Three-Dimensional Hand Tracking and Surface-Geometry Measurement for a Robot-Vision System

by

Chris Yu-liang Liu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2008

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Tracking of human motion and object identification and recognition are important in many applications including motion capture for human-machine interaction systems. This research is part of a global project to enable a service robot to recognize new objects and perform different object-related tasks based on task guidance and demonstration provided by a general user. This research consists of the calibration and testing of two vision systems which are part of a robot-vision system. First, real-time tracking of a human hand is achieved using images acquired from three calibrated synchronized cameras. Hand pose is determined from the positions of physical markers and input to the robot system in real-time. Second, a multi-line laser camera range sensor is designed, calibrated, and mounted on a robot end-effector to provide three-dimensional (3D) geometry information about objects in the robot environment. The laser-camera sensor includes two cameras to provide stereo vision. For the 3D hand tracking, a novel score-based hand tracking scheme is presented employing dynamic multi-threshold marker detection, a stereo camera-pair utilization scheme, marker matching and labeling using epipolar geometry and hand pose axis analysis, to enable real-time hand tracking under occlusion and non-uniform lighting environments. For surface-geometry measurement using the multi-line laser range sensor, two different approaches are analyzed for two-dimensional (2D) to 3D coordinate mapping, using Bezier surface fitting and neural networks, respectively. The neural-network approach was found to be a more viable approach for surface-geometry measurement worth future exploration for its lower magnitude of 3D reconstruction error and consistency over different regions of the object space.

Acknowledgements

The research reported in this paper was carried out at the Intelligent Human-Machine Systems Laboratory, Department of Systems Design Engineering, University of Waterloo, Waterloo ON, Canada. This research has been financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The author is grateful for the help of colleagues Xianghai Wu and Kiatchai Borribanbunpotkat in experiments. The author would like to thank his supervisor Dr. Jonathan Kofman for his invaluable guidance for problem solving to make an enjoyable academic life at University of Waterloo. The author would also like to thank the readers for their input of helpful comments.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Global Research and Project Scope

The ever increasing cost of labour and maturing of machine intelligence have increased interest in service robots for the general user. The ultimate purpose of a service robot is to have different types of tasks performed at different times and in various environments for users. It has always been desirable that robots can be taught interactively by an average user and increase knowledge accordingly through watching user's demonstrations and recognizing new objects via built-in sensors. The ultimate goal of the global research is to build an adaptive human-robot teaching and learning system that will enable general users to teach a robot different object-related tasks in a natural, intuitive, and interactive manner.

As part of the global research, the research presented in this thesis consists of two major components, namely three-dimensional hand tracking and surface-geometry measurement for a robot-vision system. With both components enabled, a human operator should be able to teach a robot by teleoperating the robot to perform object manipulation tasks, and instruct the robot to learn about an object by measuring its surface-geometric features. The following two sections briefly discuss these two components.

This thesis consists of five chapters. Chapter 1 discusses the global research and the scope of the proposed project, and gives a brief introduction to its two major components, which are the 3D hand tracking and object surface-geometry measurement; Chapter 2 provides background knowledge used in this project, and covers both single and stereo camera calibration and calibration experiments, 3D point reconstruction from stereo vision, point matching using epipolar geometry, Bezier surface fitting, and neural networks. The system design, detailed system implementation and related algorithms, and experiments are described for the 3D hand trackingin Chapter 3 and for the object geometry measurement in Chapter 4. Concluding remarks are given in Chapter 5.

1.2 Three-Dimensional Hand Tracking

In unstructured and dynamic environments, robots may not be able to perform complex tasks fully autonomously. Robot teleoperation using critical decision making by the human operator would be required, especially in dangerous environments. Human teleoperation can also be used in *human-guided robot learning*. In *robot learning by demonstration,* a human

operator would demonstrate a complex task to the robot through teleoperation, and the robot would observe the required related hand motions and sub-tasks to perform the task. The key part of the teleoperation is the tracking of the hand *pose* (position and orientation) of the human teacher.

Contacting mechanical devices such as robot replicas, dials, and joysticks (Postigo et al. 2000) have been used as means of teleoperation; however, such devices usually demand unnatural hand arm movements in controlling the motions of the robot. Other mechanical methods include devices worn by an operator, such as exoskeletal mechanical devices (Chang et al. 1999), gloves instrumented with angle sensors (Harada et al. 2000; Hu et al. 2005; Tezuka et al. 1994), electromagnetic (Bachmann et al. 2001; Perie et al. 2002), inertial (Bachmann et al. 2001; Verplaetse 1996) motion tracking sensors, and electromyographic sensors (Fukuda et al. 2003). However, these contacting devices may hinder dexterous human motion. Vision-based body tracking techniques (Chaudhari et al. 2001; Kofman et al. 2005; Lathuiliere and Herve 2000) include marker-based methods (Borghese and Rigiroli 2002; Boult et al. 1998; Hugli et al. 1992; Kofman et al. 2005; Peters and O'Sullivan 2002; Tieche et al. 1996) and markerless methods (Verma 2004; Kofman et al. 2007), which benefit from having no physical contacts between the human operator and the robot. These vision-based methods are less invasive, allow more freedom in moving the controlling hand and arm, and support capturing human motion without the need for active sensors.

Tracking of markers placed on the hand has been successful in permitting free operator hand and arm motion to control a robot (Kofman et al. 2005). In this approach, a stereo camera pair was used to track the human hand-arm motion and obtain two-dimensional (2D) image coordinates of the hand and arm of the human operator at the local operator site. A remote robot-site computer then performed 3D reconstruction of the hand pose in the local-site stereo-camera coordinate system based on the 2D human hand-arm position using a stereo-camera calibration performed at the local-operator site. Communication between the local operator-site and remote robot-site computers was performed over a local area network (LAN). This was a significant contribution to vision-based and vision-guidance based teleoperation, in demonstrating effective communication of tasks to a robot manipulator in a natural manner, using the same hand motions that would normally be used for a task. However, there were limitations in the practical use of the vision system due to the occlusion or hiding of markers.

The 3D hand-tracking component of this thesis improves the above marker-based vision system by introducing a third camera to the system to form a trinocular vision system. Furthermore, a new marker labeling scheme such that minimizes occlusion of markers and increases the reliability of the overall system is developed and implemented. Various enhancements have also been proposed such that the new system is able to deal with different environmental issues such as non-uniform lighting. Furthermore, the entire tracking system was rebuilt and algorithms redeveloped to enable hand tracking in real-time at video rate, as an improvement from the previous system. The entire robot teleoperation system was also reconfigured in a new laboratory at the University of Waterloo. Details of the system design of this proposed approach are discussed in Chapter 3.

1.3 Three-Dimensional Surface-Geometry Measurement

The second main component of this thesis is the design and construction of a multi-line laser and stereo-camera scanning system to measure the 3D object surface geometry of objects. This full-field range sensing system serves as an integral part of the robot-vision system to be used for object recognition and identification in the global research project.

Measurement of 3D surface object geometry is used for reverse engineering, object inspection, and object and scene modeling. Techniques include contact coordinate-measuring machines (CMMs), stereo imaging, and noncontact range sensors. Chen et al. (2000) provided an extensive overview of these techniques. Three dimensional measurement using structure light has been one of the most common approaches as it is non-contacting and can be applied where physical contact is not permitted or possible. A shape reconstruction algorithm (Knopf et al. 2002) using artificial intelligence has been proposed. This method employs a Bernstein basis function network to utilize the capacity of neural networks to generalize a learned pattern, to simplify the mapping between image and object coordinates. However, the 3D surface geometry measurement of an entire surface by scanning a single projected line over the surface is time consuming, and it required a physical means to scan the range-sensor head over the surface. A method of acquiring the full surface geometry of an object without scanning over the surface would be preferred.

A multi-line full-field laser-camera range sensor (Kofman et al. 2007) was proposed to address the limitations of single-line scanning. A full-field laser projector simultaneously projects multiple laser lines onto the object such that the whole object surface can be measured by a single image or pair of images, captured by one or two cameras, respectively, at fixed positions. Such techniques are called "*one shot*" techniques. A camera pair is employed to handle potential occlusion of the object surface from different viewpoints. The mapping of 2D image plane coordinates to 3D object space coordinates is carried out for each of the laser projections by a closed-form Bezier surface-fitting process. This approach greatly simplifies the calibration process as well as the object measurement, such that real-time 3D surface geometry measurement is possible. However, due to the approximation used in Bezier surface-fitting, higher errors occur closer to the centre of the calibrated volume, and a more reliable 2D to 3D mapping technique is desirable.

The aims of the surface-geometry measurement component of this research project are to: design and build a new robot-mounted multiple-line range-sensor, calibrate the range-sensor using the previous Bezier surface-fitting approach to perform the mapping of 2D image plane coordinates to 3D object coordinates, introduce a new range-sensor calibration approach using a direct 2D to 3D image-to-object coordinate mapping using a multi-layer perceptron (MLP) neural network, and compare the calibration accuracy of the two calibration techniques. The neural network method is expected to permit a fast calibration process and improve calibration and surface measurement accuracy. Details of surface-geometry system

design and experiments of both Bezier surface fitting and neural network approaches are discussed in Chapter 4.

1.4 Summary of Research Objectives

The objectives of the research are summarized as follows:

Three-dimensional hand tracking and robot vision

- Improve the marker-based 3D hand-tracking system by introducing a third camera to form a trinocular vision system.
- Implement a new marker labeling scheme to minimize occlusion of markers and increase the reliability of the overall system.
- Enhance the new system to deal with environmental issues such as non-uniform lighting.
- Rebuild the entire tracking system and redevelop algorithms to enable *hand* tracking in real-time at video rate, as an improvement from the previous system.
- Setup and reconfigure the entire robot teleoperation system for the new laboratory setup.

Surface-geometry measurement

- Design and build a new robot-mounted multiple-line range-sensor.
- Calibrate the range-sensor using the previous Bezier surface-fitting approach to perform the mapping of 2D image plane coordinates to 3D object coordinates
- Introduce a new range-sensor calibration approach using a direct 2D to 3D image-to-object coordinate mapping using a multi-layer perceptron (MLP) neural network
- Compare the calibration accuracy of the two calibration techniques.

Chapter 2

Background Knowledge and Preliminary Experiments

2.1 Camera Calibration

2.1.1 Camera Calibration Overview

The purpose of camera calibration is to establish the projection from the 3D world coordinates to 2D image coordinates. Once this projection is known, 3D information can be inferred from the 2D information, and vice versa. In addition, with a pair of calibrated stereo cameras, 3D coordinates of a world point can be computed using only one image frame, provided images of the world point are available in both cameras' image planes. This section covers single camera calibration, stereo camera calibration, and 3D coordinate reconstruction using stereo vision. All of the cameras used in this research project are calibrated following the same procedures. In addition to calibration of individual cameras, the trinocular vision used in the 3D hand tracking system is also calibrated using camera pairs: left-mid pair, mid-right pair, and left-right pair; since the surface-geometry system uses two cameras, they form only one stereo pair. Calibration results for both systems are provided at the end of the section.

2.1.2 Single Camera Calibration

Camera calibration is a prerequisite for any application where the relation between 2D images and the 3D world is needed. This mapping of 3D to 2D, is called *perspective projection*, and is fundamental to image analysis. Camera calibration can be modeled as an optimization process, where the discrepancy between the observed image features and their theoretical positions is minimized with respect to the camera's *intrinsic* and *extrinsic parameters* (Forsyth et al. 2003). The process of camera calibration is to recover the two sets of parameters of a camera. Intrinsic parameters determine the projective behavior of the camera while extrinsic parameters determine the mapping from the world space to the camera space.

Intrinsic parameters are also called *camera parameters* because they do not depend of the scenes viewed or where the camera is used. Intrinsic parameters include: the principle point, effective focal length, skew factor, and distortion coefficients. The principle point is the point where the optical axis coincides with the image plane (image sensor plane), and usually not the centre of the image sensor plane due to manufacturing errors. The effective focal length f has two values f_x and f_y for the focal lengths along the *x*-axis and *y*-axis respectively. The skew factor is the angle between the *x*-axis pixels and *y*-axis pixels caused by the imperfection in the manufacturing process. Distortion is due to the optical deflection introduced by the lens of a camera. It includes radial distortion and tangential distortion (Tsai 1986). Extrinsic parameters include three rotation parameters and three translation

parameters. Rotation parameters relate the orientation of the camera coordinate system to the world reference coordinate system, and translation parameters map the origin of the camera coordinate system to the world coordinate system. These six parameters follow the law of *rigid motion transformation* in space with *six degrees of freedom*.

A detailed review of the common camera models can be found in (Clarke et al. 1998). The *pinhole model* (figure 2.1) is most widely used. Various pinhole model-based algorithms for camera calibration have been reported over the years in the photogrammetry and computer vision literature (Heikkila 1997; Triggs 1998; Zhang 2000). In the pinhole model, the camera is assumed to perform a perfect perspective optical transformation.

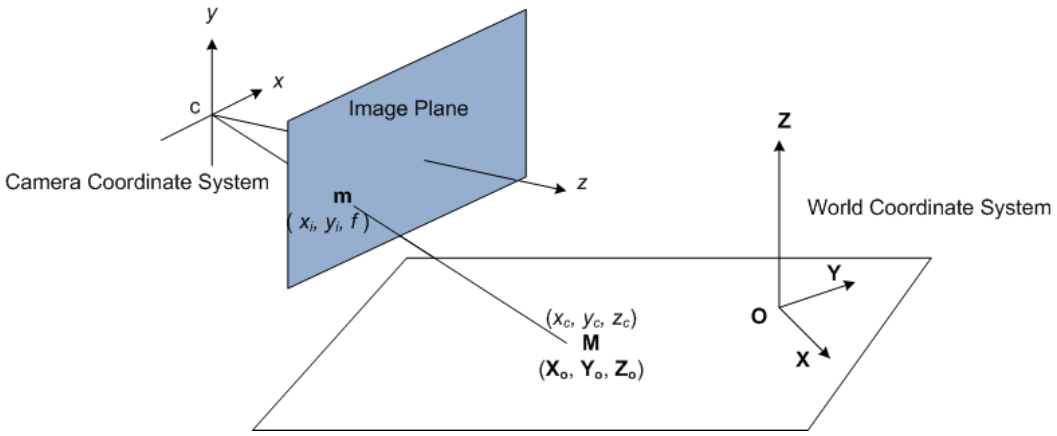

Fig. 2.1 Perspective geometry of pinhole model. Point M is a world point with coordinates (X_O, Y_O, Z_O) in the world coordinate system and (x_c, y_c, z_c) in the camera coordinate system. Point m is the image of world point M in the camera image plane with coordinates (x_i, y_i, f) in the camera coordinate system. Note that the image plane is placed before the projection centre for better visualization.

The overall mapping from image to world coordinates can be written as (Zhang 2000):

$$sm = A[R|T]M \tag{2.1a}$$

or

$$s\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1b}$$

(X, Y, Z) are coordinates of a 3D point in the world coordinate system, and (i, j) are coordinates of the point projection onto the image in pixels. Note that the image plane is placed before the projection centre for better visualization, but it is actually behind. A is called the *camera matrix*. (c_x, c_y) is the principle point and (f_x, f_y) are focal lengths

expressed in pixel-related units. Note that f_x, f_y, c_x, and c_y are scaled by the same factor $1/s$, where s is a positive real value that varies for different scenes. The joint rotation-translation matrix $[R|T]$ is called the *extrinsic matrix* and is used to describe the camera motion around a static scene: $[R|T]$ transforms coordinates of a point (X, Y, Z) in space to a coordinate system, fixed with respect to the camera, where R is a rotation matrix, and T is a translation vector. The transformation above is equivalent to the following when $Z \neq 0$ (OpenCV 2006):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T \tag{2.2}$$

$$x' = \frac{x}{z} \quad y' = \frac{y}{z} \tag{2.3}$$

$$x'' = x'(1 + k_1 r^2 + k_2 r^4) + 2p_1 x' y' + p_2(r^2 + 2x'^2), \tag{2.4a}$$
$$y'' = y'(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y'^2) + 2p_2 x' y', \tag{2.4b}$$

where

$$r^2 = x'^2 + y'^2 \tag{2.5}$$

$$i = f_x x' + c_x, \;\; j = f_y y' + c_y \tag{2.6}$$

Lens distortion is modeled with parameters k_1, k_2 for radial distortion coefficients and p_1, p_2 for tangential distortion. Tsai (1986) points out that for industrial machine vision applications, only radial distortion needs to be considered and only one term is needed.

A calibration technique is usually based on known space coordinates of geometrically configured points (calibration points) which are physically realized by marks on a calibration object. A calibration object can be a 3D object (Tsai 1987; Faugeras 1993) or 2D (Caprile et al. 1990; Liebowitz et al. 1998; Heikkila 1997) or planar (Zhang 2000). The calibration method used by Zhang is a 2D planar method that requires a planar calibration grid to be placed at different poses in front of the camera. This approach is used in this project for its ease of setup and cost effectiveness. The algorithm uses the extracted corner points of a checkerboard pattern to compute a projective transformation between the image points of different images and the corresponding 3D points. Afterwards, the camera intrinsic and extrinsic parameters are recovered using a closed-form direct linear transformation solution, while radial distortion terms are recovered by a linear least-squares solution. A final non-linear minimization of the re-projection errors is solved using a Levenberg-Marquardt method (Levenberg, 1944; Marquardt 1963), which refines all the recovered parameters. This approach requires at least five non-parallel views of a planar scene.

2.1.3 Stereo-Camera Calibration

The purpose of using stereo vision is to extract depth information of a scene using two different points of view, and only one frame is necessary. With a single calibrated camera, the depth information of a scene can only be computed using two or more frames. Depth extraction (described in Section 2.1.4 Three-Dimensional Reconstruction by Triangulation)

using a stereo pair of images requires the knowledge of the relative positions and orientations of the two cameras. This includes both the relative translation and rotation of the rigid motion transformation in space. Stereo camera calibration refers to the process of determining the relative rotation and translation of a stereo camera pair.

Rigid motion can be determined given two sets of points matched between two images. For an arbitrary point in space, $P = (X, Y, Z)^T$ is the spatial vector of the point before motion; $P' = (X', Y', Z')^T$ is the spatial vector of the point after motion; $p = (x, y, 1)^T = P/Z$; and $p' = (x', y', 1)^T = P'/Z'$ is the image vector of the point after the motion. The rigid motion can be decomposed into rotation R and translation T and is described by the following equation:

$$P' = RP + T \tag{2.7}$$

and can be rearranged (Faugeras et al. 1990) as:

$$p'^T E p = 0, \tag{2.8}$$

where $E = [\mathrm{T}]_\times R$, with $[T]_\times = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$. The matrix E is called the *essential matrix*. Details of properties of this matrix can be found in Section 2.2. The essential matrix has five degrees of freedom, hence at least five matched points are necessary to estimate it. Given n point correspondences, Equation (2.8) can be re-written as linear equations of the elements of E:

$$AE = 0, \tag{2.9a}$$

where

$$A = \begin{bmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x'_n & x_n y'_n & x_n & y_n x'_n & y_n y'_n & y_n & x'_n & y'_n & 1 \end{bmatrix} \tag{2.9b}$$

and

$$E = [e_{11}\ e_{11}\ e_{11}\ e_{11}\ e_{11}\ e_{11}\ e_{11}\ e_{11}\, e_{11}] \quad for \quad E = \left\lfloor e_{ij} \right\rfloor_{3\times 3} \tag{2.9c}$$

The rotation and translation can be solved using the *eight-point algorithm* (Longuet-Higgins 1981). This algorithm first computes the coefficients of E from equation (2.9a) and then determines T and R respectively. The essential matrix is found by minimizing the error in Equation (2.9), by solving for vector E in:

$$\min_E \|AE\|, \tag{2.10}$$

$subject\ to$: $\|E\|^2 = 2$.

The solution of (2.9) is the eigenvector of $A^T A$ associated with the smallest eigenvalue. Once E is found, T is determined as the unit eigenvector of E^T and R is found by solving the minimization problem:

$$\min_R \|[\mathrm{T}]_\times R - E\|, \tag{2.11}$$

$subject\ to: R\ is\ a\ rotation\ matrix.$

With this matched point method for stereo-camera calibration, the calibration points used in single calibration can be reused in the stereo-camera calibration. In other words, there is no need to explicitly collect stereo-calibration points provided all single-camera calibration points can be seen from both cameras in the stereo pair.

