NOTE TO USERS

Université d'Ottawa • University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Siddharth VERMA

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. A. Sc. (Mechanical Engineering)

GRADE - DEGREE

Department of Mechanical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Vision-based Markerless 3D Human-arm Tracking

J. Kofman

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

J. Hayes

D. Necsulescu

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

VISION-BASED MARKERLESS 3D HUMAN-ARM TRACKING

Siddharth Verma

A thesis submitted to the Faculty of Graduate and Postdoctoral studies
in partial fulfillment of the requirements of the degree of

MASTER OF APPLIED SCIENCE

in Mechanical Engineering

Ottawa-Carleton Institute of Mechanical and Aerospace Engineering
University of Ottawa
Ottawa, Canada

Your file Votre référence
ISBN: 0-494-01628-0
Our file Notre référence
ISBN: 0-494-01628-0

ABSTRACT

Vision-based human motion tracking has been used in surveillance, human-machine interfaces in robotics and automation, virtual reality applications and biomechanics. Most techniques require markers, use a predefined motion sequence or user-intervention for initialisation, and do not process in real-time. This thesis describes a vision-based non-invasive technique for markerless real-time tracking of human-arm motion. Human-arm motion is tracked by processing images from two calibrated cameras in real-time, to estimate the positions of the joint centres of the wrist and elbow in three dimensions (3D), and to compute the 3D positions of the index finger and thumb in order to permit determination of the orientation of the hand. The hand-arm motion tracking is implemented in a human-robot interface in performing tasks of picking up an object and placing it at a target location. Tracking of the hand and arm is carried out without any prior knowledge of subject's arm length, color, width and distance from the cameras.

Keywords: Human-motion tracking, hand-arm tracking, markerless tracking, real-time, self-initialising, joint centres, edge detection, stereo vision, human-machine interface, robot teleoperation.

ACKNOWLEDGEMENTS

Many people have contributed towards successful completion of my thesis. First and foremost I would like to thank my supervisor, Dr. Jonathan Kofman, for his supervision, giving valuable suggestions and criticism throughout my research. I am really privileged to have Dr. Kofman as my supervisor. I am grateful for his help in creating the concepts and techniques of my research. Thank you, Dr. Kofman, for your support, encouragement and mentor-ship for my research and taking keen interests in my future career.

I would like to thank the academic staff for their help and support throughout my time in the University. I would specially like to thank Xianghai Wu, for helping me in setting up the human-robot interface for my experiments. Furthermore, I will to thank my fellow classmates Terris Yakimovich, Tim Luu, Jeffery T. Wu and Jeff Bottomley for their support and encouragement throughout my research work.

Financial support for this research was provided by the Natural Sciences and Engineering Research Council of Canada through research and equipment grants.

Equally important was the constant encouragement from my parents for their moral support and blessings, without which this work would not have been possible. Finally, I would like to express my grateful thanks to all my friends Abhinandan Jain, Sumant Jha, Nilesh Patel, Sriram Kumarappan, Punnet Dargar and Shallu Bhalla, for keeping me motivated throughout my studies and research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Roman Characters

c'	Midpoint of the line joining the thumb and index finger.
e	Sum-squared error in computation of best-fit sphere.
h	2D coordinate position of the index finger tip.
i	Image coordinates parallel to X-axis of the image.
j	Image coordinates parallel to Y-axis of the image.
ℓ_1	3D distance between the unknown thumb point and the known index-finger tip position.
ℓ_2	3D distance between the unknown thumb point and the known small-finger tip position.
ℓ_3	3D distance between the unknown thumb point and the known wrist position.
q	2D coordinate position of the thumb tip.
t_u	Upper-arm thickness in 2D.
u_1	i coordinate of a point in Image Plane 1.
v_1	j coordinate of a point in Image Plane 1.
u_2	i coordinate of a point in Image Plane 2.
v_2	j coordinate of a point in Image Plane 2.
w	2D coordinate position of the wrist-joint centre.
x_0, y_0, z_0	Unknown 3D position of the thumb.
x_1, y_1, z_1	Known 3D position of the index finger.
x_2, y_2, z_2	Known 3D position of the wrist joint.

