX. NX Journals allow the use of NX element IDs as well as run time user inputs. NX Journal APIs also allow switching between different NX modules (Modeling Module, Assembly Module and Gateway) and thus are absolute playback tools inside NX. Thus the idea here is to develop a code to automatically generate such NX Journals by keeping track of numerous instances created for a particular type of operations inside NX, viz. generating line or arc, extrusion commands, Linker bodies, modeling mode, etc. Such custom NX Journal files are to be generated depending on the number of Wiimote models required in the CAD assembly. Such custom generated Journals can later simply be read inside NX to automatically generate the detailed CAD simulation of the entire Wiimote tracking system setup.

Considering the NX functions required in generating the CAD assembly of a Wiimote system, a generic algorithm and code is generated in VB .NET to generate such custom NX Journal files using NX APIs. As discussed earlier, such NX Journals can later be simply read inside NX to generate the CAD assemblies as indicated in Figure 8.7.

Figure 8.8 indicates the steps involved in automated CAD simulation generation of the Wiimote Vision System. Wiimote Navigation Toolbar inside NX allows the user to interact with the CAD assembly, reposition the Wiimotes in the virtual world, control the visibility of the components, measure the dimensions of the tracked volume etc. Part of the functionalities included in Wiimote Navigation Toolbar are coded with NX Macros and rest with NX Journals.