2.1.4 Three-Dimensional Reconstruction by Triangulation

Once a stereo camera is calibrated, 3D coordinates of a point can be reconstructed provided the 2D image coordinates of the point on the two cameras' image planes, and the two cameras' intrinsic parameters are known (calibrated). This process is called *3D reconstruction by triangulation*. The method is described as follows (Bouguet 2006):

A world point *P* has coordinates in the right and left camera coordinate systems: $\overline{X_R} = [X_R\ Y_R\ Z_R]^T$ and $\overline{X_L} = [X_L\ Y_L\ Z_L]^T$, and $\overline{x_R} \overset{\text{def}}{=} \frac{\overline{X_R}}{Z_R} = [x_R\ y_R\ 1]^T$ and $\overline{x_L} \overset{\text{def}}{=} \frac{\overline{X_L}}{Z_L} = [x_L\ y_L\ 1]^T$ in the two cameras' image planes. From rigid motion transformation:

$$\overline{X_L} = R\overline{X_R} + T \tag{2.12}$$

R and T are the rotation matrix and translation vector from right to left camera coordinate systems. For a calibrated camera pair, R and T are known, as they are determined in the stereo calibration process. The triangulation problem is to retrieve $\overline{X_R}$ and $\overline{X_L}$ from $\overline{x_R}$ and $\overline{x_L}$. Equation 2.12 may be re-written as:

$$Z_L\overline{x_L} = Z_R R\overline{x_R} + T \tag{2.13}$$

or

$$[-R\overline{x_R} \quad \overline{x_L}] \begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = T \tag{2.14}$$

Let $A \overset{\text{def}}{=} [-R\overline{x_R} \quad \overline{x_L}]$ (a 3×2 matrix), the least squares solution for Equation 2.14 is then:

$$\begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = (A^T A)^{-1} A^T T \tag{2.15}$$

Let $\overline{\alpha_R} \overset{\text{def}}{=} -R\overline{x_R}$, from Equation 2.15; an explicit expression for Z_R may be expanded as:

$$Z_R = \frac{\|\overline{x_L}\|^2 \langle \overline{\alpha_R}, T\rangle - \langle \overline{\alpha_R}, \overline{x_L}\rangle \langle \overline{x_L}, T\rangle}{\|\overline{\alpha_R}\|^2 \|\overline{x_L}\|^2 - \langle \overline{\alpha_R}, \overline{x_L}\rangle^2} \tag{2.16}$$

$where\ \langle .,. \rangle\ is\ the\ standard\ scalar\ product\ operator.$

2.1.5 Camera Calibration Experiment

2.1.5.1 Experimental Setup

The planar method of Zhang (2000) was used for camera calibration of all cameras in this project due to its ease of setup, inexpensive apparatus, quick processing, and acceptable results expected. The Matlab calibration toolbox (Bouguet 2006) was utilized to carry out both single-camera and stereo-camera calibration. Calibration involved the following steps:

1) A calibration board is prepared with black and white squares in a checkerboard pattern.
2) Images of the calibration board inside the calibration volume in space are taken with the board placed at several different locations with different poses such that no two single poses are parallel to each other. Note that the board has to remain visible in both cameras in the stereo-vision setting, and synchronous images are taken for both cameras. The poses of the calibration board are such that the board is placed to span the entire calibration volume (working space).
3) Corner extraction is performed on all acquired images. A corner is a point where edges of the squares intersect. Since the calibration board remains seen from all cameras in the image acquisition step, the corner points can be used as valid calibration points both in the single-camera calibration and stereo-camera calibration settings.
4) All cameras are calibrated individually by using the extracted corners as calibration points. Both intrinsic and extrinsic parameters along with lens distortion parameters are recovered in this step.
5) Stereo camera calibration is performed to recover the rigid motion transformation between the coordinate systems of the two cameras in terms of R and T. They are derived with the same calibration points used in single camera calibration. All the points must be arranged in a manner such that a corresponding point has the same ordered index across stereo images.

Both the trinocular vision used in hand tracking and the binocular vision used in surface-geometry measurement were calibrated using the same procedures that individual cameras are first calibrated followed by the stereo calibration for all stereo camera pairs. Figure 2.2 is a frame of the calibration board from the trinocular vision system. The top-left image is taken by the left camera, the top-right is by the right camera, and the bottom one is by the mid camera. A frame represents a calibration pose of the board. Sixteen different poses were captured. Each pose possesses 15×14 squares, which make up a total of $16 \times 15 = 240$ calibration points. All of the black and white squares are of the same size $6mm \times 6mm$. Figure 2.3 is a frame from the binocular vision system used in the surface-geometry measurement system. Also, sixteen poses were captured for calibration. Because of the spatial constraints posed by the size of the calibration board and the viewing angles of the

stereo cameras, only the central 5×5 squares are used. The square size is $3mm \times 3mm$. As seen in both figures, all four extreme corners of the calibration board are labeled in order. This is to ensure that calibration points are arranged in the correct order for stereo calibration.

2.1.5.2 Calibration Accuracy Experiment

Since the primary goal of the stereo vision setup in this project was to track points in 3D space, calibration accuracy of the stereo camera pair was more important than calibration of the single cameras individually. The measurement method was the same for all stereo camera pairs presented in this research. Several different test poses of the board were captured. All test poses were different from those used in the calibration, and they spanned the entire working space. Several corner points were extracted from each image captured. The same set of feature points were used across all images. The distances among the feature points was calculated based on all squares being the same size, and lying in the same 2D plane. The distances were saved and used as a reference. All the 3D world coordinates of the feature points were then reconstructed using the 3D reconstruction by triangulation method. The 3D distances of those featured points were calculated using their 3D coordinates in world space. Error analysis was performed on the differences between the reference distances and the reconstructed distances.

2.1.5.3 Results and Discussion

The results showing errors of the calibration accuracy for the binocular vision system for measured distances (MD) varying from 3 distances: 30 mm (120 points), 60 mm (60 points), and 120 mm (30 points) (total 210 points) (first column) are shown in Table 2.1. The second column is the root mean square error (RMSE) with respect to the corresponding measured distances. The third column is the percentage RMSE (percentage of RMSE with respect to its corresponding measured distance). The last column is the standard deviation (SD) for the error distribution. Figure 2.4 is the plot for the RMSE and standard deviation, and Figure 2.5 is the plot for the percentage RMSE. The standard deviation is shown to be very consistent with the RMSE that they almost have the same magnitude. RMSE for all the measured distances ranged from 0.424 mm to 0.580 mm. This is considered very small considering the dimensions of the calibration space, which is about 150 mm × 200 mm × 178 mm.

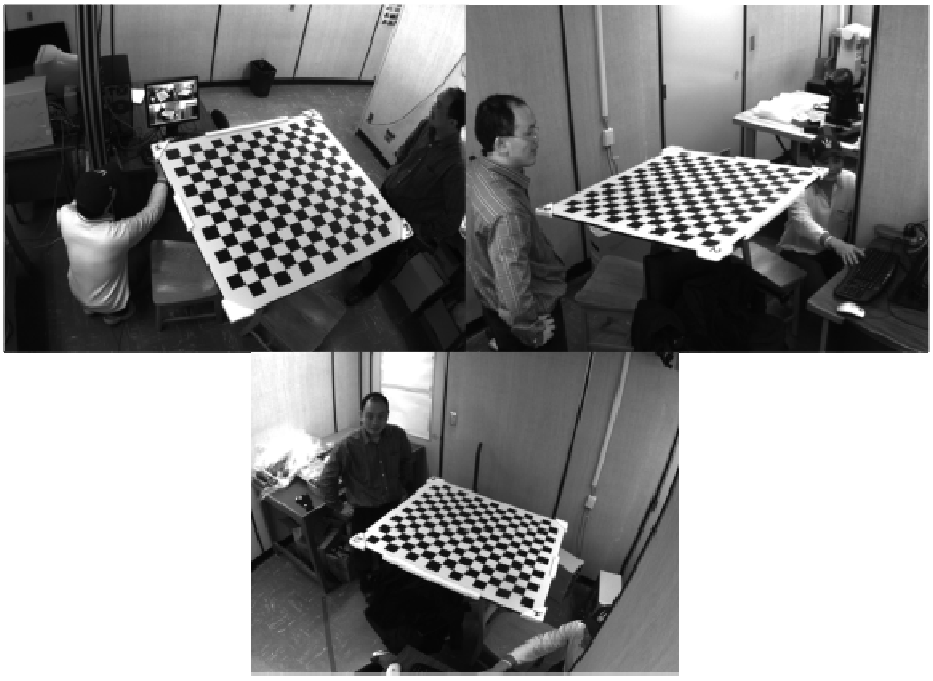

Fig. 2.2 The top-left image is captured from the left camera, the top-right image is from the right camera and the bottom image is from the mid camera. All three images are captured simultaneously from the three different views respectively.

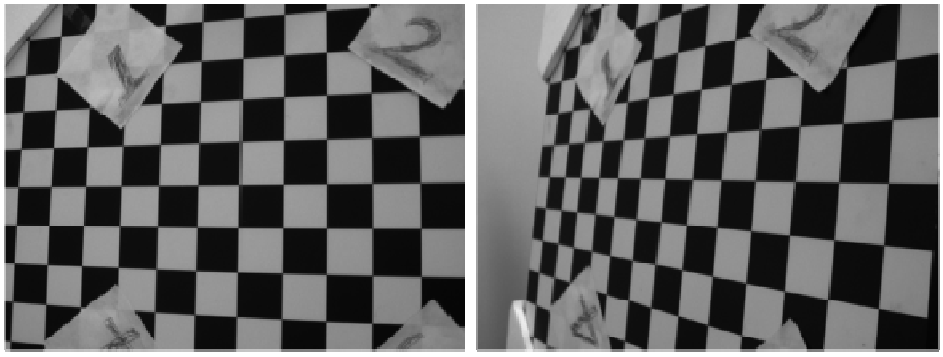

Fig. 2.3 A sample calibration pose for binocular vision. The left image is captured from the left camrea and the right image is from the right camera. Both images are captured simultaneously from the two different views respectively.

Table 2.1 Calibration error for the binocular vision system. MD: measured distance; RMSE: root mean square error; % RMSE: percentage RMSE; SD: standard deviation Note: MD is chosen from 3 distances: 30 mm, 60 mm and 120 mm.

MD(mm)	RMSE (mm)	% RMSE	SD (mm)
30	0.578	1.926	0.580
60	0.424	0.707	0.425
120	0.580	0.483	0.582

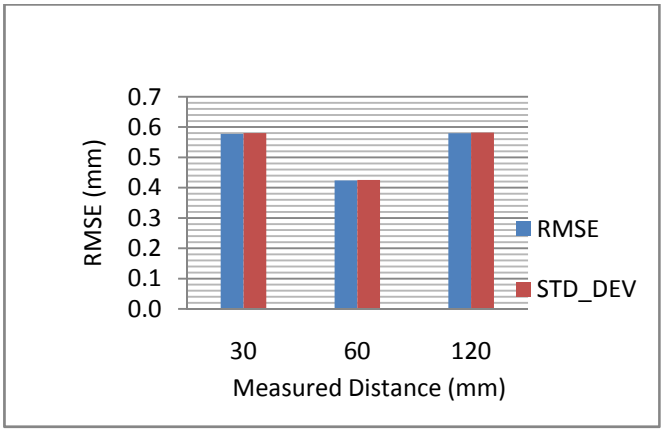

Fig. 2.4 Calibration RMSE for the binocular vision system.

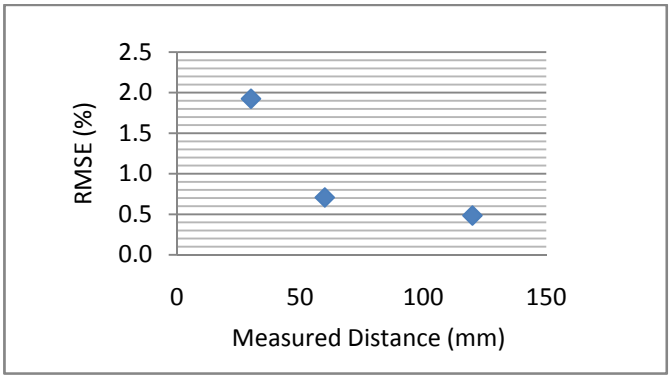

Fig. 2.5 Percentage calibration RMSE for the binocular vision system. Percentage calibration RMSE decreases as measured distance increases.

The results of the calibration accuracy for the trinocular vision system in terms of RMSE,

percentage RMSE, and error standard deviation for measured distances varying from 60 mm to 720 mm for 1820 points (60 mm: 780 points, 120 mm: 390 points, 180 mm: 260 points, 240 mm: 195 points, 360 mm: 130 points, 720 mm: 65 points), using the three stereo camera pairs, respectively, are shown in Table 2.2. Figure 2.6 is a plot for the RMSE for all three pairs. The left-right stereo pair has the lowest error overall. This can be justified that the left-right pair has the longest baseline (the distance between the two cameras). A longer baseline guarantees a wider viewing angle, which provides more depth cue for the points in space such that accuracy of 3D reconstruction by triangulation can be enhanced. RMSEs of all stereo pairs for all the measured distances are below 3.2 mm (from 1.262 mm to 3.185 mm). This is again considered very small compared to the dimension of the calibration space, which is about 1000 mm × 1000 mm × 1000 mm, and the distance from camera to object space, which is about 3000 mm. Figure 2.7 is a plot for the percentage RMSE.

Table 2.2 Calibration error for the trinocular vision system. MD: measured distance (mm), RMSE: root mean squared error (mm), %RMSE: percentage RMSE, SD: standard deviation (mm).The table is the result of three stereo camera pairs: left-mid pair, mid-right pair, and left-right pair.

Left-mid Stereo Pair				Mid-right Stereo Pair				Left-right Stereo Pair			
MD	RMSE	%RMSE	SD	MD	RMSE	%RMSE	SD	MD	RMSE	%RMSE	SD
60	1.568	2.613	1.560	60	1.595	2.659	1.593	60	1.312	2.186	1.311
120	1.647	1.372	1.597	120	1.600	1.333	1.571	120	1.262	1.052	1.246
180	1.600	0.889	1.472	180	1.620	0.900	1.547	180	1.411	0.784	1.376
240	1.876	0.782	1.674	240	1.783	0.743	1.657	240	1.460	0.608	1.395
360	2.083	0.579	1.636	360	1.837	0.510	1.540	360	1.387	0.385	1.216
720	3.185	0.442	1.843	720	2.463	0.342	1.391	720	2.056	0.286	1.554

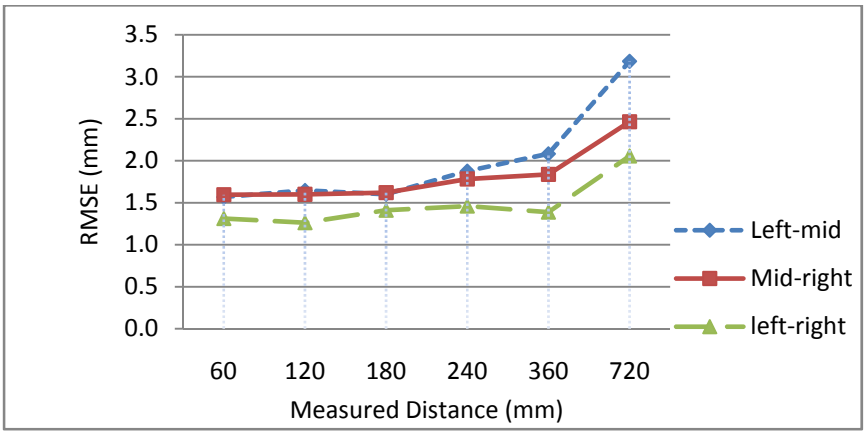

Fig. 2.6 Calibration RMSE for the trinocular vision system. All errors are low level compared to the measured distance.

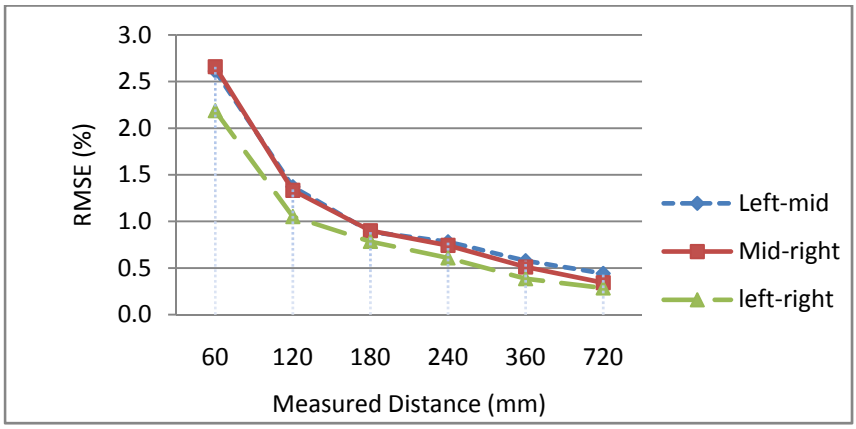

Fig. 2.7 Percentage calibration RMSE for the trinocular vision system. Percentage calibration RMSE decreases as measured distance increases for all stereo pairs.

Both the trinocular and binocular vision systems show a tendency that the percentage RMSE will decrease if the measured distance increases. In other words, if the point displacement is big, the displacement error in tracking is quite small. From the results acquired for both of the vision systems, the calibration and 3D reconstruction algorithms used here are stable and sufficiently accurate.

2.2 Epipolar Geometry

2.2.1 Epipolar Geometry Overview

Epipolar geometry is used as an essential component for marker matching and labeling in the 3D hand tracking system of this research, taking advantage of its capacity of improving marker matching efficiency by reducing the search space for point correspondences in stereo vision. This section covers epipolar geometry, its two major components: essential matrix and fundamental matrix, and how the two components are used in establishing point correspondence for the two views in stereo vision.

Epipolar geometry is the intrinsic projective geometry between different views, and refers to the geometry of stereo vision. It merely depends on the two cameras' intrinsic (internal) and extrinsic (relative pose) parameters regardless of the scene structure or the objects being viewed. The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the *baseline* as an axis. The baseline is defined as the line joining the two camera centers (optical centers of the cameras). When two cameras view a 3D scene from two distinct positions, there are a number of geometric relations between the 3D point and its projection onto the two 2D images for the two cameras, that lead to constraints between the image points. This geometry is usually motivated by considering the search for corresponding feature points in stereo matching. Details for description of epipolar geometry and its use can be found in Xu et al. (1996) and Hartley et al. (2003).

Figure 2.8 depicts the epipolar geometry. The two pinhole cameras left and right are indicated by their camera centers (same as focal points or optical centers in the pinhole camera model) C and C' and their image planes respectively. Let X be a point in 3D space imaged in two views, at x in the left image, and x' in the right. As shown in the figure, the image points x and x', space point X, and camera centers C and C' are coplanar on the plane denoted π. The plane is called the *epipolar plane*. Since the two camera focal points are distinct, each focal point projects onto a distinct point into the other camera's image plane. These two points are denoted by e and e' and are called *epipoles*. Both epipoles e and e' and both focal points C and C' lie on a single line. Line ex and line $e'x'$ are defined as *epipolar lines* such that ex is the epipolar line in left image for image point x' in the right image, and similarly $e'x'$ is the epipolar line in right image for image point x in the left image.

For example, there is an image point x in the left image and the corresponding point x' in the right image needs to be found. The epipolar plane π is determined by the baseline CC' and the ray defined by x. It is known that the ray corresponding to the unknown point x' lies in π. Hence the point x' lies on the line of intersection $e'x'$ of π with the right image plane. The line $e'x'$ is the image in the right camera view of the ray back-projected from x. In terms of a stereo correspondence algorithm, the benefit is that the search for the point correspondence to x can be restricted to only one line, in this case $e'x'$, instead of the entire

image. This constraint greatly improves the point matching efficiency by reducing the search space.

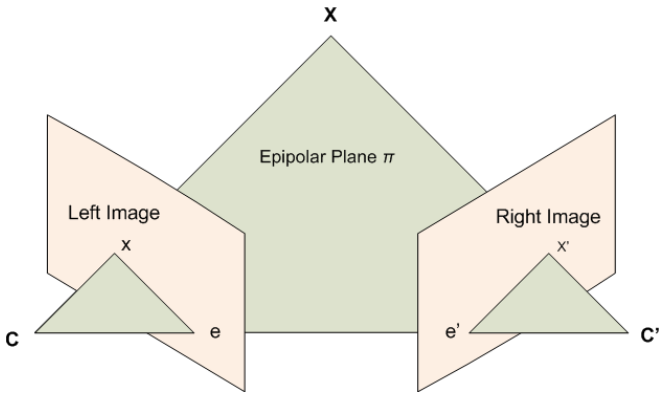

Fig. 2.8 Epipolar geometry. *X* is a world point, x is its image on left image plane and *x'* is its image on the right image plane. *C* and *C'* are camera centers for the left and right cameras respectively. Epipolar plane π is the plane formed by three points: *X*, *C* and *C'*. Epipoles *e* and *e'* are the points intersected by the two image planes and the baseline *CC'* respectively. That all the points *X*, *x*, *x'*, *e*, *e'*, *C*, and *C'* are on the single epipolar plane π, defines the epipolar geometry of stereo vision.

2.2.2 Essential Matrix

The essential matrix is a 3×3 matrix that relates corresponding points in stereo images from cameras using the pinhole camera model. It was introduced by Longuet-Higgins (Longuet-Higgins 1981) in defining an algorithm for determining the relative position and orientation of two camera views. The essential matrix can be used for establishing constraints between matching image points when both the cameras have been calibrated.