x_3, y_3, z_3	Known 3D position of the small finger.
A_1, A_2, A_3, A_4	Coefficients of plane through wrist joint, thumb and index finger.
C	2D coordinate position of the centre of gravity of a blob.
$E_{n,m}[\delta, C(i,j)]$	Blob identifier array that stores the label number and position of centre of gravity for each blob m for the frame n.
$E_{ML}(x,y,z)$	3D coordinate of elbow joint computed by markerless tracking.
$\overline{E}_{ML}(x,y,z)$	Average 3D coordinate of elbow joint computed by markerless tracking technique over 10 frames.
$E_{ICR}(x,y,z)$	Elbow joint computed by the Instantaneous Centre of Rotation (ICR) technique.
$E_e(x,y,z)$	Absolute difference between elbow joint positions for markerless tracking and marker-based tracking.
(G_{ex}, G_{ey})	2D coordinates of the elbow joint.
(G_{wx}, G_{wy})	2D coordinates of the wrist joint.
$G_e(x,y,z)$	3D coordinates of the elbow-joint centre in the current frame.
$\tilde{G}_w(x,y,z)$	Uncorrected 3D wrist-joint centre coordinates.
$G_{eg}(x,y,z)$	3D elbow joint coordinate for image pair g in wrist correction.
$G_{wg}(x,y,z)$	3D wrist joint coordinate for image pair g in wrist correction.
$G_w(x,y,z)$	Corrected value of the 3D wrist-joint centre coordinates.
H_1	Image array of raw image with pixel intensity from 0 to 255.
H_2	Segmented image matrix with pixel intensity 0 or 255.
H_3	Image array of arm contour.
H_m	3x3 convolution mask used for edge detection.

L_1 - L_{22}	DLT camera calibration parameters.
Q_{1c}	Intercept of the best-fit forearm medial line.
Q_{1m}	Slope of the best-fit forearm medial line.
Q_{2c}	Intercept of the best-fit upper-arm medial line.
Q_{2m}	Slope of the best-fit upper-arm medial line.
$Q_{3,c}$	Intercept of the dividing line.
Q_{3m}	Slope of the dividing line.
$R_{IF}^{x'}$	Distance between the tip of the finger and centroid of marker in X.
$R_{IF}^{z'}$	Distance between the tip of the finger and centroid of marker in Z.
$R_T^{x'}$	Distance between the tip of the thumb and centroid of marker in X.
$R_T^{z'}$	Distance between the tip of the thumb and centroid of marker in Z.
R_w	Ulnar-radial width of the arm at the wrist.
T_I	Intensity threshold used for image segmentation.
T_A	Threshold of blob area used for noise elimination.

Greek Characters

δ_*	Unidentified blob label.
δ	Identified blob label.
$(\Im_{ex}, \Im_{ey})$	2D coordinate position of elbow separator.
$\hat{\lambda}$	True value of 3D slope of the forearm line.
$\rho_a(x, y)$	Image coordinates in Camera View 1 for wrist joint correction.

$\rho_b(x, y)$	Image coordinates in Camera View 2 for wrist joint correction.
$\hat{\rho}\ (x, y, z)$	Array of 3D points used for wrist centre correction.
σ	A line from the wrist joint to the farthest point on the hand.
ψ	3D slope calculation operator.
Δ	3D reconstruction operator.
Φ	Error in distance between the computed 3D length and true 3D forearm length.
$\hbar$	True 3D forearm length calculated in initialisation.
Θ	Best fit-sphere operator.

CHAPTER 1. INTRODUCTION

1.1 Background

There has been a growing interest in tracking human-limb motion for human-machine interfaces [1-3], robotics and automation [4], virtual and augmented reality applications [5], animation [6], surveillance [7], and biomechanics [8].