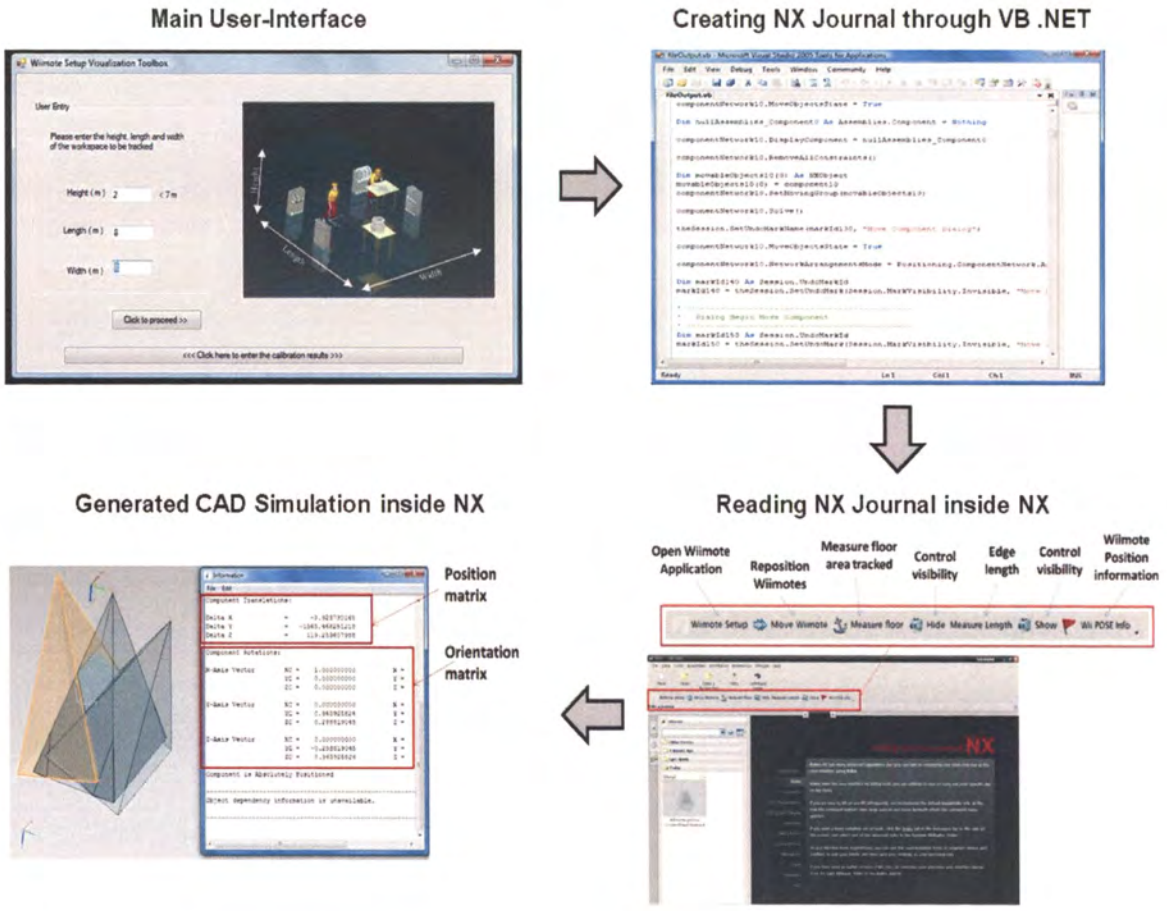


Figure 8.8 Automation of CAD simulation generation with Siemens NX5 Journals

9. APPLICATION TO ASSEMBLY SIMULATION

Based on the tracking data generated by the Wiimote vision system, CAD simulations can be directly generated by dynamically feeding position and orientation data to a standard CAD software, e.g. Siemens NX5. Figure 9.1 indicates the step-by-step data flow in a Wiimote tracking system. Siemens NX provides an Application Programming Interface (API) named NXOpen which provides interface for the user to communicate with NX models and perform various tasks through Microsoft's .NET framework. This API has been utilized to develop assembly simulations, where the user can dynamically perform play/pause, zoom, pan, rotate, and other operations and perform analysis on the simulated assembly process.

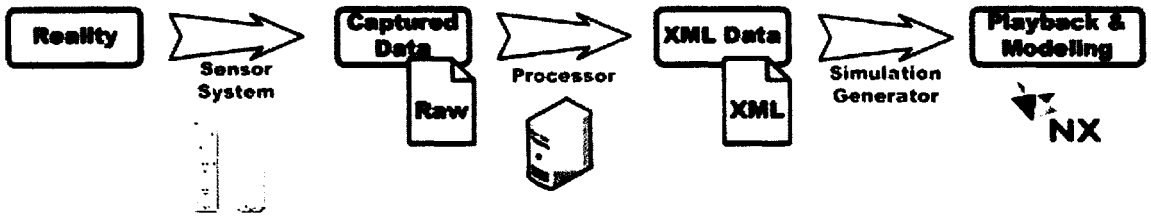


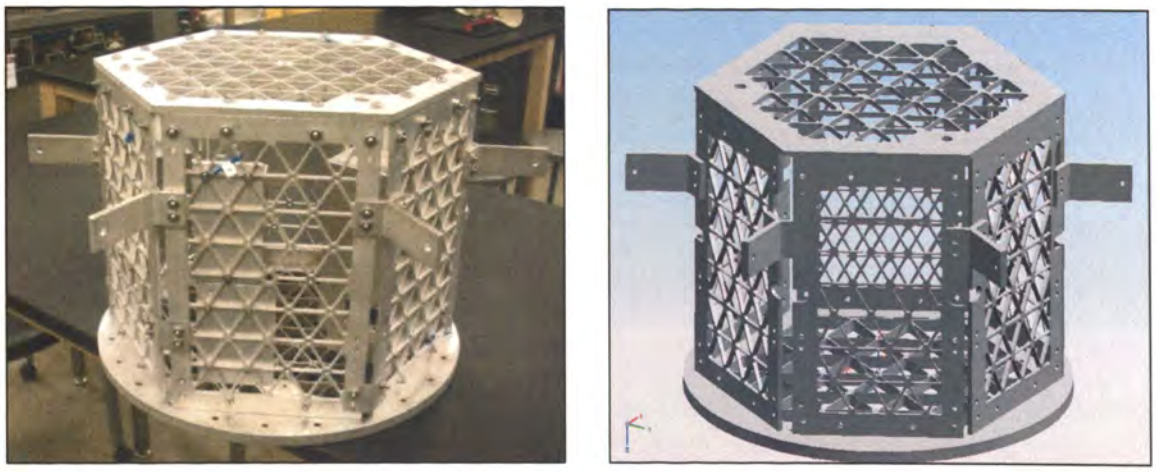
Figure 9.1 Information flow for the Wiimote tracking system

During the motion capture, the data from the Wiimote vision system is stored in an XML file, which contains information about the part being tracked, the timestamp and the coordinate details of the part. The playback animation speed can be adjusted according to the time elapsed in the simulation and the current timestamp of the data. The XML file data and the CAD models of the assembly are the inputs to the simulation

engine developed. Once the data is loaded, the user has various options, e.g. hide/ display of unused CAD models and selection of animation speed as well as ability to add more XML data files, if the assembly simulation is tracked in different stages.