For example, if there are two calibrated cameras (which means all distortion is compensated for and simply ignored in the following calculation), the two camera coordinate systems are related by rotation matrix R and translation vector T (details for how to obtain these two geometric transformation parameters can be found in Section 2.1.3 on stereo camera calibration):

$$x' = Rx + T \tag{2.17}$$

where x and x' are images of a 3D point X in homogeneous form. This equation can be re-arranged as:

$$x' \cdot (T \times Rx) = 0 \tag{2.18}$$

Since image points x, x', and camera centers C, and C' are on the same epipolar plane, the above equation can be re-written as:

$$x'^{T} E x = 0 \tag{2.19}$$

The searching for potential matching points in two images from the two different cameras is essentially testing the two points if they satisfy the above equation. Note that before testing the points, they must be rectified such that all the points are on their *ideal* image planes. An ideal image plane is the normalized plane strictly following the ideal pinhole model such that the distance from the plane to the camera centre is 1. With known rotation matrix and translation vector between the two cameras, the essential matrix E can be constructed as:

$$E = [\mathrm{T}]_{\times} R \tag{2.20}$$

where

$$[T]_{\times} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}, \tag{2.21}$$

and

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \tag{2.22}$$

From the above equations, the essential matrix does not depend on the projection geometries of the stereo cameras and the scene structures since only rotation and translation terms are encoded. Instead, it merely depends on the relative pose in terms of spatial rotation and translation between the two coordinate systems of the cameras.

2.2.3 Fundamental Matrix

As described in the previous section, the essential matrix encapsulates the relative pose between two camera views, and enables determination of the point correspondences between two images. However, because the essential matrix does not contain the intrinsic projection geometric information of the cameras, for each point matching one has to rectify the points to their ideal image planes and eliminate the relative pose difference between the world reference and the camera reference systems before the actual matching takes place. This can be very inconvenient and computationally expensive for many image processing applications. A better alternative is the *fundamental matrix*.

The formula defining the fundamental matrix was introduced by both Faugeras (1992) and Hartley (1992). Luong et al. (1996) provide an insight view of the theory, algorithms and stability analysis about the fundamental matrix. The fundamental matrix is a 3×3 matrix of the algebraic representation of the epipolar geometry that encapsulates the intrinsic geometry of the stereo cameras, as well as the relative pose between the two cameras. Like the essential matrix, the fundamental matrix does not depend on the scene structure of the objects being

viewed, and it captures the relative pose between the two cameras such that it can be used in finding point correspondences across stereo images. Unlike the essential matrix, the fundamental matrix also encapsulates the perspective projection geometry of the stereo cameras with both the intrinsic and extrinsic parameters.

The fundamental matrix is defined as follows: If a point in 3D space M is imaged as m in the left view in the homogeneous form of $m = (i, j, 1)^T$, and m' in the right image, then the image points satisfy the relation $m'^T F m = 0$, where F is called the fundamental matrix. One should note that points m' and m are homogeneous image points (points in the real image planes in terms of pixels) unlike for the essential matrix where x and x' must be points in the normalized ideal image planes.

From the essential matrix:

$$x'^T E x = 0 \tag{2.23}$$

With known camera matrices A and A' from the left and right cameras respectively (the formation of a camera matrix is described in Section 2.1 on camera calibration), one can replace the terms x' and x from the above equation using the projective geometry:

$$m'^T A'^{-T} E A^{-1} m = 0 \tag{2.24}$$

Then the relationship between the essential matrix and the fundamental matrix is:

$$F = A'^{-T} E A^{-1} \tag{2.25}$$

The fundamental matrix has the following properties:

- Transpose: If F is the fundamental matrix of the pair of cameras (C, C'), then F^T is the fundamental matrix of the pair in the opposite order: (C', C).
- Epipolar lines: For any point m in one image, the corresponding epipolar line can be calculated as $l' = Fm$. Similarly, $l = F^T m'$ is the epipolar line corresponding to m' in the other image.
- The epipoles: For any image point m other than the epipole e, the epipolar line $l' = Fm$ contains the epipole e'. Therefore, $e'^T (Fm) = \left(e'^T F\right) m = 0$ for all m. The epipoles satisfy: $Fe = 0$, and $F^T e' = 0$.

The matching of two points m and m' in the two images left and right respectively from a stereo camera pair can be achieved in two steps. The first step is to calculate the epipolar line l' for point m using formula $l' = Fm$ then test if point m' is on the epipolar line l' by $m'^T l' = 0$. If the equation is satisfied, they are a potential match. Otherwise matching is terminated with the negative result. The second step is do the opposite: calculate the epipolar line l for point m' using formula $l = F^T m'$ then test if point m is on the epipolar line l by $m^T l = 0$. If the equation is satisfied, they are a match; otherwise they are not.

2.3 Bezier Surface Fitting

2.3.1 Bezier Surface Fitting Overview

Bezier surface fitting, was one of two proposed mapping approaches (the other is neural network mapping), used in the 3D surface-geometry measurement system in this research. The purpose of applying Bezier surface in the 3D surface-geometry measurement was to establish the 2D to 3D coordinate mapping for a range-sensor system. This would simplify the range-sensor calibration process as well as the object measurement. This section describes properties of Bezier curve and how the Bezier curve is extended to a Bezier surface. The method of Bezier surface fitting for the 2D to 3D mapping is also provided.

2.3.2 Bezier Curve

Bezier curves (Zeid, 1991) were developed for *computer-aided-design* systems for car body panels and are widely used in computer graphics to mathematically model smooth curves. A Bezier curve is a parametric curve completely contained in the *convex hull* of its control points, which can be graphically visualized and used to manipulate the curve intuitively. Figure 2.9 shows typical Bezier curves of degrees up to four.

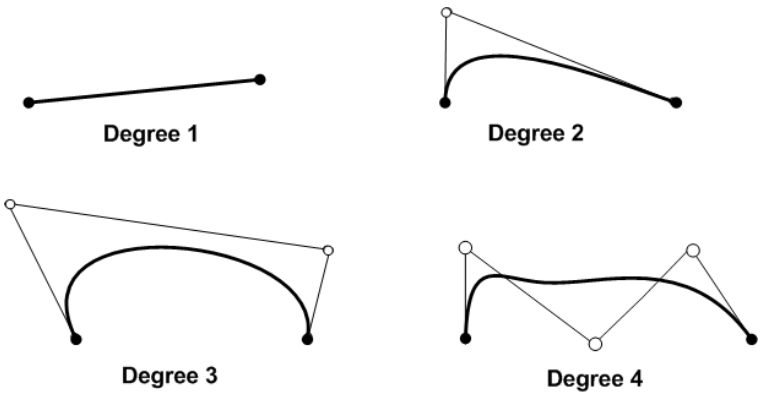

Fig. 2.9 Bezier curves of different degrees. A Bezier curve of 1 degree has two control point, degree 2 has three control points, degree 3 has four control points and degree 4 has five control points. There are *n*+1 control points for a Bezier curve of degree *n*. A set of control points uniquely define a Bezier curve. The first and last control points coincide with end points of curve. The curve is always tangent to the first and last polygon segments.

Characteristics of Bezier curves (Zeid, 1991) can be summarized as:

1) Control points uniquely define the curve shape.

2) $n+1$ control points define an n degree curve (x^n). For example (Figure 2.3.1.b), 4 points define a cubic curve (x^3), 3 points define a quadratic curve (x^2), and 2 points define a straight line (x).
3) Only the first and last control points lie on the curve as end points.
4) The curve is always tangent to first and last polygon segments.
5) The curve shape follows the polygon shape.
6) A change of position of a control point changes the global curve shape.

A Bezier curve can be represented as:

$$P(u) = \sum_{i=0}^{n} q_i \, B_{i,n}(u) \qquad 0 \le u \le 1 \tag{2.26}$$

$P(u)$ is a point on the Bezier curve, n is the curve degree (with $n-1$ control points) , q is a control point for the curve, u is a parameter from 0 to 1, and $B_{i,n}(u)$ is the Bernstein polynomial and is equal to the binomial coefficient $\binom{n}{i}$.

2.3.3 Bezier Surface

A Bezier surface is formed as the Cartesian product of the blending function of two orthogonal Bezier curves and can be used to model a smooth surface. As with a Bezier curve, a Bezier surface is defined by a set of control points (a control point grid). A Bezier surface is a parametric surface and can be of any degree. A bi-cubic Bezier surface is the most commonly used because it generally provides sufficient degrees of freedom for most graphics applications. A bi-cubic Bezier surface is defined by a 4×4 control point grid (Figure 2.10). The parametric surface follows the shape of the control point grid. The surface interpolates (passes through) the corners of the control point grid and it is contained within the convex hull of the control points.

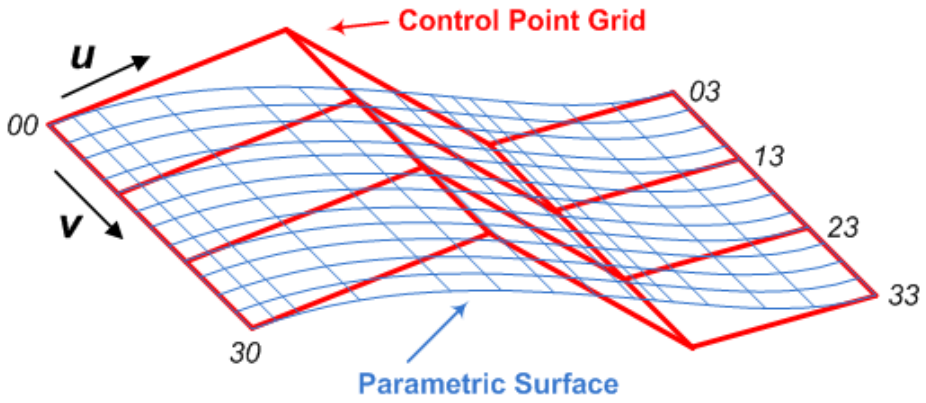

Fig. 2.10 Bi-cubic Bezier surface A bi-cubic Bezier surface is defined by a 4 × 4 control point grid. The parametric surface follows the shape of the control point grid. The surface interpolates the corners of the control point grid and it is contained within the convex hull of the control points.

The corresponding properties of the Bezier curve apply to the Bezier surface as follows:

1) The surface does not, in general, pass through the control points except for the corners of the control grid.
2) The surface is contained within the convex hull of the control points.

In addition, a Bezier surface has the following properties:

1) A Bezier surface interpolates four control corner points.
2) A Bezier surface transforms the same way as its control points under linear transformation and translation.
3) All u and v lines in the (u, v) space, and, in particular, all four edges of the deformed (u, v) unit square are Bezier curves.
4) The surface is tangent to control corner segments.
5) A closed surface is obtained by closing the polyhedron (coincident corner control points).

A Bezier surface can be defined as (Kofman et al. 2007):

$$P(u,v) = \sum_{a=0}^{n} \sum_{b=0}^{m} q_{ab}\, B_{a,n}(u) B_{b,m}(v) \qquad 0 \le u \le 1,\ 0 \le v \le 1 \tag{2.27}$$

where u and v are the parametric coordinates; q_{ab} is a control point with indices from the control polyhedron with $a = 0,1,2,\dots,n$ and $b = 0,1,2,\dots,m$; and n and m are the degrees of the curve along parametric coordinate axes u and v respectively. The number of control points for a Bezier surface is defined by $(n+1) \times (m+1)$. $B_{a,n}(u)$ and $B_{b,m}(v)$ are Bernstein polynomials and can be obtained using:

$$B_{a,n}(u) = \frac{n!}{a!(n-a)!} u^a (1-u)^{n-a} \tag{2.28}$$

$$B_{b,m}(v) = \frac{m!}{b!(m-b)!} v^b (1-v)^{m-b} \tag{2.29}$$

To fit a 3D Bezier surface with k data points, the equation can be written as:

$$P(x, y, z) = FQ(x, y, z) \tag{2.30}$$

$P(x, y, z)$ is the data point matrix of dimensions $k \times 3$. F is an expanded Bezier surface matrix. $Q(x, y, z)$ is the control point matrix which is determined in the surface fitting process. For a Bezier surface of degrees $n \times n$:

$$F = \begin{bmatrix} G_{0,0}(u_0, v_0) & \cdots & G_{0,n}(u_0, v_0) & G_{1,0}(u_0, v_0) & \cdots & G_{n,n}(u_0, v_0) \\ G_{0,0}(u_1, v_1) & \cdots & G_{0,n}(u_1, v_1) & G_{1,0}(u_1, v_1) & \cdots & G_{n,n}(u_1, v_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ G_{0,0}(u_{k-1}, v_{k-1}) & \cdots & G_{0,n}(u_{k-1}, v_{k-1}) & G_{1,0}(u_{k-1}, v_{k-1}) & \cdots & G_{n,n}(u_{k-1}, v_{k-1}) \end{bmatrix} \tag{2.31}$$

$$where$$

$$G_{a,b} = c_{n,a}[u^a (1-u)^{n-a}] \times c_{n,b}[v^b (1-v)^{n-b}] \tag{2.32}$$

$$a = 0,1,2,\ldots,n \quad and \quad b = 0,1,2,\ldots,n \tag{2.33}$$

$$c_{n,a}(u) = \frac{n!}{a!\,(n-a)!} \quad and \quad c_{n,b}(u) = \frac{n!}{b!\,(n-b)!} \tag{2.34}$$

The surface fitting process is completed by solving for the control points q_{ab} in matrix Q in a least-squares sense using the known data points $P(x, y, z)$:

$$Q = [F^T F]^{-1} F^T P \tag{2.35}$$

2.4 Artificial Neural Networks

Artificial neural networks are suitable to solve non-linear domain mapping problems. The proposed approach for the 3D surface-geometry measurement system uses an artificial neural network to map 2D image coordinate to 3D object coordinates in the calibration process as well as for the object measurement. The proposing of this neural network approach is the direct response to the inability of the Bezier surface fitting approach to generate a good approximate of the object domain surface.

An *artificial neural network* (ANN or *neural network* / NN in short), is an information processing paradigm inspired by the way biological nervous systems, such as the brain, process information. It can be viewed as a mathematical presentation of a biological neural network in life forms. The key element of this paradigm is the novel structure of the information processing, which is composed of a large number of highly interconnected processing elements (*neurons*) working in unison to solve specific problems. An ANN is configured for a specific application through a learning process. Learning in biological systems involves adjustments to the synaptic connections between neurons. This also applies to ANNs. ANNs have been applied to an increasing number of real-world problems of considerable complexity (Haykin 1994) including stock price prediction (Hellstorm 1998), image processing (Durackova 2006), and health care diagnostic systems (Atieza 2003). The most prominent advantage is in solving problems that are too complex for conventional technologies (e.g. problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found) by utilizing the learning capacity of the ANNs.

As in biological neural networks, ANNs use a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for future use. This includes two steps. In the first step, knowledge is acquired by the network through a learning process. Inter-neuron connection strengths, known as synaptic weights, are then used to store the knowledge. The processing of learning is essentially the process of adjustment of the synaptic network weights. There exist many types of ANNs in the literature. Details about the history, types, features, and their use can be found in Haykin (1994).

The type of ANN used in this research is *multilayer perceptron* (MLP) network. The MLP is the most commonly used type of neural networks due to its simplicity in design and great efficiency in execution. It implements *feed-forward* data flow (data flows in only one direction and there are no recurrent links) and consists of one input layer, one output layer, and an arbitrary number of hidden layers. Each layer has one or more neurons that are directionally linked with the neurons from the previous and the next layer. Figure 2.11 shows an example of a 3-layer MLP with 3 inputs, 2 outputs and a single hidden layer with 5 neurons. All of the neurons in a MLP are similar so that each neuron takes several input and output links. A neuron takes the output values from several neurons in the immediate preceding layer as input, and passes the response to several neurons in the immediate following layer. The values obtained from the preceding layer are summed with certain weights that can be different across individual neurons, plus the bias term, and the sum is then transformed using the designated *activation function* that may also be different across different neurons.

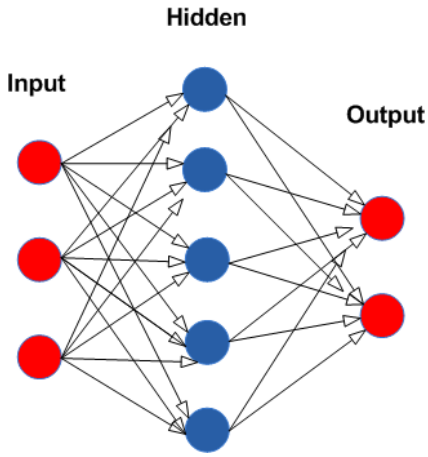

Fig. 2.11 Multilayer Perceptron (MLP) neural network. A sample MLP has one input layer, one output layer and one hidden layer. The number of nodes in a layer and the number of hidden layers for an MLP can be arbitrary. Output of a neural node stays the same for all outgoing connections. All the connections are from left to right for a feed-forward neural network.

Figure 2.12 depicts the working scheme of a typical neuron. For the given outputs $\{x_j\}$ of the layer n, the outputs $\{ y_i \}$ of the layer $n + 1$ are computed as (note that the bias is treated as x_0):

$$u_i = \sum_{j=0}^{N} w_{ij} x_j \tag{2.36}$$

$$y_i = f(u_i) \tag{2.37}$$

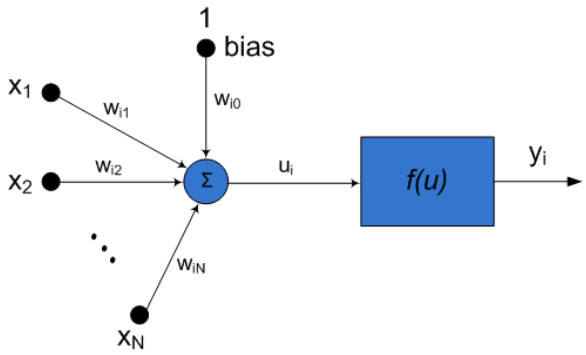

Fig. 2.12 Neural model. A weighted sum of all the inputs for a node including the bias is input into the activation function. Output of the activation function is defined as the final output of the neural node.

Function f is the *activation function* of that particular neuron. The activation function can be a linear, threshold, or sigmoid function. For a linear activation function, the output activity is proportional to the total weighted output. For a threshold activation function, the output is set at one of only two levels depending on whether the total input is greater than or less than some threshold value. For a sigmoid function, the output varies continuously but not linearly as the input changes according to the sigmoid equation. A sigmoid function bears a greater resemblance to real biological neurons than linear or threshold functions. Figure 2.13 shows a symmetrical sigmoid function. A typical sigmoid function takes the form of $f(x) = \beta * (1 - e^{-\alpha x})/(1 + e^{-\alpha x})$, where α and β are arbitrary values used to control the overall shape of the function.

An MLP network works as follows: It takes a feature vector, the size of which is equal to the size (number of neurons) of the input layer, as input. The input values are passed to the first hidden layer. The outputs of the hidden layer are computed using the network connection weights and activation functions and passed onwards until reaching the output layer. MLP networks are often used in *supervised learning* problems, which incorporate an external evaluator so that each output unit is compared with its expected response to the input signals. This means that there is a training set of input-output pairs and the network must learn to model the dependency between them. The training means adapting all the weights and biases to their optimal values for the given pairs. The criterion to be optimized is typically the squared reconstruction error. The supervised learning problem of the MLP can be solved with the *back-propagation algorithm*. The back-propagation algorithm consists of two steps. In the forward pass, the predicted outputs corresponding to the given inputs are evaluated using the expected outputs. In the backward pass, partial derivatives of the cost function with respect to the different parameters are propagated back through the network. The chain rule of differentiation gives very similar computational rules for the backward pass

as the ones in the forward pass. The network weights can then be adapted using any *gradient-based optimization algorithm*. The whole process is iterated until the weights have converged.

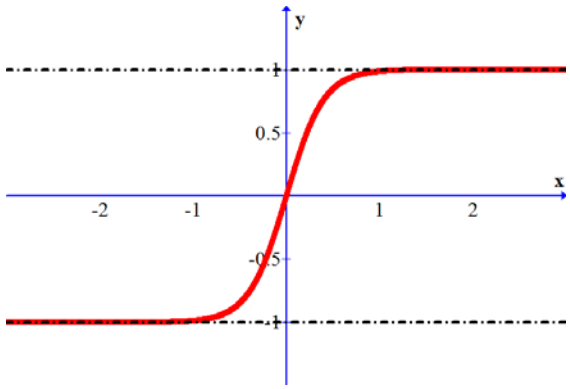

Fig. 2.13 Symmetric sigmoid function (Haykin 1994). The Sigmoid function is the most common type of activation function for neural networks because it greatly resembles the biological features of a real neuron presented in life forms while having ease of computation.

The more complex (number of hidden layers and/or number of neurons per layer) a neural network is, the more the potential network learning flexibility can be achieved. The training error can be made very small by using a complex network; however, the learned network will also adapt to the noise existing in the training data such that the error usually starts increasing after the network size reaches some threshold. In this case, the network exhibits the over-trained effect. In addition, large networks take significantly longer to train compare to their smaller counterparts. Hence it is always advised to start training using a relatively smaller network with only the essential features.

Once the artificial neural network is trained using sample input and corresponding output data, the network can be run to compute a set of output data, for any new input data. In this research, input 2D image coordinates are mapped to 3D object coordinates to serve as the calibration of the range sensor for 3D surface measurement. 3D object coordinates can then be calculated for any new 2D image coordinate captured by the range-sensor during a 3D surface measurement. This is discussed in detail in Section 4.6.