Motion tracking is a technique of determining an object's instantaneous *pose* (position and orientation), as it changes in time. Human-limb tracking involves *real-time* measurement of the position and orientation of the human limb. Human-motion tracking can be done using sensors placed on the human body [9-12] or by vision-based methods, which may involve markers [2-3, 13] or markerless techniques [14-19]. Human-limb motion tracking is achieved by the use of contacting sensors such as electromagnetic tracking devices, inertial sensors, pressure sensors and gloves instrumented with sensors [9-12]. However, these methods are invasive and may hinder the natural human-limb motion. Vision-based techniques have the advantage over sensor-based tracking techniques, as they are non-contacting, and therefore non-invasive and do not hinder the natural human-limb motion.

Vision-based tracking techniques involve processing of digital images often captured using Charged Coupled Devices (CCD) cameras. A camera image is a two-dimensional (2D) projection of the three-dimensional (3D) space. A digital image (Figure 1.1a) is a rectangular 2D array of pixels for which there is a colour intensity or grey level intensity for a monochrome image, at each point in the image array (Figure 1.1b). Processing of digital images to extract useful information such as edges, contours, and textures, and calculating the shape of the object and tracking motion sequences is referred to as digital image processing.

The level of accuracy required in the human-motion tracking depends greatly on the intended application. Existing techniques use information based on the joint-centre

calculation of the human limb for accurate motion tracking. Human-limb motion tracking used for applications in surveillance, virtual reality applications and animation, generally requires coarse tracking that is achieved by modelling the human body. Human-limb motion tracking can also be used in human-machine interfaces to control the movement of a robot, to teach a trajectory to a robot to achieve automation in industry, or for biomechanics. For the above-mentioned applications motion of the human limbs has to be tracked more accurately.

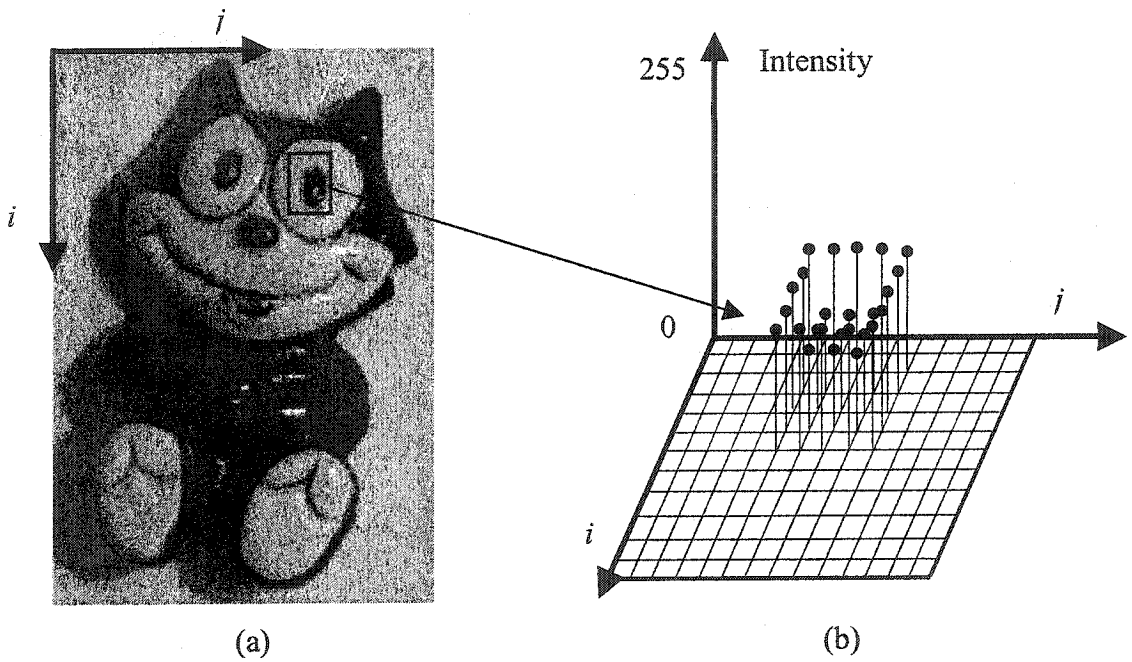


Figure 1.1 (a) Digital image and (b) graph of image intensity as a function of spatial location in image. Approximate intensity levels for some points are shown for clarity.