9.1 MOTION TRACKING WITH A WIIMOTE VISION SYSTEM

For demonstration, a Wiimote based vision system has been implemented for tracking the assembly operation of a MSAT (Mobile Satellite) prototype frame shown in Figure 9.2. Here, as the total tracking area required is (1m x 1m) a single stereo vision system with 2 Wiimotes is sufficient to cover the entire tracking volume up to a height of 3m above the floor. CAD models were generated in NX for various parts of the physical prototype.



(a) physical prototype (b) CAD assembly model

Figure 9.2 The MSAT prototype frame

The locations of the LEDs mounted on the various parts of the physical prototype were decided beforehand. The local coordinate system in the virtual world was set at the

same location as the LED marker on the physical part. Points of symmetry were preferably chosen in mounting the LEDs, considering the ease of locating them on the physical parts and specifying the corresponding coordinate frames on the CAD models. Attention was paid to ensure that the marker was exposed to the tracking system continuously while carrying out the entire assembly process. The LEDs used in the motion tracking were powered by button cells. The small size of these cells offers easy mounting of the LEDs on the parts of the MSAT prototype. Figure 9.3 shows the corresponding sensor locations on the CAD models for various parts of MSAT assembly.

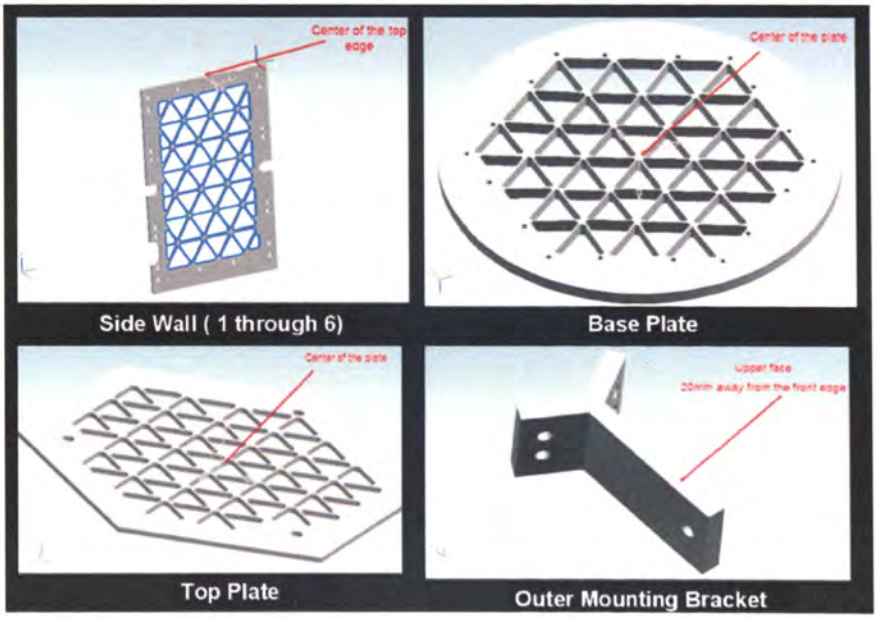


Figure 9.3 LED locations for several parts of the MSAT prototype

Figure 9.4 depicts the motion capture and the generation of CAD model based simulation in real time during the MSAT assembly process.