Chapter 3

Three-Dimensional Hand Tracking

3.1 Three-Dimensional Hand Tracking System Design

3.1.1 Hand Tracking System Setup

The 3D hand tracking system permits a human operator to communicate simultaneous motions tasks to a robot manipulator by having the operator perform the 3D human hand-arm motion that would naturally be used to complete an object manipulation task. As depicted in Figure 3.1, there are two sites in the 3D hand tracking system: operator site and robot manipulator site. The two sites are next to each other and linked together by two controlling computers connected to a *local area network* (LAN). All the communication between the two sites takes place on the two controlling computers. The computer at the operator site controls the three cameras recording the hand-arm motions of the human operator, and sends the information to the computer at the robot site through the LAN. The computer controlling the robot manipulator takes hand-arm pose and grasping position information and instructs the robot to act accordingly by sending the information to the robot controller (robot actuator). The hand tracking system captures hand motion events at the operator hand-tracking site; the events at the robot-manipulator site are therefore beyond the focus of this tracking system.

Figure 3.2 shows the physical setup of the operator hand-tracking site. At the operator site, there are three cameras rigidly mounted on three mutually orthogonal walls respectively. All cameras have a fixed focus optical lens and they face the same operation volume/space. Three circular markers (19 mm (0.75 inch) in diameter) are adhered to the top of the operator's glove such that all the cameras can see the markers. The human operator controls the movement of the robot manipulator by moving the hand with the glove in the operation volume to simulate object task operations such as grabbing, moving, rotating, relocating, and releasing an object. The robot manipulator at the other site will copy the motion in real-time as the human operator moves their hand. The robot manipulator (Figure 3.3) has six joints, which provide six degrees of freedom. This allows the robot to move the end-effector throughout the working space within the reach of its arms. There is a two-finger robot gripper at the robot manipulator end-effector. When the two fingers are closed, a grab operation is performed while a release operation is carried out when the fingers are moved apart.

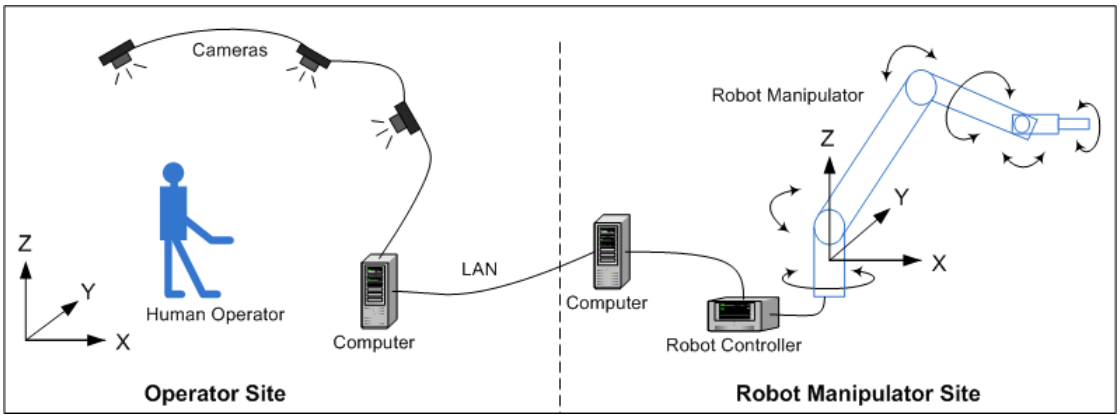

Fig. 3.1 Schematic representation of the hand tracking system. The hand tracking system consists of two sites, the operator site and robot manipulator site. The two sites are connected by a LAN connected to the two controlling computers. Task demonstration is performed at the operator site while task actuation is carried out by the robot manipulator at the robot manipulator site. The human operator wears a hand glove with three physical markers on it. A sequence of actions of the human operator is captured using the three calibrated cameras at the operator site. Information about the hand-arm motion is extracted from the image sequence by analyzing the location changes of the markers. Motion information is then sent to the robot controller at the robot manipulator site for action actuation. The robot manipulator has six joints, which provides six degrees of freedom. There is a two-finger gripper located on the end-effector of the robot manipulator. The two fingers can either be closed or opened. A closing operation resembles a grab while an opening operation resembles the release object manipulation task.

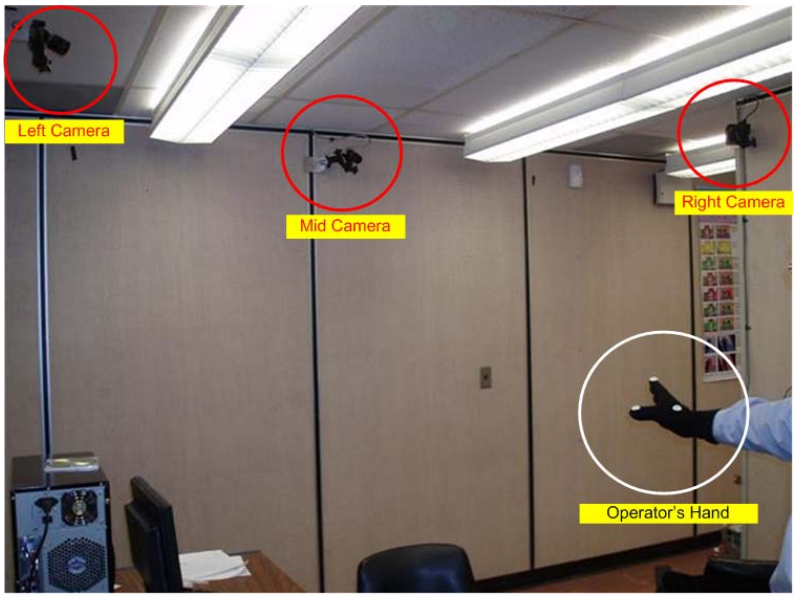

Fig. 3.2 Operator site of the robot-teleoperation system. Three cameras are approximately two meters above the ground and mounted on three different mutually orthogonal walls respectively. The human operator wears a glove with three markers on it. Hand motion of the operator remains in the calibrated working volume of approximately one cubic meter within the visible area of all cameras.

Fig. 3.3 Robot manipulator site of the robot teleoperation system. The robot manipulator has six joints for six degrees of freedom. There is a two-finger gripper at the end effector to perform object manipulation tasks. The fingers are closed to grab an object, and opened to release an object.

3.1.2 Hand Tracking Operation and Design Requirements

The tracking approach is marker-based, where the human operator is required to wear a black glove with three circular markers on it. The three markers are placed near the tips of the thumb and index finger, and on the wrist of the glove, respectively. The 3D hand tracking in this project is basically the tracking of the three markers attached on the operator's glove from the three different camera views. The 2D coordinates of the three markers are extracted from the three images captured simultaneously from the three cameras, respectively. 3D coordinates of the markers with respect to the world coordinate system are computed from the obtained 2D coordinates using the 3D reconstruction by triangulation operation from the three stereo camera pairs. There are three cameras namely left, mid, and right that form three stereo pairs : left-mid pair, mid-right pair, and left-right pair, respectively. The 3D coordinates of the markers are then sent to the computer controlling the robot manipulator end-effector so that the robot manipulator can mimic the hand-arm motions demonstrated by the human operator with the relative position and orientation of the hand preserved. This occurs in real-time, at thirty frames per second, so that the time lag between the human motion and the robot manipulator actuation is minimized and the robot motion can be monitored by direct visual feedback.

In order to achieve a fast and reliable hand tracking system, the following requirements must be met:

a) Tracking operations should be performed in real-time.

b) Ideally, the tracking system should be able to perform well enough that background objects can be ignored. However, in the initial intended application of the robot-teleoperation system, the operator site can be a structured environment, where the background is specially suited for the hand-tracking task.

c) Ideally, the tracking should function well under different lighting environments especially under a non-uniform lighting distribution. However, in the initial intended application of the robot-teleoperation system, the operator site can be a structured environment, where the lighting is specially suited for the hand-tracking task.

d) The merging of markers in camera views should be accommodated such that they can be separated in order to continue tracking of the hand orientation. (e.g. for the task of grabbing an object, the markers on the thumb and index finger tend to merge into a single blob in a camera image).

e) Occlusion of markers in camera views must be handled by the tracking system (e.g. when the hand is rotated, one camera may not be able to see all three markers on the glove).

3.2 Hand Tracking Scheme

The 3D hand tracking scheme consists of multiple steps that can be summarized as follows (Figure 3.4):

1) Images of the human operator's hand are captured simultaneously by the three cameras of the trinocular vision system.

2) A search window for markers is constructed for each individual image. The entire image will be used if this is the first image frame. Blobs (connected pixels in an image) are extracted from the three captured images within the search windows, and blob filtration is performed in order to consider only potential markers and ignore the noise.

3) If there are fewer than two blobs remaining, further processing will be ignored by exiting the cycle. In such case, the program terminates if an exit signal is encountered; otherwise, the search window will be enlarged and a new cycle will be initiated.

4) Blobs are matched using epipolar geometry across different camera images from the same frame such that blob correspondences are established.

5) Blobs analysis is performed to label the markers in desired order (e.g. the first marker is always the wrist marker, the second is the thumb marker, and the third is the index finger marker).

6) 3D coordinates of the labeled markers with respect to the world coordinate system are reconstructed using 3D reconstruction by triangulation based on stereo vision.

7) 3D coordinates of the labeled markers, which encode the position and orientation of the operator's hand, are sent through the LAN to the robot manipulator site.

8) Missing or occluded markers (if there is any) in images will be reconstructed virtually using 3D world to 2D image projection.

9) Each search window is updated based on the location of the markers in the images.

10) The cycle ends and a new cycle will be started.

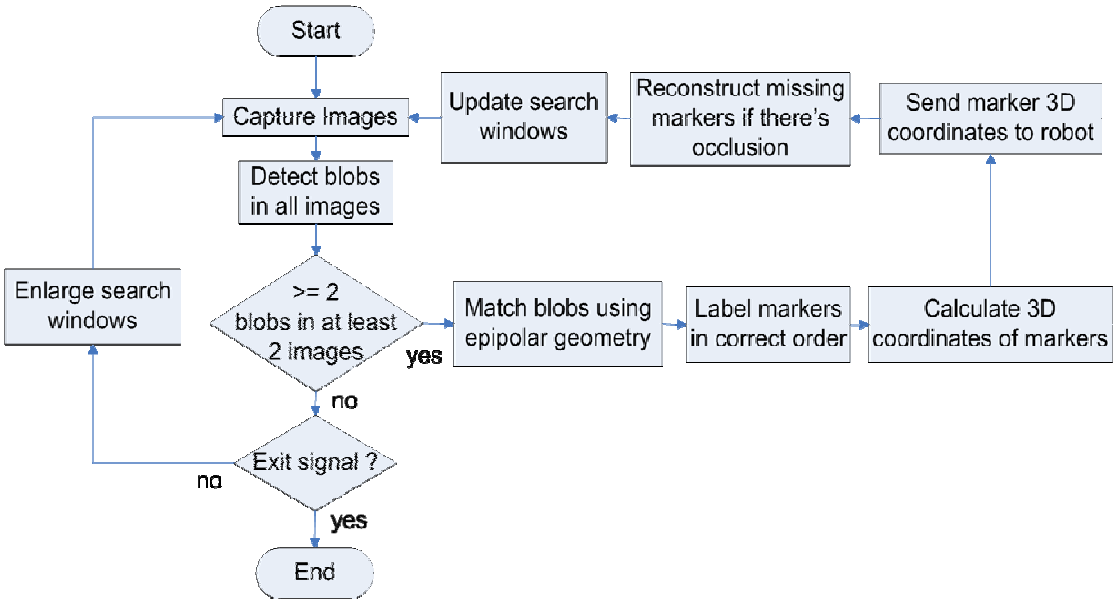

Fig. 3.4 Hand tracking scheme.

A search window is a virtual rectangular region defined in an image, in which the markers are searched for and the rest of the image is set to be invalid for the search and simply ignored. The search windows for the markers in the images are dynamically updated. In the initialization stage, the whole image is used as the initial search image. The search windows are updated in the subsequent frames in a way that the detected markers in the current frame are always in the centre of the search window for the next frame. This makes the search windows as small as possible but sufficient to contain the markers for the next marker search. It is crucial to the real-time hand tracking scheme since marker detection is costly in terms of computing resources. The decrease in search window size boosts the searching efficiency and enhances the system responsiveness by reducing the search time required. Among all the steps in the hand tracking scheme, Steps 3 to 5 play the most prominent role that affects the overall reliability of the whole tracking system. These steps can be seen as marker detection and marker matching and labeling, and are discussed in detail in the following two sections.

For the 3D reconstruction of markers, a single stereo camera pair, which consists of only two cameras that strictly follow the epipolar geometry of stereo vision, will be sufficient to provide enough information to reconstruct the depth information about the markers: With a single camera, the 2D information is known in the camera coordinate system. The other camera gives the opportunity to use the 3D triangulation technique to calculate the depth coordinates. The introduction of the extra camera serves as the mechanism to deal with marker occlusion that appears in some of the camera views such that the hand rotation angle permitted within the operation volume can be significantly increased. In addition to marker occlusion due to hand rotation, obscuring of markers may also occur by the body of operator and this can also be solved using the extra camera. In addition, the extra camera can be utilized to verify the validity of the marker 3D coordinate calculation. Most of the time during the 3D hand tracking, only the primary pair of cameras is used to carry out all the computation, including marker matching, marker labeling and 3D reconstruction of marker position by triangulation. The extra camera is simply ignored for greater efficiency in processing. The primary pair can be a random choice among the three camera pairs. The full set of cameras is used only in one of two cases: when occlusion is detected or the merging of the thumb marker and the index finger marker is detected. The second case happens when the operator is grabbing an object so that the thumb and index finger markers may appear as a single blob in the camera image. In order to deal with marker occlusion and marker merging without compromise of efficiency in processing, a camera-pair utilization scheme is proposed as follows:

A. For any given frame, only one pair of cameras is used for marker matching and labeling and marker 3D reconstruction by triangulation.

B. If all markers are successfully detected by the primary pair, only this pair is used.

C. Back-projection is performed for the ignored view. The locations of the markers in the image plane are computed by back-projecting the three markers from 3D space (computed by the utilized stereo pair) onto the image plane of the ignored view. These 2D image positions are used to update the search window for this view to be used in the next frame.

D. If not all markers are detected by the primary pair, other pairs will be tried until a successful pair is found. If all trials fail, Step E is performed.

E. If there are only two markers detected by two or three of the three images, it is assumed that the thumb marker and the index finger marker are merged into one in the images. Computation is performed on the image pair that detects only two markers. If there is more than one valid camera pair, the primary pair always takes the highest precedence.

F. If there is no valid pair found after performing D and E, an invalid frame is marked. In such case, the system simply ignores this frame.

Figure 3.5 is a normal valid frame extracted the 3D hand tracking operation. The top-left image is captured from the left camera, the top-right image is from the right camera, and the bottom image is from the mid camera. The large white rectangles in the images are the

current search windows for the markers. The large circle in the right image indicates that the markers in this image are back-projected from the markers' true 3D coordinates in space. Back-projection takes place on the right image since the primary (left-mid) stereo camera pair is utilized is this frame; therefore, the right image is simply ignored in computation of the 3D locations of the markers. The figure also shows the three correctly matched and labeled markers. The marker bounded by a square is the wrist marker, the one encircled is the thumb marker, and the one marked with a cross is the index finger marker. Figure 3.6 is a valid frame where marker occlusion occurred in the left image. The occlusion invalidated the primary camera pair and the mid-right camera pair is utilized instead in this case. Back-projection takes place in the marker-missing image (left). The occlusion problem is hence solved by the proposed camera utilization scheme. This also applies to situations where all the markers are totally occluded in an image.

3.3 Marker Detection

The operating environment for the 3D hand tracking is a slightly cluttered laboratory. Noise elimination in marker detection is therefore a significant consideration in the design phase of the whole system. Blob detection is used as a means of detecting the three markers in the image frames in this hand-tracking system. The initial intensity values of the thresholds for the individual images captured from the three cameras are determined in an automatic manner. Essentially, a search for the three markers in the image is carried out while varying the threshold value in the thresholding process. The initial threshold is set to the current value when three markers are successfully recognized in the image. This is done only once and is only for the initialization stage since this automatic threshold determination process is computationally expensive and hence not suitable for real-time or time-lag sensitive applications as in this case. However, marker detection problems may arise after the first frame as the operator moves the hand around inside the operating volume. With the initial threshold value, it may be possible that not all of the markers will be able to be detected due to the non-uniform distribution of light in the operating volume, the obscuring of the light source by the operator's body, or the environment background containing some brighter regions seen as noise in the image in a more complex operating environment. This results in the same marker having different intensity values across different image frames, and it makes the marker detection fail. Thus, a global threshold for all frames will not function as desired, and an adaptive thresholding technique must be designed.

Fig. 3.5 Hand tracking with all markers visible in all cameras. The top-left image is captured from the left camera, top-right from the right camera, and bottom one from the mid camera. The big white rectangles in the images are the current search windows for markers. The large circle in the right image shows that the markers in this image are back-projected from the markers' true 3D coordinates in space. Back-projection takes place in the right image since the primary (left-mid) stereo camera pair is utilized is this frame; therefore, the right image is simply ignored in computing the 3D locations of the markers. The three markers are correctly matched and labeled. The markers marked by a square, circle, and cross are the wrist, thumb, and index-finger markers, respectively.

Fig. 3.6 Hand tracking with marker occlusion. Occlusion occurs in the left image and invalidates the primary camera pair. The mid-right camera pair is utilized instead. Back-projection takes place in the marker-missing image (left). The occlusion problem is hence solved by the proposed camera utilization scheme. This approach is also used when all the markers are totally occluded in an image.

The proposed thresholding approach is a dynamic multi-thresholding technique that incorporates multiple thresholds derived from a single base threshold value. The base threshold value *b* is the initial threshold value used to perform the blob detection, and it is determined automatically in the initialization stage. A pre-defined one dimensional array of threshold offsets is set before the program starts.

The threshold value determination scheme is ruled by the following criteria:

1) The last successful offset index *s* is set to 0 for the first frame.
2) The current threshold value for marker detection is: $T = b + offset[i]$.
3) For the same frame, the offset index *i* iterates from left to right in array *offset* until the markers are found in the image or the loop terminates and marks the frame as invalid frame if end index is reached and markers are still not found. Iteration always starts from: $i = s - 1$ *for* $s > 0$, *or* $i = s$ *for* $s = 0$.
4) The current offset index *i*, for which markers are successfully detected, is stored in memory as : $s = i$, to be used in the immediate subsequent frame.

A typical array *offset* can be set as: [0, -2, -4, -8, -16, -32, -64]. In this case, if first index (0) is used, the current threshold value is the same as the base threshold value. The values contained in the array are usually negative because operation generally starts in a well-lit region, as the operator moves, some light source maybe blocked by the operator's body such that lower threshold values are required to detect markers in a dim image frame. Since the operator moves around inside the operating volume, the hand of the operator may sometimes move from a dark region back to a bright region. If the threshold value is too low when the hand is in the bright region, there will be too many blobs detected, many unrelated to the markers. This will greatly degrade the performance of the system or even make it unusable. A means to accommodate this transition is hence necessary. The one-step-back iteration mechanism (step 3) in the scheme solves this problem by allowing a smooth transition for the operator to move from the dark region back to the bright region. Only one thresholding step overhead is involved, which still permits tracking in real-time at video frame rate. It should be noted that, for this algorithm to work, the auto gain feature must be turned off. The auto gain feature is present in many higher-end cameras and it is enabled by default in order to take properly exposed pictures. It automatically adjusts the intensity values of image pixels based on the brightness, contrast, and image histogram. This makes intensity values of markers fluctuate over different frames and makes marker detection less reliable.

3.4 Marker Matching and Labeling

After blobs are detected from the previous step, the system needs to know the correspondence between blobs among the different images, as well as which blobs correspond to the wrist, thumb, and index finger, respectively. This is necessary in order to disregard the irrelevant blobs and calculate the 3D positions of the markers. Marker matching is the process to find the marker correspondences across different images in the same frame. It is essentially the feature point matching problem in stereo vision. Marker labeling is the process to number the three detected markers in the three images in the same order in every frame, e.g. first, second and third markers always correspond to the wrist, thumb, and index finger over all frames. With markers correctly matched and labeled, the current 3D positions of the markers can be calculated. The position and orientation of the hand and arm of the operator can be determined from these markers in the 3D world coordinate system.

Figure 3.7 is the flow chart for the algorithm of marker matching and labeling for the utilized camera pair. Before the marker matching and labeling is performed, potential markers are isolated in the images and the rest is disregarded as noise. This is a two-step process. The first step is to eliminate the blobs which have a size beyond a certain range. Because all markers have a fixed size, the size of the markers in the images can be approximated using the known camera focal lengths and distances from the operating volume to the cameras. This step is performed in all images separately. It effectively eliminates many irrelevant objects from the complex background.