1.2 Problems in conventional tracking of human motion

The goal of this research is to develop a non-invasive vision-based technique for tracking hand-arm motion in real-time. The 3D position and orientation of the hand and arm should be tracked accurately enough to control a robot to perform desired tasks via a vision-based human-machine interface. The problems in conventional tracking are discussed below in the view of this goal.

Vision-based tracking of the human body has often relied on tracking markers that are manually placed at anatomical joint locations to trace the motion of the human limbs

[2-3, 13]. However, it is not always possible to install markers on the human body as they may hinder the motion and furthermore they can be occluded (hidden) during body motion. Therefore a technique that does not require any marker for motion tracking is advantageous.

Markerless tracking of human limbs motion has been a challenging topic of research for several years [14-19]. A human limb can assume various intricate shapes while moving due to the human muscle structure and soft tissue. This makes it very difficult to identify that limb in different orientations. The shape and color of various limbs are also very similar and there are no specific body features that can be used in all orientations to identify the limb. It is quite difficult to articulately identify two body parts in the image such as the forearm from upper-arm, just by viewing the image, as there are no set boundaries that can be used to separate them. Also, the problem of self-occlusion, in which a body part becomes hidden behind another part, adds to the challenge of body tracking. The biggest difficulty with a markerless approach is in estimating the joint centre positions from an image. Nevertheless, the use of a markerless technique is highly recommended for the purpose of human-limb tracking as it is non-invasive and does not have the limitation of markers becoming hidden.

To automate the process of tracking would add further difficulty, as it would mean requiring minimal user intervention at all stages. Markerless techniques have generally required user-intervention to locate anatomical joint positions in the initial frames [13, 20-21] or they required subjects to perform predefined motion to initialise the tracking algorithm [16-17, 22-26]. A self-initialising technique that requires less time in initialisation and involves minimum user intervention at all stages would be advantageous.

Many existing tracking techniques capture images and later compute the motion, as a post-process [16-17, 22, 25- 28]. One application, where real-time tracking is required, is in a human-machine interface used to perform robot teleoperation. The tracking technique should be able to compute the position and orientation of the hand and

arm in real-time such that the information computed from the tracking algorithm can be immediately used.

Most of the techniques for motion tracking require prior modelling of the complete human body or of a particular limb of interest [21, 29-33]. Generally, coarse modelling of the body is achieved using simple geometric shapes such as cylinders and spheres; however, these primitives do not represent the accurate shape of the body as the human limbs can attain different shapes in different orientations. Most of the accurate modelling techniques are either customised to a particular person [13], require a laborious initialisation sequence [16], or cannot be used for real-time person-independent applications [17].

Many of the techniques mentioned in the literature only provide 2D image information of the human motion [23-24, 34-35]. The 3D coordinate positions of the limb joint centres with respect to a fixed reference system contain more useful information compared to the 2D information of the joint centres in the camera reference plane. In robot teleoperation, this would be an essential requirement, as the robot would generally require the 3D position and orientation data to move in real space.