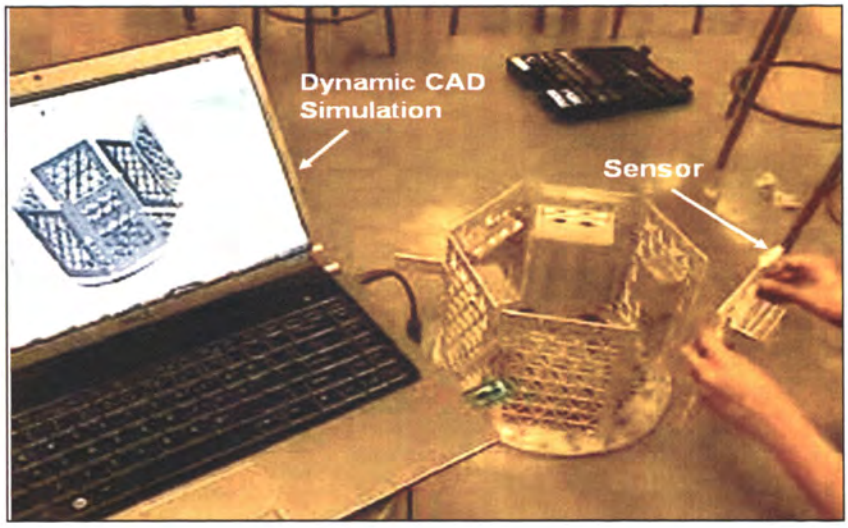
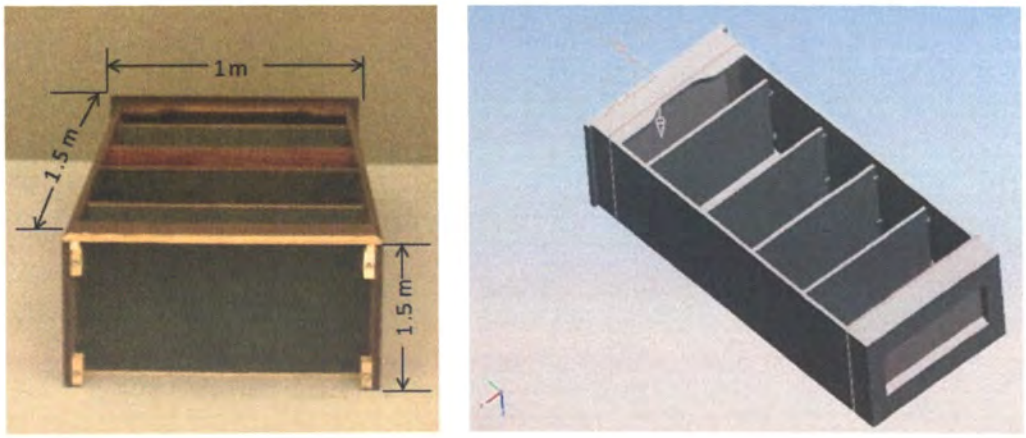


Figure 9.4 Motion capture and the generated CAD simulation in real-time

9.2 IMPLEMENTATION WITH AN INTEGRATED TWO VISION SYSTEM

This implementation example involves the use of an integrated two vision system with four Wiimote cameras for tracking the assembly of a bookshelf. The assembly of a wooden bookshelf (1 m x 1.5 m x 0.5 m), as shown in Figure 9.5 was tracked. CAD models were generated after taking measurements on the physical bookshelf. Figure 9.5 indicates the actual bookshelf and the CAD assembly model generated.



(a) The physical bookshelf (b) The CAD assembly model

Figure 9.5 The bookshelf used for demonstration with the two stereo vision systems

Figure 9.5 shows the motion tracking experimental setup, where four Wiimotes were mounted on the ceiling to form two stereo vision systems, which were integrated into one system for motion tracking. The four Wiimote were located so as to ensure coverage of the entire volume within which the assembly took place. The world coordinate system was defined with the help of an L-shape wand as shown in Figure 9.6.