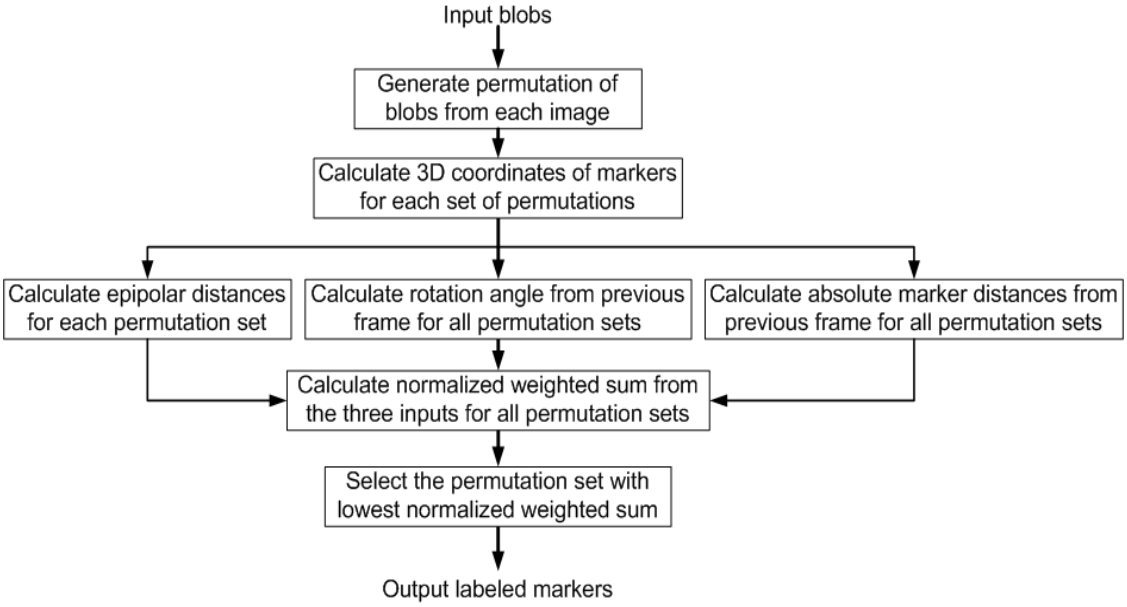

Fig. 3.7 Marker matching and labeling algorithm

The second step uses the epipolar geometry of the three pairs of stereo cameras (left-mid, mid-right, and left-right camera pairs). As discussed in Section 2.2 on epipolar geometry, point correspondence can be found in a stereo camera pair by computing the epipolar line in Image 2 for a point in Image 1. If a point to be matched in Image 2 lies on the epipolar line, the point is a potential match for the point in Image 1. If a point in Image 1 has no potential match in Image 2, it is disregarded as noise. In preliminary tests, it was found that this determination of correspondences did not work well due to errors in the camera calibration process or in the calculation of the blob centre in the blob detection process. A matched point may not lie exactly on the epipolar line but it will be very close as expected. A threshold distance from a point to be matched to the epipolar line was therefore used instead in all calculations of epipolar distance, and it proved to be very fast and reliable. After taking the two veto steps above, most of the noise blobs are eliminated, and the rest, considered as potential markers, are processed by the marker matching and labeling algorithm.

Both matching and labeling can be done effectively in one algorithm. Firstly, for every frame, the permutations of three blobs in the blob pool are generated for each individual image in the utilized stereo pair. Each image and blob are treated with equal probability. Iteration through all permutation sets is done in all images. A permutation set from each image is used, and three scores used as criteria to match and label the blobs (potential markers) are computed. The three scores are the epipolar distance score, pose rotation score, and markers translation score, respectively. The epipolar distance score is calculated as the sum of the epipolar distances from both directions, D, (explained and defined below; details about epipolar geometry are given in Section 2.8.) for all blobs with the same indices in the two chosen permutation sets. For those indices, it is assumed that this first index marks the

wrist marker, the second marks the thumb marker, and the third marks the index finger marker. This index labeling scheme applies to the calculation of the other two scores as well. The epipolar distance for both directions is defined as follows: for a stereo camera pair, there is a point P_1 in Image 1 and a point P_2 in Image 2. The epipolar line in Image 2 for point P_1 is l_1, and the epipolar line in Image 1 for point P_2 is l_2. The distance can be calculated as follows:

$$D = \|P_1 - l_2\| + \|P_2 - l_1\| \tag{3.1}$$

Note that this epipolar distance score is calculated using the blobs in the same frame. The calculations of the other two scores, however, use the previous frame as a reference. The assumed 3D locations of the potential markers are calculated using the 3D triangulation technique with the utilized stereo pair. The pose rotation is the rotation angle of the *pose axis* with respect to the previous frame. The pose axis is defined as the line in 3D space passing through the wrist marker and the virtual point between the thumb marker and the index finger marker, as shown in Figure 3.8. (The real markers are white, and the centres of the markers are used for all marker distance calculations). The virtual point lies on the straight line connecting the thumb and index-finger markers in 3D space, and the distance from the virtual point to the thumb marker is one third the distance from the thumb to the index marker. This definition of the pose axis was used to maintain the axis in the same orientation when the hand is closing or opening, based on earlier experience (Kofman et al., 2005). The translation score is the sum of the 3D displacements of the assumed markers with respect to the true markers computed in the previous frame. A weighted sum of the three scores are then calculated and used to evaluate the validity of the permutation set. The values of the weights were empirically determined through repeated tests. The permutation set with the lowest weighted sum is then selected as the correct labeling scheme. The calculation of the pose-rotation and marker-translation scores are based on the assumption that the hand movement is slow enough that the rotation and displacement of the operator's hand are small in magnitude as seen by the cameras, for a real-time tracking system.

Fig. 3.8 Pose axis, defined as the line in 3D space passing through the wrist marker and the virtual point between the thumb marker and index finger marker. The distances of the virtual point to the thumb and index-finger markers is 1/3 and 2/3 the distance between the thumb and index-finger, respectively.

The marker matching and labeling algorithm can also handle merging of the thumb and index finger markers such as in an object grabbing pose. In this case, only two markers are detected for the utilized stereo camera pair. For the generation of the three-marker permutation sets in the algorithm, one of the two markers is treated as two markers for the thumb and index finger (merging only occurs with this pair).

3.5 Hand-Tracking Experiment

3.5.1 Experimental Setup

The controlling computer at the hand tracking site has an Intel Pentium 4 2.4 GHz processor, 1.5 GB PC3200 DDR RAM, 200 GB ATA133 hard drive, and a PCI Firewire800 host card. The operating system is Windows XP with Service Pack 2. The hand tracking system is developed with Microsoft Visual Studio 2005 with Service Pack 1. OpenCV 1.0 and cvBlobFinder computer vision programming libraries are used for image manipulation and blob detection. The three cameras are off-the-shelf products Model Flea2 (Point Grey Research Inc.), with 8-bit grey-scale, and capable of capturing images at 30 frames per second (fps) at a resolution of 800 × 600 pixels. The camera interface is Firewire800, which is able to asynchronously stream data at a speed of 800 Mbits/s. Each camera has a fixed focal length lens, 3.6 mm for the left camera (closest to the operation volume) and 6 mm for the other cameras. Each camera is rigidly mounted on an adjustable mounting head that is attached on a wall. The walls are mutually orthogonal. All cameras are approximately two meters above the ground. The left camera is approximately 1.5 m from the operating volume while the other two cameras are approximately 3 m away. Three white circular markers, 19

mm in diameter, are attached to a black glove that is worn by the human operator. The operating volume is approximately 1 m^3.

3.5.2 Experiment

A series of tests were performed to verify that the tracking system can permit a typical task, such as a pick-and-place task, to be completed by an operator teleoperating the robot. The tests evaluate whether the maker-based hand-tracking vision system is accurate enough to enable a human operator to control the movements of the robot manipulator in real-time to perform the object manipulation tasks. Tests were performed with three project researchers as operators repeating a pick-and-place teleoperation task ten times in three series of tests, respectively. The motion of the robot manipulator was directly observed by the human teleoperator without the use of cameras at the robot site that were used previously (Kofman et al., 2005). The tests also served as a preliminary evaluation of the ability of the operator to use the human-robot interface using direct visual feedback. For each test, the operator wore a glove with three markers on the wrist, thumb, and index finger positions respectively, and moved their arm in the stereo camera calibrated operating volume to control the motion of the robot end-effector for a common pick-and-place object manipulation task. The task involved controlling the robot manipulator to pick up an object from a predetermined starting position and place the object with a predetermined corner of the object at a predefined corner location on a target with the specified object and target edges aligned as shown in Figure 3.9. The complete set of subtasks for a pick-and-place task include: approach object, grab object (closed gripper), lift object and move it over the target, place object on target, release object (open gripper), move away from object. Figure 3.10 illustrates the sequence of actions during a successful teleoperation. For a complete task for the teleoperation, the operator performs a sequence of actions, as shown in the figure from (A) to (E). The task object was a rigid plastic foam block 200 mm x 100 mm x 50 mm. The target location was a 100 mm x 100 mm square paper. The tool roll of the end-effector was fixed during the tests. The errors in rotation and translation in placing the object on the target with the correct edge alignment and corner location was recorded for each test.

3.5.3 Results and Discussion

Tables 3.1 to 3.3 show the positioning errors for the three series of tests, respectively. The errors in translation in *x* and *y* axes directions are parallel to the two target edges, and the rotation of the object is about a vertical axis. The mean of the absolute values of the error and standard deviation are given. Figure 3.11 is shows the errors in position and orientation for all teleoperation test combined (30 tests). The error ranged from -22 mm to +15 mm with a mean of 8.8 mm along the *x*-axis, from -20 mm to + 22 mm with a mean of 8.4 mm along the *y*-axis, and from -7 to +15 deg with a mean of 4.6 deg about the *z* axis. The results for the translations are acceptable considering that the robot manipulator was 2.5 m from the operator, and the robot manipulator can often block the view of the object and target. The rotation error in *z* was quite small. At a long distance from the robot, the humans can detect small angles more easily than the small displacements in 3D space. The tracking system

permitted the typical pick-and-place task to be completed by the operator including: approach object, grab object (closed gripper), lift object and move it over the target, place object on target, release object (open gripper), move away from object. The experiment did not aim to accurately assess the tracking accuracy. An alternative experiment using an independent tracking method, such as using inertial sensors placed on the hand, would have to be used.

The ability to use the teleoperation system could be improved using indirect feedback to the user from cameras at the robot site, or using a robot vision system to accurately align an object on a target. Both of these methods have already been partly developed (Kofman et al., 2005) and will be the subject for future work. This was not the focus of this research. Alternative arrangement of the robot manipulator with respect to the operating space could be investigated as well. The users performed the tests without explicit training in grabbing, relocating, aligning, and releasing the object. They did not experience any noticeable time lag between their hand motion and the motion of the robot manipulation. The users were able to perform real-time teleoperation with natural hand-arm motions in a natural indoor laboratory environment with slight clutter. Tracking of the three markers was accomplished at 30 fps to meet designed requirements.

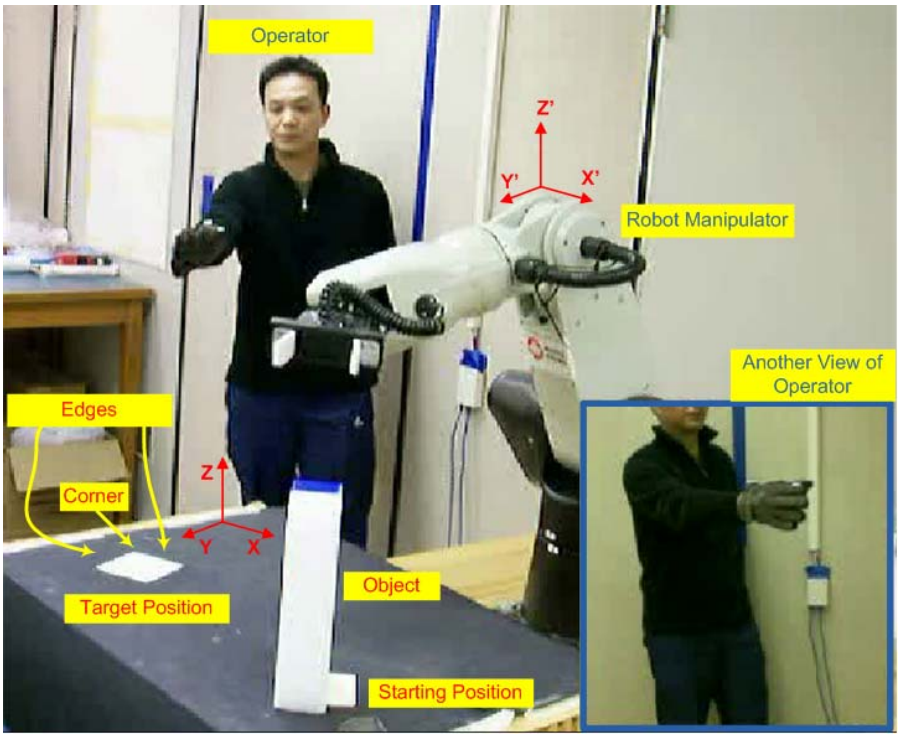

Fig. 3.9 Experimental setup for teleoperation of the robot manipulator. The task is to remotely control the robot manipulator to pick up the object form the starting position by grabbing, moving, and releasing the object on the target using direct visual feedback. The operator controls the robot by moving their hand and directly observing the motion of the robot manipulator. The object, a foam block, is initially in the starting position and it is to be placed with two edges aligned on the target.

(A) Teleoperation starts and operator is ready to grab object.

(B) Object is grabbed

(C) Object is being moved.

(D) Object is aligned at the target position.

(E) Object is released on the target and teleoperation is completed

Fig. 3.10 Teleoperation of the robot manipulator using direct observation and visual feedback. Actions are performed from (A) to (E) for a complete pick-and-place task by teleoperation.

Table 3.1 Positioning error in teleoperation for first operator.

	Positioning Error		
Test	X (mm)	Y (mm)	Z rotation (deg)
1	-7.0	6.5	-1.0
2	-6.5	6.0	0.0
3	-12.0	2.0	-2.5
4	-2.0	2.0	-1.5
5	-10.0	0.0	1.0
6	-15.5	6.0	1.0
7	-15.0	3.0	-2.0
8	3.0	7.0	6.5
9	4.5	3.5	3.5
10	10.0	12.0	9.0
Mean Absolute Error	8.6	4.8	2.8
Standard Deviation	8.7	3.4	3.8

Position and orientation errors are the differences between the final object position and orientation from the predetermined target corner position and orientation, respectively.

Table 3.2 Positioning error in teleoperation for second operator.

	Positioning Error		
Test	X (mm)	Y (mm)	Z rotation (deg)
1	9.5	6.5	2.0
2	-15.0	22.0	-7.0
3	-22.0	9.0	-5.0
4	13.5	11.5	9.0
5	0.0	13.5	1.0
6	-12.5	3.0	2.0
7	-7.0	10.0	6.5
8	7.0	21.0	-3.5
9	0.0	13.0	8.0
10	0.5	18.0	1.5
Mean Absolute Error	8.7	12.8	4.6
Standard Deviation	11.4	6.1	5.4

Position and orientation errors are the differences between the final object position and orientation from the predetermined target corner position and orientation, respectively.

Table 3.3 Positioning error in teleoperation for third operator.

	Positioning Error		
Test	X (mm)	Y (mm)	Z rotation (deg)
1	-15.5	2.0	10.0
2	2.0	8.5	-4.0
3	-15.0	-11.5	7.0
4	-13.0	-20.0	-4.0
5	-10.5	-8.0	-4.0
6	-9.0	5.5	15.0
7	-3.5	10.5	5.5
8	-6.0	1.0	-4.0
9	-8.0	5.0	4.0
10	8.0	4.5	6.0
Mean Absolute Error	9.1	7.7	6.4
Standard Deviation	7.5	9.8	6.8

Position and orientation errors are the differences between the final object position and orientation from the predetermined target corner position and orientation, respectively.

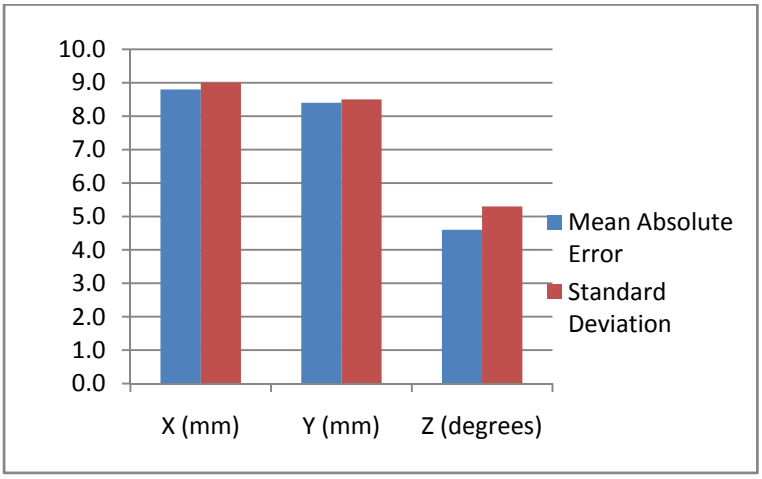

Fig. 3.11 Position and orientation errors in teleoperation for all tests combined.

3.6 Conclusion

The method for three-dimensional hand tracking based on physical markers using three calibrated cameras was successful in permitted fast tracking for real-time teleoperation of the robot manipulator. The proposed hand tracking scheme is able to deal with marker occlusion and marker merging during normal teleoperation of an object manipulator task. Marker occlusion can be solved even for all three markers missing in one of the three views by employing the camera pair utilization scheme. The multi-threshold marker detection algorithm enables hand tracking under normal indoor environments with non-uniform

lighting distribution condition, as well as occlusion of light sources during object manipulation operation. The proposed score-based marker matching and labeling algorithm provides an expandable mechanism suitable for a real-time hand tracking system.

Chapter 4

Surface-Geometry Measurement

4.1 Surface-Geometry Measurement System Design

A laser ranger sensor consists of a laser projector and one or more cameras. The proposed object surface-geometry measurement system utilizes a multi-line full-field laser projector to project multiple laser lines onto an object to be measured, and a stereo camera pair to capture images of the deformed light patterns formed on the object from two different views. The concept behind the application of structured light for 3D measurement can be summarized as follows:

Range sensor calibration

1) Cameras are used to acquire image data of known calibration points along the paths or planes of the laser projection in a controlled physical setting.
2) With the acquired image data and predefined information about the physical setup, a mathematical relationship can be derived from the geometry transformation between the 2D image coordinates (i, j) and the 3D object coordinates (x, y, z) in the 3D world space.

3D surface reconstruction

3) During an object surface measurement, with the relative positions and angles between the cameras and laser projector unmodified from Step 2, images of an unknown object with laser patterns projected onto it are acquired.
4) Using the same mathematical relationship determined in Step 2, 2D image data points of the light patterns deformed by the object to be measured are converted into their estimated 3D world coordinates.

The determination of the relationship between 2D image points and 3D object points is called *range sensor calibration*. Range sensor calibration involves the first two steps summarized above. Once the range sensor is calibrated, the 3D surface reconstruction (steps 3 and 4) of an object can be carried out for any image data point acquired by projecting onto the object, the same pattern used in the calibration. The laser-sensor calibration is the focus of this 3D laser object surface-geometry measurement component of this research.

Figure 4.1 is a photograph of the laser range-sensor mounted on the robot manipulator and Figure 4.2 is a close-up view. Figure 4.3 is the stand-alone ranger sensor detached from the robot manipulator for better illustration. Two cameras sit at the two ends of the range sensor. The laser projector is located in the centre, and projects multiple vertical laser lines into object space. The two cameras and the laser projector together form a *multi-line full-field laser two-camera range sensor*. This full-field range sensor can also be seen as the union of two individual *multi-line full-field laser single-camera range sensors* and a stereo vision

system. One sensor is formed by the left camera and the laser projector (the left sensor) while the other sensor is the right camera and the same laser projector (the right sensor). Both calibration and 3D reconstruction are performed on both of the two single-camera sensors simultaneously. The stereo vision system is composed of the two cameras only without the laser projector.

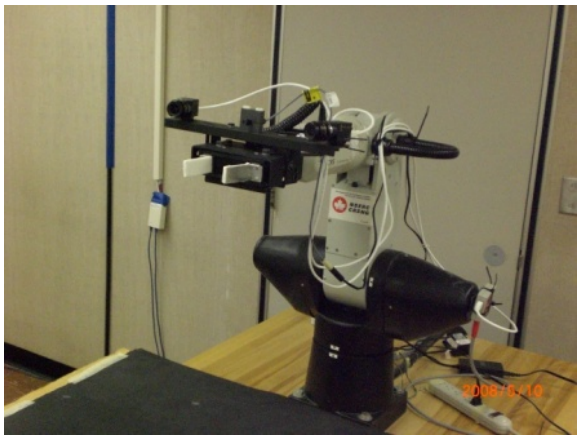

Fig. 4.1 Robot manipulator with range sensor mounted on the end effector.

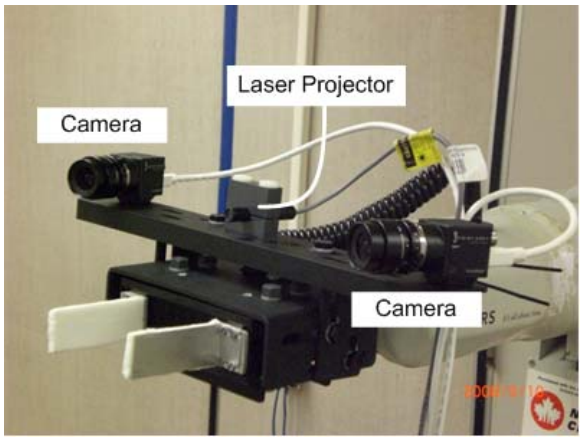

Fig. 4.2 Close-up view of range-sensor mounted on the robot end-effector.