1.3 Goal of the current research

The current research aims to develop a non-invasive vision-based technique for human arm motion tracking that avoids problems of hindrance of natural motion that would occur with physical sensors and cables. It is also intended to develop a markerless technique for motion tracking such that there is no limitation due to occlusion of markers. The technique should be able to self-initialise, therefore requiring minimum dependence on other people, and the initialisation should require minimal time. The technique should be able to compute the 3D position and orientation data in real-time, so that the method can be used in a human-robot interface whereby a human can control the movement of a robot by their own hand-arm motion. It is also intended to use a model-free approach for motion tracking of the hand-arm movements, so that position and orientation data is calculated accurately enough for a human operator to control a robot in order to perform

required tasks. Furthermore, the technique should dynamically adapt to the varied arm length, arm width, distance from the cameras, and texture of the arm of each new subject, making it a person-independent technique. The tracking should be performed without the knowledge of these parameters and therefore without requiring their physical measurement.

The proposed body-tracking method described in this thesis is intended to be used in a human-machine interface, as a part of a robot-teleoperation research program at the Human-Machine Interfaces and Intelligent Systems Laboratory (HMIIS Lab) at the Department of Mechanical Engineering at University of Ottawa. An existing human-robot interface has been developed using marker-based tracking as a temporary interface. The current research is an extension of this work. The focus of this research is on the markerless hand-arm tracking rather than the robot-teleoperation interface. As the main intended application of the markerless tracking research was for robot teleoperation, tests with the interface have been included to demonstrate the applicability of the research. The proposed method in this thesis is limited to human hand-arm motion tracking and does not aim to achieve full-body tracking. The tracking will be achieved in a controlled and calibrated environment using two cameras at fixed positions and does not aim to achieve motion tracking in uncontrolled area. The technique is based on processing images of an unclothed body part against a black background and therefore, does not handle clothed arms, cluttered backgrounds, or occlusion by other objects.

1.4 Overview of the thesis

The research described in this thesis is presented as follows. Background information on the methods of conventional motion tracking and their limitations have been given above in this chapter to understand the problem addressed by this research. Chapter 1 also describes the proposed application of this research. In Chapter 2, a review of the recent body-tracking approaches which either address only part of the problem, or which have significant limitations is given. In the same chapter, methods of hand-arm motion tracking are reviewed. In Chapter 3, the proposed method of real-time markerless hand-arm motion tracking is described in detail. In Chapter 4, the main application of the thesis

in the human-robot interface is described. Chapter 5 describes experiments conducted to test the markerless hand tracking and the implementation in the human-robot interface. This is followed by presentation of the results in Chapter 6. In Chapter 7, implications of the results are discussed along with a comparison to the previous work. Finally, the conclusions and recommendations for further work are given in Chapter 8.

1.5 Applications of real-time markerless hand-arm tracking

1.5.1 Human-machine interface

The markerless arm-tracking technique is intended to be used in a human-machine interface between the human operator and a robot, in such a way that the human operator at the local-site can control the motion of the robot at a remote site. This human-limb-tracking based human-machine interface has wide-ranging applications areas where it is inadvisable and dangerous for a human operator to work but the task requires human decision-making skills. Applications include mine exploration, geological exploration, working in radiated environments, rescue work and bomb diffusion.

The human-robot-interface set-up shown schematically in Figure 1.2 is divided into two parts, the local-site where motion tracking is performed by calibrated cameras and the remote-site where the robot is located. The human-robot interface operates by sending the 3D coordinates of the hand-arm position and orientation to the robot and having the robot copy the motion in real-time. Four un-calibrated cameras installed at the robot-site are used to provide visual feedback of the robot motion to the user via a monitor.

1.5.2 Trajectory teaching

The vision-based markerless arm-tracking technique can be useful in industry where robots are used to automate the production cycle. It can be used to teach a trajectory to a robot with little effort in programming and setup. This could be repeated for each new task to achieve automation for complicated tasks. As an example, for a task of painting a car panel in an automobile production line, the 3D spatial coordinates of all the points where the robot should reach to paint the entire car panel, have to inputted to the robot's

controller. The process of estimating 3D points in space for the robot's movement would be very laborious and may require a long time in programming and measurement. Using markerless tracking, a subject would be asked to simply perform the task of painting the car panel once in an area calibrated by cameras, and the motion tracking application would process and save the 3D spatial coordinates of the arm movement as the spatial points required for robot end-effector to paint the entire panel. These saved points can be used repeatedly by the robot to paint the car panel in an automated painting process.