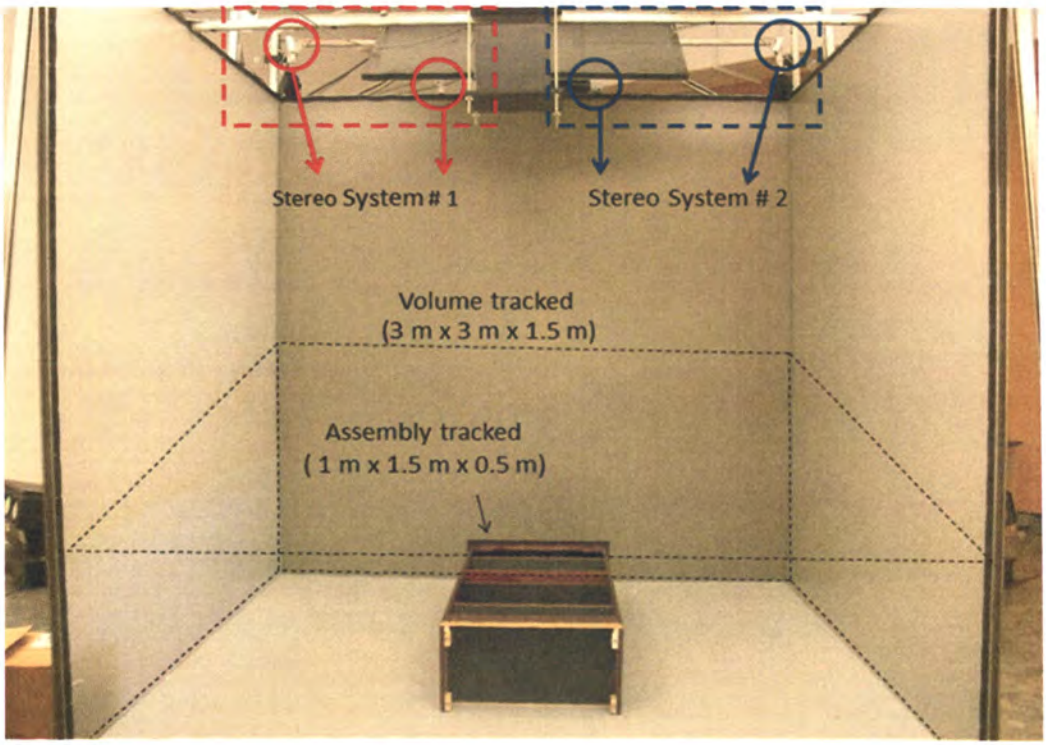


Figure 9.6 The experimental setup for motion tracking for the bookshelf assembly

All 6 DOF of the parts were tracked with the help of the wand mounted with 4 IR LEDs. The locations for mounting the wand on the various parts of the bookshelf were decided beforehand. Figure 9.7 shows the locations of the wand for the parts to be tracked as specified on the CAD models.

Figure 9.8 displays different stages of the CAD assembly simulation within NX5.

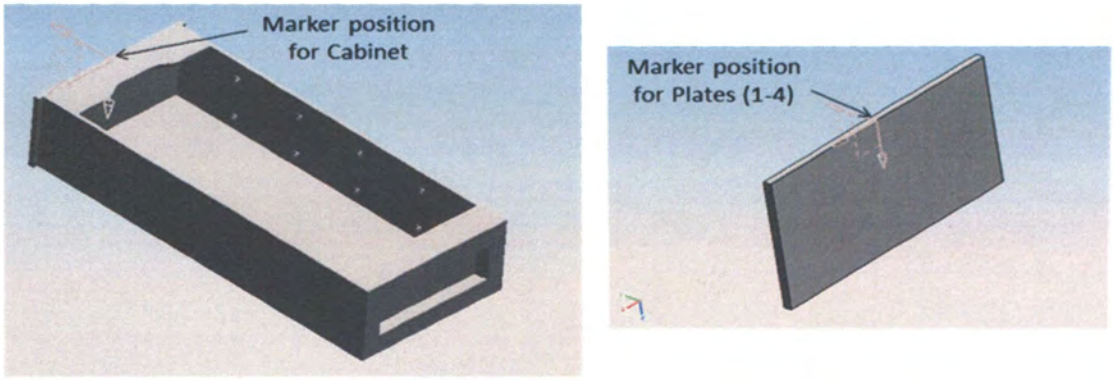
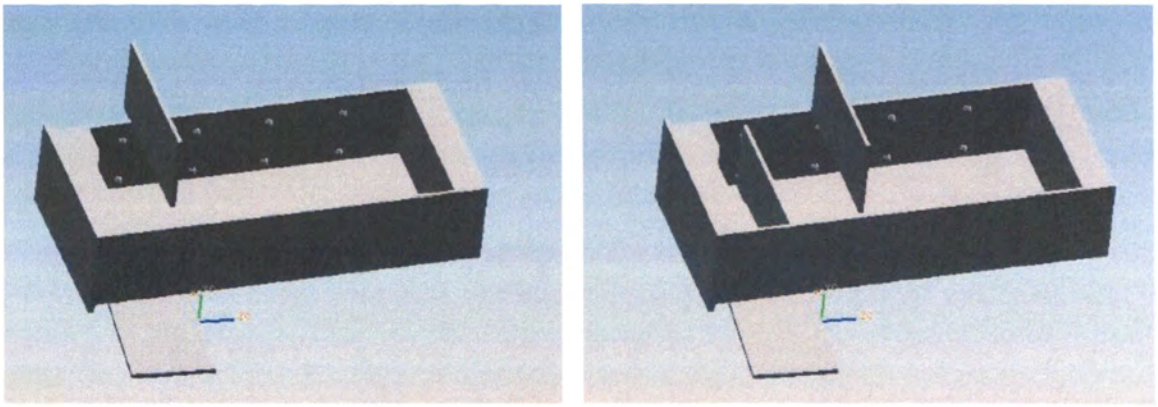
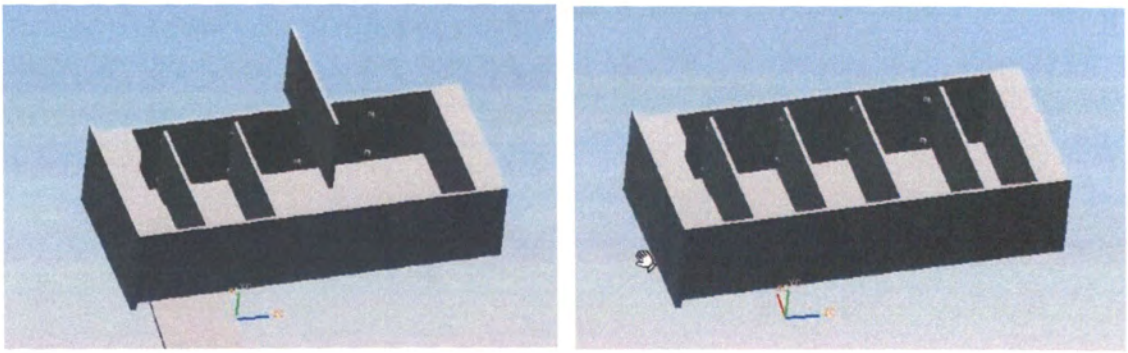