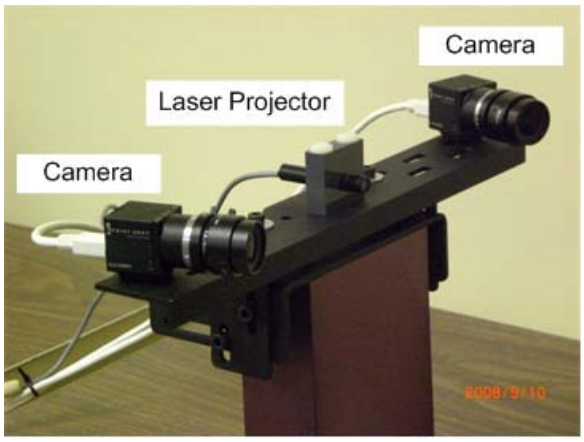

Fig. 4.3 Stand-alone range sensor detached.

A traditional laser scanner usually consists of only one camera and one laser projector. The laser projector typically projects only a single laser line. In order to scan the entire surface of an object, either the object or the laser projector is moved across the object at fixed intervals. This moving process applies to both the calibration and the reconstruction processes, and is time demanding. Hence it is not suitable for time-lag sensitive applications as in this research. The use of a multi-line laser projector simplifies the measurement process by projecting multiple laser lines onto the object at the same time to eliminate the need to move the laser projector or the object being scanned to perform a measurement. However for objects with more complex surface geometry, especially for objects with large concavities or convexities, only one camera may not be able to see the entire surface clearly from its viewpoint and light patterns may be occluded by the surface. A second camera permits more laser light patterns in a wider region to be captured than a single camera sensor would.

During the surface-geometry measurement, the proposed full-field range sensor is mounted on top of the end-effector of the robot manipulator and it serves as a vision sensor. Whenever the robot is instructed to learn about an object, it will perform the following actions in sequential order:

a) The robot manipulator moves its end-effector to the front of the object to be measured.

b) The laser projector projects laser patterns onto the surface of the object.

c) Images of the object with the light patterns formed on the object are taken simultaneously by the stereo cameras.

d) The 2D image coordinates of the laser patterns are extracted from the acquired images.

e) The 3D object surface is reconstructed using the 2D to 3D relationship derived from the calibration process.

f) The object surface geometry is then stored in computer, and it can be retrieved later in order to perform object modeling or object identification and recognition.

For Step (e), the object surface is actually reconstructed in 3D space using the two individual range sensors, the left range sensor and the right range sensor. The left sensor constructs the left part of the object surface while the right sensor constructs the right part. The two parts are constructed in a parallel fashion. After the two individual surface reconstruction operations, the two resulting parts are then merged into one unified surface.

4.2 Range Sensor Calibration

4.2.1 Range Sensor Calibration Setup

Just as for any other vision system, before the full-field range sensor can be used, it needs to be calibrated in order to compute the 2D to 3D mapping, to later permit the 3D surface reconstruction. In order to do that, the laser sensor must be detached from the robot manipulator and placed in a physical calibration jig. A top view of the range sensor calibration setup showing the calibration geometry of the range sensor is illustrated in Figure 4.4. Only three of the nineteen laser lines/profiles are shown for clarity: first, centre and last laser profiles. Both cameras are adjusted to directly face the calibration space centre. There are six calibration positions at equal intervals. In this setup, only six positions are used; however, more positions can be used as desired. The laser projector projects nineteen vertical laser profiles simultaneously to the object space from the front. The nineteen profiles have equal angular spacing (0.77 deg), with each profile a fixed projection angle with respect to the laser projection centre. The 3D object space volume is highlighted as a trapezoid for this top view and it is bounded by the first laser profile, the last laser profile, first position of the calibration plate, and last position of the calibration plate. The corresponding physical setup of the calibration jig is shown in Figure 4.5 a more detailed description is given below.

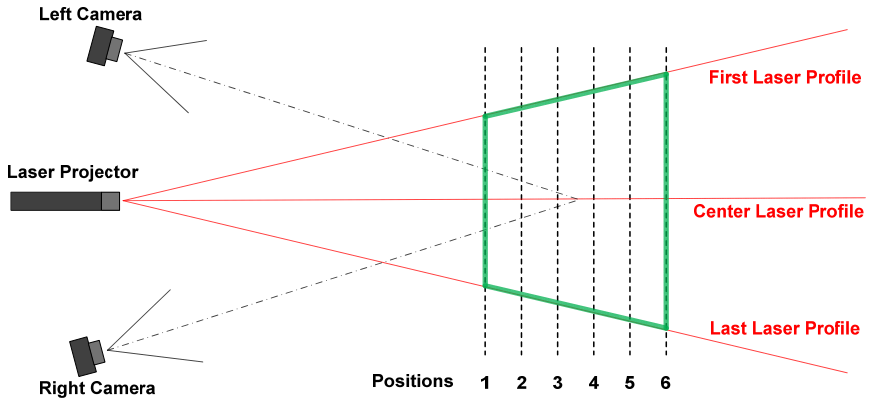

Fig. 4.4 Range sensor calibration geometry. Only three of the nineteen laser profiles are shown: first, centre and last laser profiles. The calibration/object space is bounded by first laser profile, last laser profile, first calibration position and last calibration position. The calibration space is within the visible area of both the left and right cameras. Both cameras are adjusted to face the calibration space centre. There are six calibration positions at equal intervals.

The detailed component view of the calibration jig is shown in Figure 4.5 without component labels in Figure 4.6 for clarity. The calibration jig consists of eleven components: (1) left camera, (2) right camera, (3) laser projector, (4) laser projector mount, (5) laser sensor mount, (6) laser sensor stand, (7) base bar, (8) space bar slot, (9) calibration plate mount, (10) calibration plate, and (11) space bars. Note that the calibration plate is rigidly fixed to the calibration plate mount, which can be positioned along the base bar using space bars. During the calibration process, the calibration plate will be moved to multiple designated positions with fixed intervals. These positions are precisely set by placing space bars inside the space bar slot in the base bar as place holders. Space bars (figure 4.7) are in lengths from 5 mm to 200 mm and multiple bars are used for different locations of the calibration plate. The full-field laser range sensor refers to the two cameras, the laser projector and the laser sensor mount to which these components are rigidly mounted. The laser sensor mount can be detached from the laser sensor stand of the calibration jig after the calibration process is completed and it then be mounted on the end-effector of the robot manipulator to provide it 3D vision. Both 3D surface geometry measurement by the laser-camera range sensor, and stereo vision, would be possible for object identification and recognition capability. Once the laser-camera range sensor is calibrated, the relative geometry between the cameras and the laser projector are maintained when the range-sensor mount is removed from the calibration jig. During the calibration, the range sensor mount is attached on top of the laser sensor stand, which is perpendicularly fixed to the base bar. The calibration plate is also perpendicular to the base. Multiple horizontal white lines are marked

on the calibration plate at fixed intervals of 25.4 mm (one inch). During an object measurement, the object to be measured is placed inside the object space (Figure 4.8) which is fixed relative to the range-sensor.

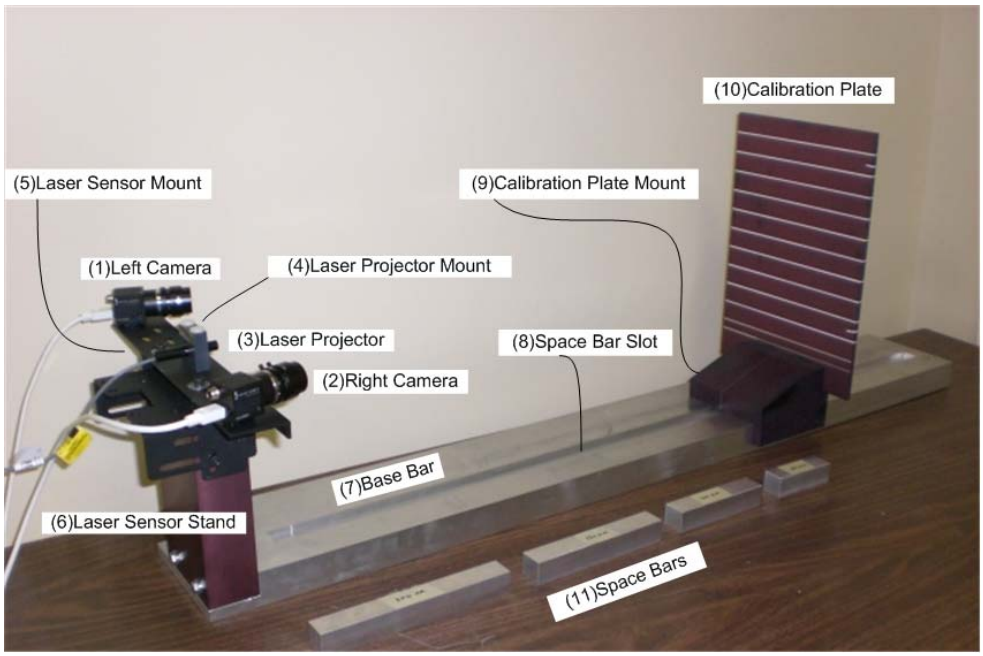

Fig. 4.5 Component view of range sensor mounted in calibration jig. The calibration jig consists of eleven components: (1) left camera, (2) right camera, (3) laser projector, (4) laser projector mount, (5) laser sensor mount, (6) laser sensor stand, (7) base bar, (8) space bar slot, (9) calibration plate mount, (10) calibration plate, and (11) space bars. The calibration place is rigidly fixed to the calibration plate mount which can be moved along the base bar for different calibration positions. Space bars are placed inside the space bar slot for precise calibration positions.

Fig. 4.6 Calibration jig with range-sensor and space bars.

Fig. 4.7 A sample of space bars of varying length. They are available in several lengths between 5 mm and 200 mm.

Fig. 4.8 Object placed in calibration volume. For surface-geometry measurement of an object, the object (here a human mask) must be placed within the calibration volume.

4.2.2 Range Sensor Calibration Methodology

Range sensor calibration involves taking images of the calibration plate using the two cameras with the calibration plate at multiple predefined positions, with the laser projector turned on and off at each position. Calibration of the sensor includes the calibration of both the left and right sensors in a parallel fashion using exactly the same process. The remaining description of range sensor calibration refers to either one sensor calibration (left or right sensor), and applies to the unified two-camera laser range sensor as a whole. The cameras used in the range sensor are RGB color cameras of resolution 680 by 480 pixels (Point Grey Research Inc). During calibration, as well as the surface-geometry measurement process, only the red band of the cameras is used rather than the three RGB channels. This is to take advantage of the bandwidth purity of the laser light in order to eliminate irrelevant ambient light noise. Figure 4.9 is one image of the calibration plate with laser patterns acquired by the left camera. There are ten horizontal white lines marked on the plate. The nineteen projected laser profiles intersect with the horizontal bright lines on the calibration plate. Due to the physical constraints (angles) of the viewing cameras, only the centre eight bright lines are considered instead of the total ten lines. Calibration points are extracted from the image at all intersections of the laser light and the eight central white horizontal lines. Figure 4.10 shows the calibration points extracted in the 2D image. With this configuration, there are a total of 152 calibration points at each calibration position.

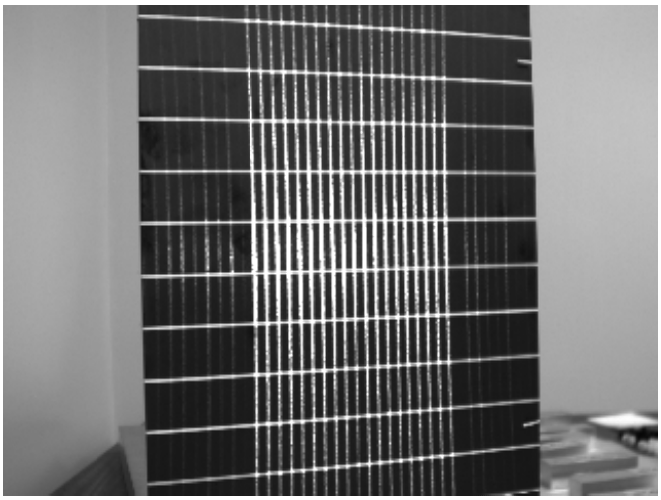

Fig. 4.9 Image of calibration plate with laser patterns at one position for the left sensor. There are 19 laser lines projected. Eight central horizontal white lines marked on the plate are used. This configuration provides 152 calibration points for each plate position (image).

Fig. 4.10 Calibration points for one position of the left sensor. There are a total of 152 calibration points for each calibration position as there are 152 intersections generated by 8 horizontal bright lines and 19 laser profiles.

By moving the calibration plate to different known calibration positions, calibration points can be acquired across the entire calibration volume (object space) shown in Figure 4.11. The corner generated by first laser profile and bottom white line of the calibration plate at the last calibration position is the world coordinate system origin. The calibration points generated from the last laser projection intersecting with the eight horizontal white lines on the

calibration plate at six calibration positions is shown in Figure 4.12. A synthetic image of the calibration points on a laser projection is shown in Figure 4.13. The outer bounding frame, which exists for each laser projection, helps in finding laser profile correspondences. The process of calibration is essentially to compute the mapping for the calibration points from the 2D image coordinates (i, j) to the 3D world coordinates (x, y, z) for all laser planes (Figures 4.11 - 4.13). There are nineteen mappings that need to be computed, one for each of the nineteen laser projections. Each mapping is computed individually in the same manner.

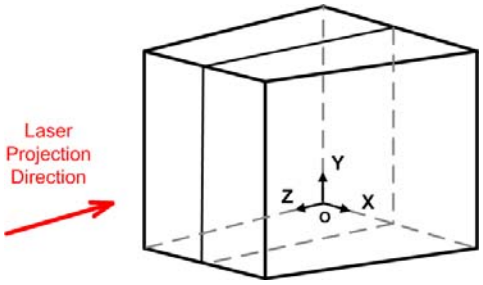

Fig. 4.11 Calibration volume - object space. The corner generated by first laser profile and bottom white line of the calibration plate at the last calibration position is the world coordinate system origin.

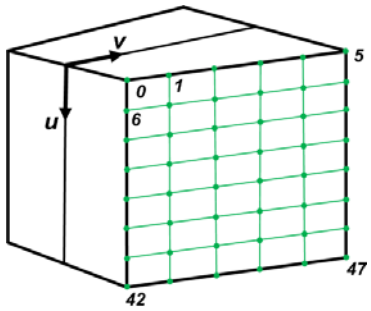

Fig. 4.12 Calibration points for the last laser projection. There are a total of 48 calibration points for a laser projection.

Fig. 4.13 Synthetic image of calibration points for one laser projection. There are 48 image blobs corresponding to 48 calibration points.

When calibration is completed, the surface of an object can be measured as follows:

1) Remove the laser sensor from the calibration jig and mount it on the robot manipulator if it is to be used for in the robot environment.
2) Place the object in the (relative) object space.
3) Project laser patterns onto the object (Figure 4.14).
4) Acquire one frame of synchronous images of the object with both cameras.
5) Extract the images of the laser profiles from the images.
6) Record the 2D image coordinates of the laser profiles in the image.
7) Transform the image coordinates into world coordinates for points of each laser profile using the corresponding mappings derived during the calibration process.
8) Reconstruct the object (surface) in world space using the computed 3D points.
9) The 3D geometry of the object surface is stored and made available for various applications.

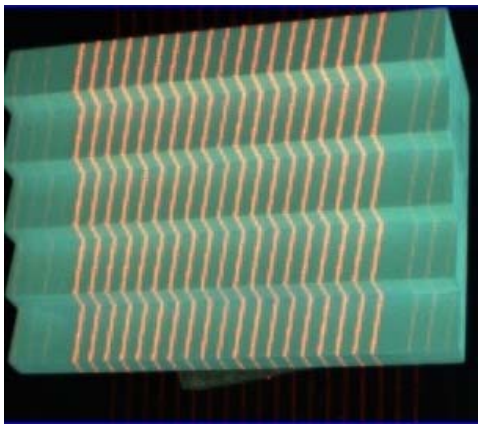

Fig. 4.14 Object with projected laser patterns

Before the calibration points are ready to be used, their 2D and 3D coordinates must be known. The 2D coordinates are essentially the (i, j) pixel coordinates of points in the image. These can be obtained by using standard image processing techniques such as image segmentation and blob detection. The 3D coordinates are the (x, y, z) values of the points with in the world coordinate system presented in Figure 4.12. Since the calibration plate is placed at known positions with fixed intervals, the z values can be determined. The horizontal white lines on the calibration plate are positioned with known vertical spacing of 25.4 mm (1 inch); therefore, the y values can be calculated. The x coordinate of calibration points are computed as described below.

The following section (Section 4.3 Conical Diffraction) introduces a method to calculate the x coordinates of the calibration points using the projection geometry of the laser projector. Later in this chapter two different approaches for 2D to 3D mapping for the laser range sensor calibration are presented. One uses Bezier surface fitting while the other uses neural network mapping. Initially, only Bezier surface fitting is considered for this mapping; however, experimental results showed that Bezier surface fitting had some limitations (discussed later) and a more reliable technique (neural network) was explored. Details are in the following sections in this chapter.

4.3 Conical Diffraction

The multi-line laser projector (StockerYale Inc.) projects nineteen laser lines simultaneously. The diverging light rays are produced by light passing through a grating, as shown in Figure 4.15. The order of projection, *m*, is counted from the centre line, *m*= 0, outward. The centre line is projected as a plane. However, the higher order projections, m≠0, diffracted by the grating, have curved projections. This unavoidable optical phenomenon, called *conical diffraction*, is commonly exhibited among diffractive optical products. Figure 4.16 shows the effect of the diffraction for the different laser projections projected onto a plane

perpendicular to the laser optical projection axis. Ideally without conical diffraction, the laser projector would generate multiple planes converging at the projection centre of the laser projector. With conical diffraction, only the centre line projection is planar, while the projection of all other lines is curved.

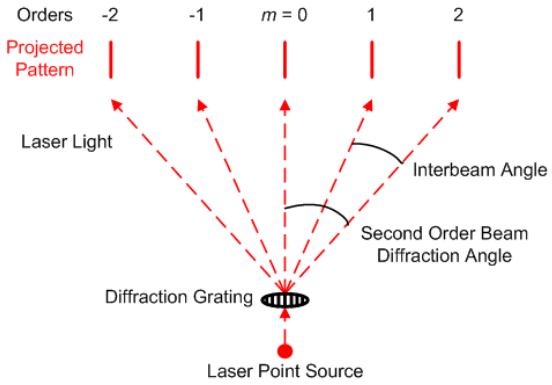

Fig. 4.15 Projection geometry of the 19-line laser projector. The inter-beam angle is 0.77 deg between all adjacent laser lines. The order *m*, is shown with the centre laser line having order 0.

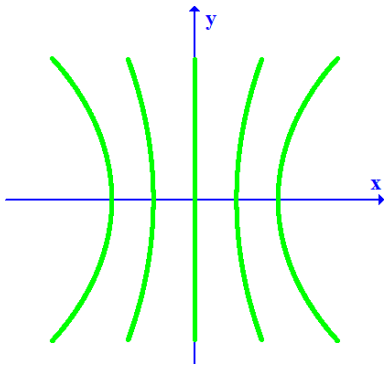

Fig. 4.16 Conical diffraction effect. Only the centre laser profile has a vertical straight line projection; the projections of the other laser profiles are curved. The curvature is greater the further from the centre. Laser projection of profiles is symmetric about *x* and *y* axes.

In order to compute the 3D coordinates of the calibration points, one must make use of the conical diffraction governing the projection geometry of the laser lines. As shown in Figure 4.17, C is the optical/projection centre of the laser projector, O is the image of the projector's

projection axis on the calibration plate, curve l is the image of a laser profile (any profile other than the central one) on the plate, and h is a horizontal white line on the plate. Point P is the intersection of curve l and line h, which defines a calibration point. B is the point projected by point P onto the x-axis. Since the central laser profile is a vertical straight line projected onto the centre of the plate, it coincides with the y-axis in this coordinate system. This coordinate system is different from the one presented in Figure 4.11. The transformation between coordinate systems is not simply one of translation. For simplicity and ease of understanding, the coordinate system in Figure 4.17 is assumed in the following discussion. Suppose that the y and z coordinates of the calibration points are known (the derivation of these two coordinates is discussed in next section of this chapter). The only unknown is the x coordinate. In other words, OC and OA are known, and OB (same as AP) is to be computed. $\angle ACP$ is the incident diffraction angle. For all incidence points P lying on line l, the incident diffraction angle will always remain the same, and it can be calculated as:

$$\angle ACP = m\,\alpha \tag{4.1}$$

where m is the order of the laser ray incidence (as in Figure 4.15), the number of laser profiles away from the centre profile, α is the inter-beam angle, which is constant for a fixed grating, which is known from the projector specifications from the manufacturer. The above governs the projection geometry in space for all laser profiles.

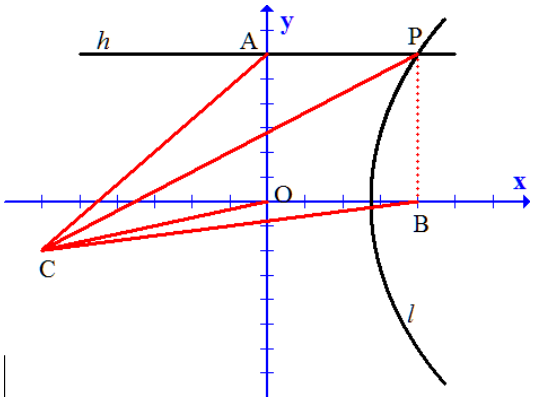

Fig. 4.17 Conical diffraction geometry. Point *C* is the optical/projection centre of the laser projector, *O* is the image of the projector's projection axis on the calibration plate, curve *l* is the image of a laser profile (any profile other than the central one) on the plate, and *h* is a horizontal white line on the plate. Point *P* is the intersection of curve *l* and line *h*, which defines a calibration point. *B* is the point projected by point *P* onto the *x*-axis.

From trigonometry:

$$AC^2 = OC^2 + OA^2 \tag{4.2}$$

$$\Rightarrow \quad AC = \sqrt{OC^2 + OA^2} \tag{4.3}$$

The x coordinate of point P can then be calculated as:

$$OB = AP = AC \tan(\angle ACP) \tag{4.4}$$

4.4 Derivation of Standoff and Offset of Laser Projection

As discussed in the previous section, the y and z coordinates for a point are needed in order to calculate its x coordinate using the conical diffraction geometry. Ideally, the y coordinate can be calculated by counting the number of white line intervals away from centre of projection, and the z coordinate can be calculated by measuring the distance between the calibration plate and the laser projector. However, because the laser range sensor is casually mounted on the laser sensor unit stand for flexibility of easy attachment and detachment, the projection of the laser projector's optical axis may not lie exactly on the centre point of the calibration plate. In other words, the coordinate system origin O is usually not at the centre of the plate. In addition, the optical centre of the laser projector is not precisely aligned with the tip of the projector nor the tail. Instead, it lies somewhere in between, and this information is not provided in the product specifications provided by the projector manufacturer. Thus, the two coordinates of a point, *y* and *z,* need to be estimated explicitly.

Two important parameters are required to permit calculation of the *y* and *z* coordinates of a point. Point *O* is the intersection of the optical axis of the laser projector and the calibration plate (Figure 4.18). The *standoff* , OC, is the distance from the optical centre of the laser projector to the calibration plate at the first calibration plate position, which is the closest distance from the optical centre of the laser projector, *C*, to the object space. Because the calibration plate will be placed at multiple positions at predetermined known intervals, the z coordinates of calibration points can be easily by determined, knowing the number of intervals from the first position, once the standoff is known. Point *A* is the intersection of the top-most white line on the plate and the centre laser line projection on the calibration plate, which is a straight line. The *offset* is the distance is defined as OA. Recall that all calibration points occur at the intersection of a horizontal white line on the calibration plate and one of the laser line projections, and that all white lines on the calibration plate are at known 25.4 mm (1 inch) intervals. Therefore, once the offset is known, the y coordinate of a calibration point can be determined using the offset and the number of intervals a horizontal line corresponding to a calibration point has from the top-most bright line. In the Figure 4.18, point P' is the intersection of another bright line h' and the same laser profile curve l shown previously, B' is the projection of P' onto the *x*-axis, and K is the projection of P' onto the *y*-axis. Plane *x*-O-*y* corresponds to the plane of the calibration plate at the first calibration position and horizontal line l is the top-most white line.

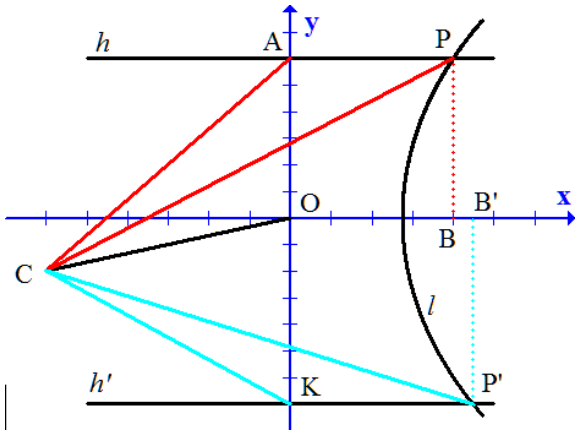

Fig. 4.18 Standoff and offset estimate. Point P' is the intersection of another bright line h' and the same laser profile curve l shown previously, B' is the projection of P' onto the x-axis, and K is the projection of P' onto the y-axis.

The standoff and offset can be calculated based on four calibration points at the first calibration position. Points A, K, P and P' are intersection points of a laser profile and a horizontal white line, and they are four calibration points for that plate position. Once the image coordinates are obtained for these points, the corresponding 3D coordinates with respect to the camera coordinate systems can be calculated using the 3D reconstruction by triangulation technique for a calibrated stereo vision system. AP and KP' can then be determined using these 3D coordinates in the camera coordinate system. From conical geometry, the diffracted angles of P and P' can be calculated from the number of inter-beam intervals of the laser projection. Since P and P' lie on the same curve l, their diffraction angles are the same: $\theta = \angle ACP = \angle KCP'$.

4.5 Surface-Geometry Measurement Using Bezier Surface Fitting

4.5.1 Bezier Surface Fitting Methodology

The Bezier surface fitting approach for surface-geometry measurement is an indirect technique for range-sensor calibration. The 2D image space is not directly mapped to the 3D world space. The 2D image coordinates of a point (i, j) are first transformed into parametric coordinates (u, v). The 2D to 3D mapping then takes place from the 2D parametric space to

the 3D world space coordinates (x, y, z). Since a Bezier surface is a parametric surface, the process of transforming from image space to parametric space is call *parameterization*. The method for parameterization (Kofman et al. 2007b) is summarized below and illustrated in Figure 4.19.

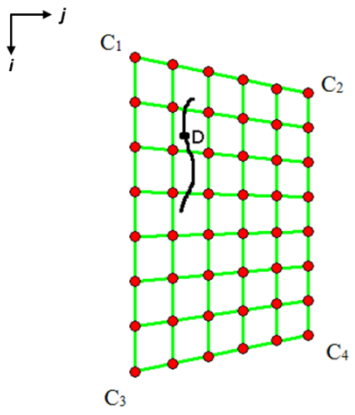

Fig. 4.19 Parameterization of one image point. C_1, C_2, C_3, and C_4 are the corner points of a grid of 2D image plane calibration points. Point D is a data point in the image.

C_1, C_2, C_3, and C_4 are the corner points of a grid of 2D image plane calibration points. Point D is a data point in the image. The corresponding parametric coordinates (u, v) for this point on a Bezier surface are determined as follows:

$$v = \frac{\left| j_D - \frac{j_{C_1} + j_{C_3}}{2} \right|}{\left| \frac{j_{C_2} + j_{C_4}}{2} - \frac{j_{C_1} + j_{C_3}}{2} \right|} \tag{4.5}$$

$$if\ (v < 0), then\ v = 0; \qquad if (v > 1), then\ v = 1 \tag{4.6}$$

$$u = \frac{\left| i_D - i_{C_1} \right|}{\left| i_{C_3} - i_{C_1} \right|}(1 - v) + \frac{\left| i_D - i_{C_2} \right|}{\left| i_{C_4} - i_{C_2} \right|} v \tag{4.7}$$

$$if\ (u < 0), then\ u = 0; \qquad if\ (u > 1), then\ u = 1 \tag{4.8}$$

Calibration for this method is done using the following steps:

1) Transform all the calibration points from image space to parametric space as above.
2) Form the F matrix based on the transformed calibration points (details are in Section 2.3 on Bezier surfaces).
3) Form the P matrix by arranging the calibration points in 3D world space (Section 2.3).
4) Compute the Bezier-surface control point matrix Q: $Q = [F^T F]^{-1} F^T P$.

The Bezier-surface control point matrix Q is the result of the calibration. It encodes all the geometry transformation mapping for all points from the 2D parametric space to the 3D world space. One matrix Q corresponds to the mapping for a single laser projection; therefore, there are a total of nineteen Q matrices that are determined to complete the range-sensor calibration. Once the complete range-sensor calibrated is done, any surface point of any laser light profile formed on the object surface, can be determined in 3D world space by knowing its corresponding 2D image coordinates. The 2D image coordinates of the point are first transformed into the Bezier space using the same parameterization method discussed above. The 3D coordinates of the point can then be computed by using the reverse equation of the one used in the calibration process:

$$P(x, y, z) = FQ \tag{4.9}$$

4.5.2 Bezier Surface Fitting Experiment

The calibration and measurement experiment of surface-geometry measurement using Bezier surface fitting is conducted on the same software platform as the one used in the 3D tracking system with the calibration jig equipped with color cameras. Both cameras have an optical lens with focal length 4 mm. Six calibration plate positions are used with 40 mm spacing between adjacent positions. Positioning is achieved using the space bars. To determine the accuracy of the calibration, five intermediate plate positions are used as test positions. An intermediate position is the centre position between two adjacent calibration positions. There are five test positions in total as there are six calibration positions. Eight horizontal white lines (25.4 mm spacing) on the calibration plate are used both in calibration and for the testing. All the points for the calibration and test are generated using the same method. Points for a position are the intersection points of the nineteen laser projections and the eight horizontal white lines on the plate. There are $8 \times 19 = 152$ points for each position, $152 \times 6 = 912$ calibration points in total, and 152×5=760 test points in total. For such a configuration, each laser plane generates $8 \times 6 = 48$ calibration points and $8 \times 5 = 40$ test points. The laser projector is positioned so that its optical centre is 352.9 mm away from the first (closest) position. Each camera is 150 mm away from the laser projector to the left or right, respectively. This enables each camera to have approximately 12 degrees of viewing angle of the object volume. The height dimension (y) of the object space in 3D space is bounded by the top-most and bottom-most white lines on the calibration plate. The width (x)

dimension is bounded by the left-most and right most laser planes generated by the laser profiles. The depth (z) dimension is bounded by the first and last calibration position.

4.5.3 Results and Discussion

The experiment was conducted using various degrees of the polynomials of the Bezier surface. (For example, a bi-cubic surface would have a degree configuration of 3x3, that is 3 in u , and 3 in v). Because there are six calibration positions and eight white lines on the calibration plate, the highest Bezier surface dimension is 7×5, in which, 7 is the u dimension while 5 is the v dimension. The set of dimensions tried was: $[3 \times 3, 3 \times 4, \ldots, 7 \times 4, 7 \times 5]$. The best set of dimensions that produces the smallest root mean square error (RMSE) was found to be 6×4. Error analysis is therefore performed for these Bezier surface dimensions. The RMSE of calibration points for the six calibration positions are shown in Table 4.1 and Table 4.2 shows the RMSE of test points for the five test positions. The *x*-dimension RMSE was found to be very small, less than 0.1 mm for both the calibration and test RMSE; the *z*-dimension RMSE was less than 1 mm; and the *y*-dimension error was greatest at 2.76 mm and 3.18 mm, respectively. In general, the test RMSE was greater than the calibration RMSE, which is expected for any experiment involving system calibration. The range sensor was developed for use on the robot manipulator end-effector for the purposes of object identification and recognition. The accuracy of the range sensor based on the test error of 3.18 mm is suitable for these purposes. The system may not be practical in surface measurement applications such as part inspection in manufacturing where higher accuracy would be needed. Analysis of the positional RMSE for the calibration and test positions (Tables 4.1 and 4.2) revealed that the error was very inconsistent over different positions. As the *y*-dimension had the greatest error, the RMSE for calibration and test points for the *y*-dimension is plotted in Figures 4.20 and 4.21, respectively, for analysis. Test RMSE ranged from 1.14 mm to 4.64 mm for test points, while calibration RMSE ranged from 0.31 mm to 5.08 mm for calibration points. For this experiment, the following pattern was observed: 1) the *y*-dimension has significantly greater error over the other two dimensions; 2) errors tend to be significantly greater toward the centre of the calibration space. These two findings were consistent with the findings of Kofman et al. (2007), although the error magnitude with a longer baseline between cameras led to lower error. Because of this non-uniform error distribution tendency and the magnitude of the error, an alternative 2D to 3D mapping approach was investigated.

Table 4.1 Calibration position RMSE for Bezier surface mapping of 2D to 3D coordinates for the six calibration positions.

	Overall Calibration RMSE (mm)		
	x	y	z
	0.057	2.761	0.532
	Positional Calibration RMSE (mm)		
pos	x	y	z
1	0.017	0.311	0.216
2	0.053	2.606	0.422
3	0.084	5.082	0.594
4	0.064	3.348	0.539
5	0.063	1.194	0.787
6	0.038	0.620	0.456

Table 4.2 Test position RMSE for Bezier surface mapping of 2D to 3D coordinates for the five test positions.

	Overall Test RMSE (mm)		
	x	y	z
	0.079	3.179	0.851
	Positional Test RMSE (mm)		
pos	x	y	z
1	0.080	3.794	0.733
2	0.085	4.424	0.762
3	0.094	4.637	0.903
4	0.079	1.966	1.103
5	0.092	1.144	1.095

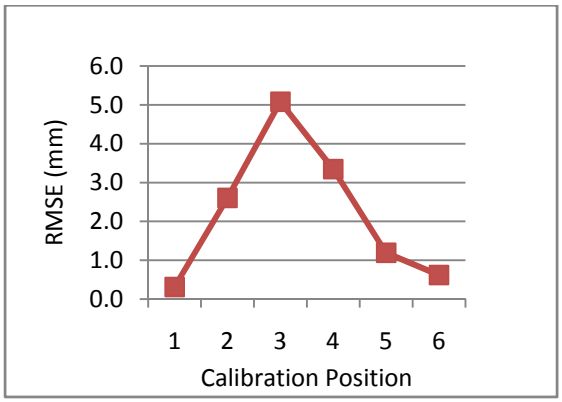

Fig. 4.20 Calibration position RMSE for Bezier surface mapping of 2D to 3D coordinates. Centre calibration positions tend to have greater errors than the end positions.

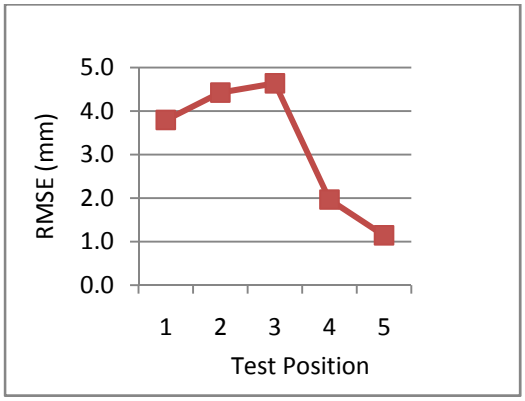

Fig. 4.21 Test position RMSE for Bezier surface 2D to 3D mapping. Centre calibration positions tend to have greater errors than the end positions.

4.6 Surface-Geometry Measurement Using Neural Networks

4.6.1 Neural Network Mapping Methodology

As discussed in the previous section, the Bezier surface fitting approach may not be ideal due to the error distribution across calibration plate positions. Error tends to be greater toward the centre of the object space. This may be a significant problem for some real applications such

as optical system measurement, as the centre position may be the location used most often. A more reliable method is therefore desirable. A neural network approach was proposed as an alternative technique for the 2D to 3D mapping. As opposed to the Bezier surface fitting technique, this neural network technique is a direct mapping approach, which means the 2D image space is directly mapped to the 3D world space without any intermediate transformation of coordinate systems.

The type of neural network used in this mapping approach is the multi-layer perceptron (described in Section 2.4). There are two nodes in the input layer corresponding to the (i, j) coordinates of the training points in 2D image space, and there are three output nodes in the output layer to represent the (x, y, z) coordinates of the training points in the 3D world space. Each laser profile corresponds to one MLP neural network of the same configuration so that there are nineteen MLP networks in total for a single-camera laser range sensor (system consists of two single-camera range sensors: left and right). This neural network approach uses the same hardware and software platform used in the Bezier surface fitting method. Since an MLP network requires more training points, both the calibration and test points used in the Bezier surface fitting approach are employed as input training points. In this setup, all calibration and test positions from the Bezier surface fitting approach become training positions in this approach. This means there are $(6 + 5) \times 8 = 88$ (six calibration positions, five test positions and eight white lines on the plate) training points for each laser projection. All the neurons (perceptrons) in the MLP are similar and have the same activation function. The MLP networks are fully connected in a way such that each node in a layer connects to all nodes in the previous and next layers if it exists. The training of a network is essentially the computing of the network connection weights for all nodes in the network. After a network is trained, its connection weights encode all the 2D to 3D geometry mapping information. The (x, y, z) coordinates of any point in 3D space can be reconstructed provided the (i, j) 2D image coordinates of the point are known. This is done by taking the 2D coordinates as network input, feeding the inputs into the corresponding MLP, and computing the outputs in terms of (x, y, z) using the trained network weights.

4.6.2 Neural Network Mapping Experiment

Since each neural network only had a small coordinate data set (only 88 input-output pairs for one laser projection as opposed to at least 200 pairs for robust neural network training), the network configuration was optimized based on the network performance measured by stratified k-fold cross-validation. In this approach, k was set to 11 for the number of calibration positions, and training and test data sets were partitioned dynamically during network training with 10 folds for the training data set and 1 fold for test set. Experiments for the neural network 2D to 3D mapping was conducted using different neural network configurations (e.g. varying the dimensions of hidden layer(s) and types of activation function). The number of hidden layers was chosen from 1 to 2 with 3 to 200 hidden nodes for each layer. Two different activation functions were selected: Gaussian and Sigmoid. The best dimensions for each configuration were recorded for analysis. All networks converged successfully. It was found that with sigmoid activation function, and network configuration

with 43 hidden nodes on each hidden layer produced the best result for both a single hidden layer and two hidden layers. For the Gaussian activation function, the single hidden layer network produced the best result with 33 hidden nodes while the two hidden layer network had the best result with hidden node dimension of 33×33. The number of cycles through the training data was set at 10000. Repeated experiments determined that increasing the number of cycles through the training data produced no significant improvement in reconstruction error. To evaluate the practical use of this surface-geometry measurement system using the neural network surface mapping, an additional experiment was conducted on a real object to regenerate its surface.

4.6.3 Results and Discussion

The spatial, *x*, *y* and *z* RMSE for the best results for different network configurations are shown in Table 4.3, and Figure 4.22 shows this graphically. The first column of the table shows the four different neural network configurations, which include Gaussian activation function with a single hidden layer, Gaussian activation function with two hidden layers, sigmoid activation function with a single hidden layer, and sigmoid activation function with two hidden layers. The second to fourth columns contain errors for the *x*, *y*, and *z* dimensions respectively. Two trends can be observed from the result: 1) the network configuration with two hidden layers produces better results than with the single hidden layer; and 2) the network configuration with sigmoid activation function produces better results than with the Gaussian activation function. For the *x* dimension, RMSE for all configurations are very low varying from 0.16 mm to 0.32 mm. The *y* dimension has the greatest error followed by the *z* and then *x*. This is very consistent with the findings for the Bezier surface fitting approach. For both *y* and *z* dimensions, the network configuration with the sigmoid activation function and the two hidden layers of dimension 43×43 produces the best result, 1.12 mm and 0.89 mm. In other words, this network configuration best maps the 2D to 3D coordinates for surface-geometry measurement. By considering this finding along with the low level of error in the *x* dimension, this network configuration is sufficient to be used for object surface measurement in this research.

Table 4.3 RMSE in spatial coordinates for the NN 2D to 3D mapping.

NN\Dimension	X(mm)	Y(mm)	Z(mm)
Gauss_1	0.316	5.245	3.223
Sigm_1	0.233	2.485	2.350
Gauss_2	0.122	1.247	1.720
Sigm_2	0.160	1.117	0.888

Four network configurations are compared: Gaussian activation function with one hidden layer, sigmoid activation function with one hidden layer, Gaussian activation function with two hidden layers, and sigmoid activation function with two hidden layers.

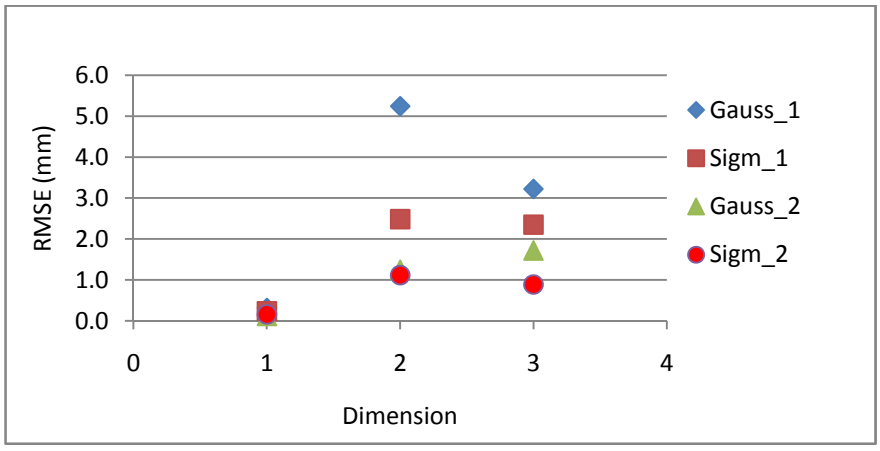

Fig. 4.22 RMSE coordinate errors for the neural network 2D to 3D mapping. The neural network configuration with two hidden layers and sigmoid activation function generates the least error in general.