Figure 9.7 Locations of the LED marker for tracking different parts of the bookshelf



(a) (b)



(c) (d)

Figure 9.8 Snapshots of the bookshelf assembly simulation

10. CONCLUSIONS

The feasibility in forming vision systems using Nintendo Wii Remotes for complete 6 DOF motion tracking with good accuracy is proved in this thesis study. Such systems are much cheaper than commercially available motion tracking systems, thus providing great economic benefits.

A new camera calibration technique has been developed by integrating Zhang's and Svoboda's algorithms. The basic idea of this integrated calibration technique is first using Zhang's algorithm to determine the intrinsic parameters and then using Svoboda's algorithm to determine extrinsic parameters. The integrated calibration technique is further used with Horn's algorithm for integration of two or more multi-camera systems into a single vision system to increase measurement volume. The calibration technique is evaluated for different arrangements of the Wiimotes and over different tracking volumes.

The distributed vision system architecture implemented hereby, along with inexpensive cameras enables the creation of a low-cost, versatile motion tracking system which is wireless in communication. With the Wiimote based motion tracking system, it is proved hereby that the simulation of an assembly operation can be generated automatically in real time with fairly good accuracy. Since the physical objects in the assembly simulation are represented by CAD models, the assembly simulation is capable of taking advantage of all the built-in viewing utilities such as pan, zoom, rotate etc. available from a commercial CAD system such as Siemens NX5. The CAD model based simulation tool with real-time motion capture offers the user a low-cost simulation tool to understand the assembly techniques and sequences by providing detailed simulations of

assembly operations with animated movements of tools and parts. This is helpful in understanding the assembly sequences and valuable for training of assembly operations, improvement of existing assembly tasks, and reuse of software to plan new assembly tasks with similar features.

Such Wiimote based tracking systems are much cheaper than most of the commercially available motion tracking systems. However, Wiimote based systems pose limitations in terms of maximum numbers of markers that can be tracked for a given instance. With the existing calibration technique, the accuracy offered by such systems also deteriorates considerably over the large tracking areas. Wiimote based systems also inherit all the limitations of a typical IR based tracking systems, viz. line of sight problem etc. Thus, Wiimote based systems suit more for a particular class of applications wherein the tracking needs to be done over the large areas, the tracking systems needs to be portable, number of objects to be tracked at the same time is less and the tracking accuracy required is a few millimeters.

BIBLIOGRAPHY

[1] P. Baggett and A. Ehrefeucht, "Building Physical and Mental Models in Assembly Tasks," International Journal of Industrial Ergonomics, 7(3), pp. 217-227, 1991.

[2] D. Kibira and C. McLean, "Virtual Reality Simulation of a Mechanical Assembly Production Line," Proceedings of Winter Simulation Conference, pp. 1130-1137, 2002.

[3] Lee, J., "Head Tracking for Desktop VR Displays Using the Wii Remote," http://johnnylee.net/projects/wii/, 2007.

[4] S. Hay, J. Newman, and R. Harley, "Optical Tracking Using Commodity Hardware," Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), 2008.

[5] I. Kitahara, H. Saito, S. Akimichi, T. Onno, Y. Ohta, and T. Kanade, "Large–scale Virtualized Reality," Computer Vision and Pattern Recognition, Technical Sketches, June 2001.

[6] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol. 3, No. 4, 1987.

[7] J. Heikkila and O. Silven, "A Four-step Camera Calibration Procedure with Implicit Image Correction," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000.

[8] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D-machine Vision," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 364-374, Miami Beach, Florida, 1986.

[9] Zhang, Z., "A Flexible New Technique for Camera Calibration," IEEE Transactions on Pattern and Machine Intelligence, 22(11), pp. 1330-1334, 2000.

[10] T. Melen, "Geometrical Modeling and Calibration of Video Cameras for Underwater Navigation," Ph.D. Thesis, Institutt for Teknisk Kybernetikk, Norges Tekniske Hogskole, 1994.

[11] H. Zollner, R. Sablatnig, "Comparison of Methods for Geometric Camera Calibration using Planar Calibration Targets," Proceedings of 28th Workshop Austrian Association of Pattern Recognition, pp. 234-244, 2004.

[12] T. Svoboda, D.Martinec, and T.Pajdla, "A Convenient Multi-camera Self-calibration for Virtual Environments," Teleoperators and Virtual Environments, Vol. 14, No.4, 2005.

[13] B. K. P. Horn, "Closed-form Solution of Absolute Orientation Using Unit Quaternions," Journal of the Optical Society of America A, Vol. 4, pp. 629, 1987.

[14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", The MIT Press, 2008.

[15] J. Y. Bouquet, "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2008.

[16] Pintaric, T. and Kaufmann , H., "Affordable Infrared-Optical Pose-Tracking for Virtual and Augmented Reality," Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop, Charlotte, NC, 2007.

[17] M. Ribo, "State of the Art Report on Optical Tracking," Technical Report VRVis 2001-25, TU Wien, 2001.

[18] C. Wengert and G. Bianchi, "Implementation of Closed-form Solution of Absolute Orientation using Unit Quaternions", http://www.mathworks.com/matlabcentral/fileexchange/22422-absolute-orientation,, as accessed on July 29, 2010.

[19] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, pp. 381–395, 1981.

[20] D. Martinec and T. Pajdla, "Structure from Many Perspective Images with Occlusions," Proceedings of the European Conference on Computer Vision, Volume II, pp. 355–369, Berlin, Germany, 2002.

[21] P. Sturm and B. Triggs, "A Factorization Based Algorithm for Multi-image Projective Structure and Motion," European Conference on Computer Vision, Volume II, pp. 709–720, 1996.

[22] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2003.

LIST OF PROJECT PUBLICATIONS

1. W. Zhu, A. Chadda, A. Vader, M. Leu and F. Liu, "Low-Cost Versatile Motion Tracking for Assembly Simulation," International Symposium on Flexible Automation, Tokyo, Japan, July 12-14, 2010.

2. A. Vader, A. Chadda, W. Zhu, M. Leu and F. Liu, "An Integrated Calibration Technique for Multi-Camera Vision Systems," ASME World Conference on Innovative Virtual Reality, Ames, Iowa, May 12-14, 2010.