The RMSE for neural network mapping using the best network configuration (sigmoid activation function with hidden layer dimensions of 43×43) is shown in Table 4.4 and Figure 4.23 for each of the eleven training positions. The RMSE in *x* ranged from 0.12 mm to 0.19 mm with a mean of 0.16 mm, in *y* ranged from 0.98 mm to 1.42 mm with a mean of 1.12 mm, and in *z* ranged from 0.76 mm to 1.12 mm with mean of 0.89 mm. For all dimensions, the error distribution curves over different positions fluctuate by a small magnitude. The 2D to 3D mapping error is consistent over all training positions within the object domain.

Table 4.4 RMSE(mm) for NN 2D to 3D coordinate mapping for all eleven positions.

	X (mm)	Y (mm)	Z (mm)
Overall	0.160	1.117	0.888
Position	X (mm)	Y (mm)	Z (mm)
1	0.187	0.977	1.117
2	0.193	0.982	0.811
3	0.143	1.013	0.693
4	0.163	0.916	0.802
5	0.138	1.083	0.870
6	0.164	1.073	0.761
7	0.168	1.098	0.704
8	0.138	1.154	0.963
9	0.129	1.158	0.991
10	0.124	1.310	1.050
11	0.190	1.423	0.893

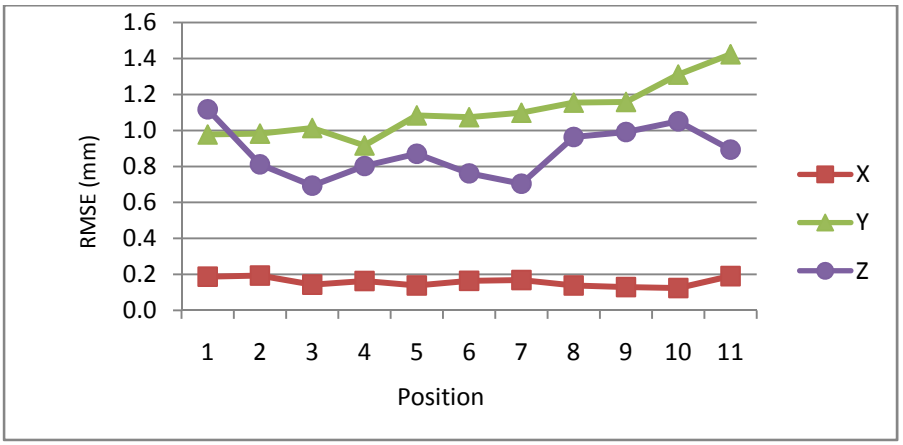

Fig. 4.23 RMSE for the neural network 2D to 3D mapping for all positions of the calibration plate. The errors are somewhat consistent across positions for *x*, *y*, and *z* dimensions.

A comparison of RMSE for the Bezier surface-fitting and neural-network 2D to 3D mapping techniques in *y* are shown in Figure 4.24. The positions of the neural network method are selected for six from its eleven positions to match with the six positions used in

calibration based on Bezier surface fitting. The centre position, which had the most significant errors present in the Bezier surface fitting method, does not have large errors with the neural-network approach. Instead, the error curve of the neural network approach is very flat compared to that for the Bezier RMSE curve. In other words, the neural network approach for surface fitting is more robust to calibration position changes than the Bezier counterpart. The neural networking surface fitting is more suitable to be used in this surface-geometry measurement component of this research project based on the error magnitude, error distribution, and usability of the useful central region.

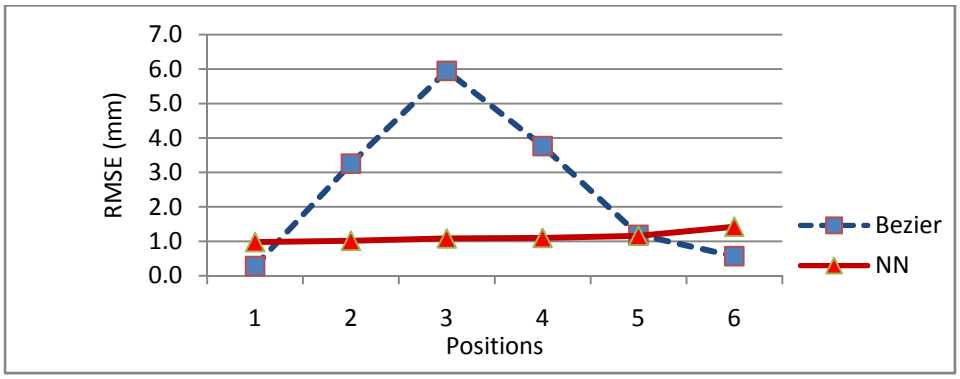

Fig. 4.24 Comparison of RMSE in *y* for the Bezier and NN 2D to 3D mapping approaches. The NN approach is significantly more consistent regarding error distribution over positions. The error magnitude of the NN approach is also smaller compared to that for the Bezier approach.

To evaluate the practical use of this surface-geometry measurement system using the neural network mapping, an additional experiment was conducted on a real object. A stepped object was placed in the object space. Since the coordinate mapping function was already trained, laser profile reconstruction takes less than a second to perform. The object measured is shown in Figure 4.25. Nineteen laser profiles were projected onto the object. The 3D reconstructed laser profiles using the neural network mapping is shown in Figure 4.26. The entire surface of the object could be regenerated using the collected 3D points on the laser profiles and surface interpolation, although this was beyond the scope of this project. Before interpolation, a smoothing filter should be applied to the regenerated surface points. The general surface geometry is regenerated as shown in Figure 4.26; however, the surface is still noisy. This is because for a good approximation using neural networks, 200 training points are recommended as a minimum. However, in this project, only 88 training points were used for each laser profile. In other words, it is believed that increasing the number of training positions will result in a better surface geometry approximation. This is considered for future work.

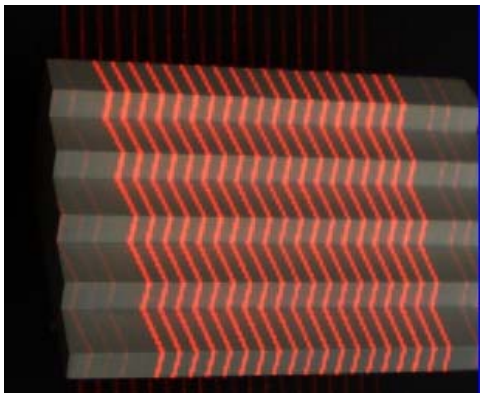

Fig. 4.25 Stepped object surface being measured.

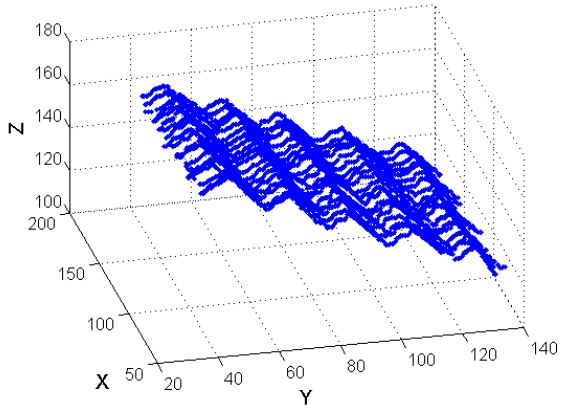

Fig. 4.26 3D reconstructed profiles of the stepped-object surface.

4.7 Conclusion

This chapter presented two approaches for surface-geometry measurement using a multi-line stereo camera range sensor. For the first approach using Bezier surface fitting, where object surface is approximated using a two-dimensional Bezier control point grid, surface-geometry measurement is determined impractical to be used in this context due to the significant magnitude of surface reconstruction error and non-uniform error distribution over positions across the depth of the calibrated volume. For the second approach, a 2D to 3D mapping technique is proposed using artificial neural networks to take advantage of the capability of neural networks being able to deal with complex nonlinear domain mapping problems. Even though there are not enough training points to train the networks for this experiment, preliminary results indicate that artificial neural networks are potentially capable of generating a good approximation of the object surface for the surface-geometry measurement component of this research project. This defines the direction for future exploration of surface-geometry measurement incorporating neural networks.

Chapter 5

Conclusion and Recommendations

5.1 Three-Dimensional Hand Tracking

A method of optical marker vision-based tracking of the human hand for teleoperation has been presented. Teleoperation of a robot manipulator was actualized using a non-contacting optical interface that enables the operator to communicate motion tasks directly to the robot using simultaneous hand motion in six degrees of freedom. The 3D hand tracking was possible using inexpensive equipment as only one hand glove and three paper sticker markers are required. The use of a trinocular vision relaxes the constraint in viewing angle imposed by binocular stereo vision that hinders the natural movement of the hand in terms of rotation inside the operation space. This provides more freedom in hand motion such that more complex object manipulation tasks can be carried out. This real time tracking permits robot-teleoperation via the vision-based human-robot interface through direct visual feedback under the operator's direct position control of the robot end-effector. The 3D hand tracking of the human hand was performed in real time at video rate by estimating the locations of the wrist joint, thumb tip and index finger tip. Through the 3D locations of the markers, the 3D position and the orientation of the hand are determined accurately enough for a human operator to control the robot manipulator to perform the task of picking up an object and placing it at a designated location. A novel expandable score-based hand tracking scheme was proposed employing dynamic multi-threshold marker detection, a stereo camera-pair utilization scheme, and marker matching and labeling using epipolar geometry and hand pose axis analysis. This enabled robust real-time hand tracking under marker occlusion and non-uniform lighting distribution environments. This tracking approach offers a viable means to robot manipulator teleoperation as it allows a natural means of communicating a whole task to a robot rather than using separate controls for limited motions as with gesture-based approaches.

The current tracking accuracy evaluation method was not able to capture the true accuracy of the 3D hand tracking system due to the constraint imposed by the physical setup that the robot arm compromises the operator's view of the object to be manipulated. It is believed that by properly arranging the robot manipulator and operating space, a significantly better measured accuracy can be achieved even when using the current evaluation method. Another more robust evaluation method that truly captures the accuracy of 3D hand tracking is desirable. Further investigation of the operator behavior in completing object tasks would be useful toward improving human-robot interaction. Other natural means of robot control, such as voice commands and gesture recognition, will also be considered in future work.

5.2 Surface-Geometry Measurement

Object surface-geometry measurement using a multi-line laser range sensor for fast acquisition of a range image using a pair of camera images captured synchronously was presented. It provides the robot-vision system with the ability to identify and recognize an object by measuring its surface. A full-field laser projector simultaneously projects multiple laser lines onto the object such that the whole object surface can be measured by a single image or pair of images, captured by one or two cameras, respectively, at fixed positions. A stereo camera pair is employed to handle potential occlusion of the object surface from different viewpoints. This surface-geometry measurement setup provides the advantages that accurate alignment of the laser projector and cameras, and knowledge of the geometry of the laser-camera setup are not mandatory. It lifts the restrictions requiring accurate measurement of the position and orientation of the laser sensor head using other devices, and eliminates requirements on the working environment and the complex viewpoint planning and acquisition imposed by other common techniques. The use of the structured light method exploits the known projection geometry of a laser projector to ease the acquisition of feature points in the calibration process. The use of stereo views and multiple profiles in each camera image view provides sufficient geometry information to permit object surface fitting for objects with complex surface-geometry.

Two approaches for the mapping of 2D image plane coordinates to 3D object space coordinates were analyzed. For the approach using Bezier surface fitting, mapping is carried out by a closed-form Bezier surface-fitting process using a two-dimensional Bezier control point grid. This approach greatly simplifies the calibration process as well as the object measurement, such that real-time 3D surface geometry measurement is possible; however, experiments indicated that this Bezier surface fitting approach is not able to generate a good approximation of the object surface due to the approximation used in the Bezier surface fitting process; higher errors tend to occur closer to the centre of the calibrated volume. This has led to the proposition of the neural network mapping approach for a more accurate surface approximation. The proposed neural-network approach directly maps 2D to 3D coordinates using a multi-layer perceptron neural network utilizing the capacity of neural networks to solve complex non-linear domain mapping problems. Preliminary results have shown that the neural network mapping approach has potential to generate a good approximation for object surface despite that there were insufficient training points for the networks in the experiments. In addition, the execution time for network surface fitting is within a fraction of a second, which will permit surface-geometry measurement in real-time.

Higher surface 3D reconstruction accuracy would be expected by using more training positions to provide more training points for each laser profile, as well as by using a longer baseline of the laser sensor to allow more image pixels to represent the object surface. Further investigation of the different types of neural networks in the mapping would be useful toward improving the accuracy of object surface approximation. Other techniques for object identification and recognition using the stereo camera in the laser range sensor will also be considered in future work.

References

Atieza, F., Alzamora, N. M., Velasco, J. A. D., Dreiseitl, S., and Ohno-Machado, L. 2003. Risk stratification in heart failure using artificial neural networks. Valencia, Spain.

Bachmann, Eric R., McGhee, Robert B., Yun, Xiaoping, and Zyda, Michael J. 2001. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. ACM Symposium on Virtual Reality Software and Technology (CRST), 9-16. Banff, Canada.

Borghese, N. Alberto and Rigiroli, Paolo. 2002. Tracking densely moving markers. IEEE First International Symposium on 3D Data Processing and Transmission, 682-685. Padova, Giugno.

Bouguet, J. 2006. Camera calibration toolbox for Matlab. Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/

Boult, Terry E., Micheals, R., Erkan, A., Lewis, P., Powers,C., Qian, C., and Yin, W. 1998. Frame-rate multi-body tracking for surveillance. Proceedings of DARPA IUW, 305-308.

Caprile, B. and Torre, V. Mar, 1990. Using vanishing points for camera calibration. The International Journal of Computer Vision, 4(2):127-140.

Chang, S., Kim, J., Kim, I., Borm, J. H., and Lee, C. 1999. KIST teleoperation system for humanoid robot. Preceedings 1999 IEEE/RSJ International Conference on the Intelligent Robots and Systems, 1198-1203.

Chaudhari, Ajit M., Bragg, R. W., Alexander, E. J., and Andriacchi, Thomas P. 2001. A video-based markerless motion tracking system for biomechanical analysis in an arbitrary environment. BED-50 Bioengineering Conference, 777-778.

Chen, F., Brown, G. M., and Song, M. 2000. Overview of three-dimensional shape measurement using optical methods. Optical Engineering, 39(1):10-22.

Clarke, T.A. and Fryer, J.G. 1998. The development of camera calibration methods and models. The Photogrammetric Record, 16(92):293-312.

Durackova, D., Grega, P. Dec, 2006. Image processing by using a novel neural network simulator. Hybrid Intelligent Systems. HIS apos;06. Sixth International Conference, 61 – 61.

Faugeras, O. and Maybank, S. 1990. Motion from point matches: multiplicity of solutions. Intl. Journal Computer Vision, 4:225-246.

Faugeras, Olivier D. 1992. What can be seen in three dimensions with an uncalibrated stereo rig?. Proceedings of European Conference on Computer Vision.

Faugeras, O. Three-dimensional computer vision: a geometric viewpoint. 1993. MIT Press.

Forsyth, D. and Ponce, J. 2003. Computer vision-a modern approach. New Jersey: Prentice Hall, ch.2.

Fukuda, O. Tsuji, T. Kaneko, and Otsuka, A. Apr, 2003. A human-assisting manipulator teleoperated by EMG signals and arm motions. IEEE Trans Robot. Autom, 19(2):210-222.

Harada, Tatsuya, Sato, Tomomasa, and Mori, Taketoshi. 2000. Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. Workshop on Human Motion (HUMO'00), 99-106.

Hartley, Richard I. 1992. Estimation of relative camera positions for uncalibrated cameras. Proceedings of European Conference on Computer Vision.

Hartley, Richard and Zisserman, Andrew. 2003. Multiple view geometry in computer vision. Cambridge University Press. ISBN 0-521-54051-8.

Haykin, S. 1994. Neural Networks. A Comprehensive Foundation. Macmillan College Publishing, New York.

Hellstrom, T. 1998. A random walk through the stock market. Licentiate Thesis, Department of Computing Science, Umea University, Sweden.

Heikkila, J. and Silven, O. 1997. A four-step calibration procedure with implicit image correction. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’97). San Juan, Puerto Rico. IEEE.

Hu, Haiying. Li, Jiawei, Xie, Zongwu, Wang, Bin, Liu, Hong, and Hirzinger, Gerd. 2005. A robot arm/hand teleoperation system with telepresence and shared control. Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 1312-1317.

Hugli, H., Maitre, G., Tieche, F. and Facchinetti, C. 1992. Vision-based behaviours for robot navigation. Proceedings of the Fourth Annual SGAICO Meeting, Neuchatel, Switzerland.

Klette, R., Schluns, K. and Koschan, A. 1998. Computer vision-three-dimensional data from images. New York:Verlag, ch.2.

Knopf, G. K., and Kofman, J. Apr, 2002. Surface reconstruction using neural network mapping of range-sensor images to object space. Journal of Electronic Imaging, 11(2):187-194.

Kofman, Jonathan, Verma, Siddharth, and Wu, Xianghai. 2007. Robot-manipulator teleoperation by markerless vision-based hand-arm tracking. International Journal of Optomechatronics, 1:331-357.

Kofman, J., Wu, J. T., and Borribanbunpotkat, K. 2007. Multiple-line full-field laser-camera range sensor. Optomechatronic Computer-Vision Systems II, Proc. of SPIE, 6718:67180A.

Kofman, J., Wu, X. Luu, T. and Verma, S. 2005. Teleoperation of a robot manipulator using a vision-based human-robot interface. IEEE Transactions on Industrial Electronics, 52(5):1206-1219.

Lathuiliere, F. and Herve, J.Y.. 2000. Visual hand posture tracking in a gripper guiding application. Proc. Int. Conf. Robotics and Automation (ICRA), 1688-1694.

Levenberg, K. 1994. A method for the solution of certain problems in least squares. Quart. Appl. Math. 2, 164-168.

Liebowitz, D. and Zisserman, A. June, 1998. Metric rectification for perspective images of planes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 482-488. IEEE Computer Society. Santa Barbara, California.

Longuet-Higgins, H. C. 1981. A computer algorithm for reconstructing a scene from two projections, Nature 293, 133-135.

Longuet-Higgins, H. C. 1984. The reconstruction of a scene from two projections - configurations that defeat the 8-point algorithm. Proc. IEEE 1st Conf. On Artif. Intell. Applications, 395-397.

Luong, Q-T., Faugeras, O.D. Jan, 1996. The fundamental matrix: theory, algorighm, and stability analysis. International Journal of Computer Vision, 17(1):43-7533.

Marquardt, D. 1963. An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Appl. Math. 11, 431-441.

Netravali, A. et. al. 1989. Algebraic methods in 3-D motion estimation from two-view point correspondences. Intl. Journal of Imaging Systems and Technology, 1:78-99.

OpenCV. 2006. Open Source Computer Vision Library. Available at: http://opencvlibrary.sourceforge.net/

Perie, D., Tate, A. J., Cheng, P. L., and Dumas, G. A. 2002. Evaluation and calibration of an electromagnetic tracking device for biomechanical analysis of lifting tasks. Journal of Biomechanics, 35(2):293-297.

Peters, C. and O'Sullivan, C. 2002. Vision-based reaching for autonomous virtual humans. Proceedings of AISB02 symposium: Animating Expressive Characters for Social Interactions, 69-72.

Postigo, J.F., Mut, V.A., Carelli, R.O., Baigorria, L.A., and Kuchen, B.R. 2000. Hand controller for bilateral teleoperation of robots. Robotica 18, 677-686.

Tezuka, T., Goto, A., Kashiwa, K.I., Yoshikawa, H., and Kawano, R.. 1994. A study on space interface for teleoperation system. IEEE International Workshop on Robot and Human Communication, 62-67.

Triggs, B., 1998: Autocalibration from planar scenes. ECCV 98, 89-105.

Tsai, R.Y. 1986. An efficient and accurate camera calibration technique for 3D-machine vision. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 364-374, Miami Beach, Florida, IEEE.

Tsai, R.Y. Aug, 1987. 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, RA-3(4).

Verma, S. 2004. Vision-based markerless 3D human-arm tracking. M.A.Sc. Thesis, Department of Mechanical Engineering, University of Ottawa, Ottawa, Canada.

Verplaetse, C. 1996. Inertial proprioceptive devices: self-motion-sensing toys and tools. IBN Systems Journal, 35(3):639-650.

Xu, G. and Zhang, Z. 1996. Epipolar geometry in stereo, motion and object recognition. Kluwer Academic Publishers.

Zeid, I. 1991. CAD/CAM theory and practice. McGraw-Hill, New York

Zhang, Z. 2000. A flexible new technique for camera calibration. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 1330-1334.