APPENDIX

Steps in calibrating a Wiimote Vision System are listed below:

- Estimate the number of Wiimotes required for having the required coverage volume and area with Wiimote Vision System Setup Toolkit as discussed in Section 8.
- Calibrate all Wiimotes to be used for intrinsic parameters with 36 LED calibration plate as discussed in Section 3. Collect the calibration data using the Camera Calibration program for different positions of the 36 LED calibration plate in 3D space. Make sure that the calibration data is collected over the entire Field of View of the individual Wiimotes. Experiments have shown that 25 sets of observations provide consistently good estimation of the intrinsic parameters for Wiimotes.
- Copy the above calibration data in \\Camera_Calibration_Toolbox\C1.m and \\Camera_Calibration_Toolbox\C2.m for left and right Wiimotes respectively. Run both the mentioned files manually to solve for camera intrinsic parameters. These intrinsic parameter results are further modified during combined stereo calibration for 2 Wiimotes. To calibrate them as a stereo system, run the MATLAB function \\Camera_Calibration_Toolbox\calib_stereo.m All the calibration results are written in the file \\Camera_Calibration_Toolbox\Calubration_Results_stero.mat Repeat this step for all the Wiimotes to be used for setting up camera vision systems.
- To calibrate the Wiimotes for extrinsic parameters, mount the Wiimotes as described by Wiimote Vision System Setup Toolkit and collect the single point

calibration data with Camera Calibration program. Note that the program records the calibration data only when the IR LED is seen by all the Wiimotes contributing to the concerned vision system as discussed in Section 5. Make sure that the calibration data is collected over the entire Field of View of the vision system. Experiments have shown that 1500 observations provide consistently good estimation of the extrinsic parameters for Wiimotes.

- Copy the above calibration data in \\Extrinsic_Calibration\TestDataOneSide\Points.m file in MATLAB. Mention the correspondences between calibration points in the MATLAB file \\Extrinsic_Calibration\TestDataOneSide\IdMat.m. Mention the number of Wiimotes being calibrated together in the configuration file \\Extrinsic_Calibratio\BlueCCal\CommonCfgAndIO\configdata.m . To solve for extrinsic parameters run \\Extrinsic_Calibration\ \BlueCCal\MultiCamSelfCal\gocal.m in MATLAB. The camera calibration results are written in \\Extrinsic_Calibration\TestDataOneSide\Pmatrices.dat file.
- To Split the individual camera Euclidean matrices into intrinsic and extrinsic parameters use MATLAB function \\Extrinnsic_Calibration\BlueCCal\ MultiCamValidation\CoreFunctions\P2KRtC.m
- Replace the value of intrinsic parameter matrix (K) in MATLAB as obtained with 36 LED calibration plate. To formulate the Euclidean matrix with new intrinsic parameters use the MATLAB function \\Extrinnsic_Calibration\BlueCCal\MultiCamValidation\CoreFunctions\DownIsUp.m

- Copy the above new calibration results in \\Calibration_Program\Bin\Debug\pMat.dat. Select the appropriate number of IR LEDs being used for tracking in Camera Calibration program. As discussed in Section 7, only 1 IR LED is required for 3 DOF tracking, whereas for complete 6 DOF a wand with 4 IR LEDs is used. Click on Motion Capture to start the tracking.

VITA

The author, Anup Madhav Vader, was born on November 26th, 1984 in India. He received his degree of Bachelor of Engineering in Mechanical Engineering from Walchand College of Engineering, India in May 2006. He was awarded numerous departmental and national scholarships for his academic achievements during his under-graduate work including the prestigious National Merit Scholarship (Government of India, India) (2000-2002), State Mathematics Scholarship (Government of Maharashtra, India) (2000) and Department of Mechanical Engineering Scholarships (Walchand College of Engineering, India) (2002-2004). His undergraduate research was mainly focused on CAD software development and CAD product customization, for automation of the design processes in mechanical assemblies.

The author worked with India's largest automobile manufacturing company, TATA Motors Ltd, as Assistant Manager (Engineering Research Center) from July 2006 to July 2008.

In August 2008, he joined the Master of Science program in Mechanical Engineering at Missouri University of Science and Technology, Rolla, MO. His research work with Dr. Ming C. Leu at Missouri S&T is mainly focused on developing affordable motion capture systems and its integration with commercial CAD softwares. The author received his Master of Science degree in Mechanical Engineering from Missouri S&T in Dec 2010. He has been a member of Tau Beta Pi - The Engineering Honor Society and American Society of Mechanical Engineers (ASME) since August 2008.

Currently, the author is working as Associate Design Engineer with Caterpillar Inc., Mossville, IL.