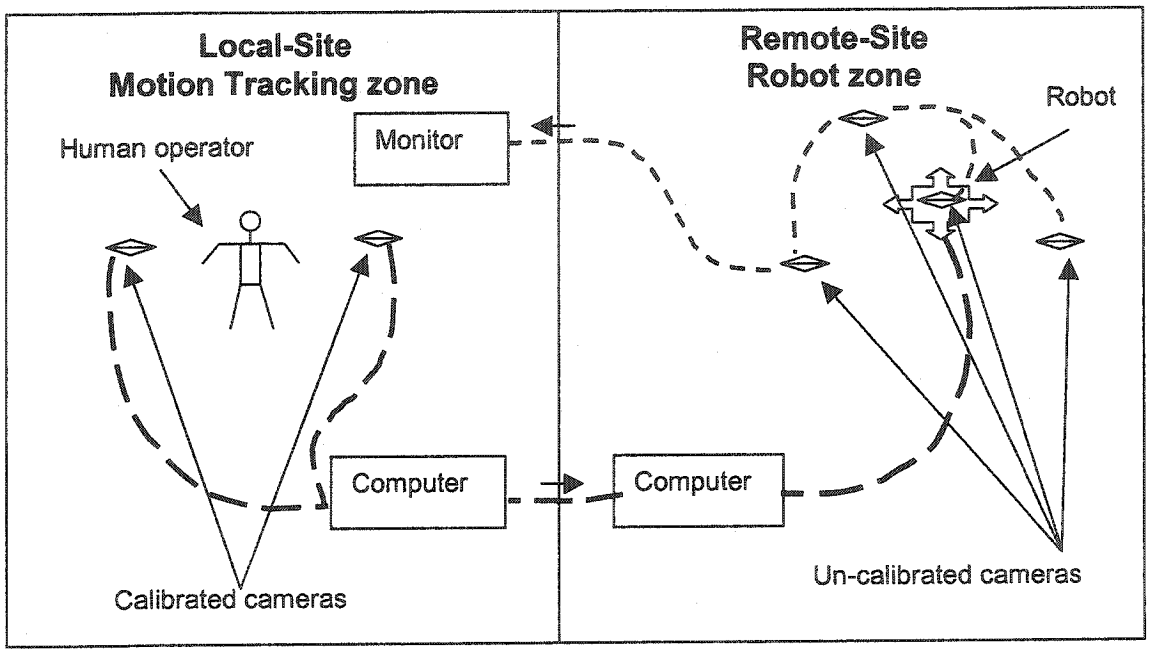


Figure 1.2. Schematic representation of the human-robot interface.

1.5.3 Biomechanics and rehabilitation

The 3D hand and arm tracking technique if extended for the whole body, could be utilized for markerless-tracking of human-motion for biomechanics especially sports-biomechanics applications. Only markerless non-contacting vision-based technique is appropriate in this case, as it is impractical to put markers or sensors on the body of the performing athletes because they would hinder the movement of the athlete and furthermore, the markers could get occluded or hidden due to the complicated motion sequence.

CHAPTER 2. LITERATURE REVIEW

Numerous research groups have addressed the problem of human-limb motion tracking for applications in human-machine interfaces, robotics and automation, biomechanics and athletic-performance analysis. However, all the techniques have limitations in their use.

2.1 Human-limb motion-tracking techniques

2.1.1 Limitations of non-vision based tracking techniques

Human limb motion tracking has been addressed using sensors such as electromagnetic tracking devices, inertial sensors and pressure sensors [9-12]. The sensors used for motion tracking are mounted on the human body and are therefore contacting and invasive to the human subject. Moreover, these sensors may hinder the natural movement of the human limbs. Ideally, the technique used for motion tracking should be non-invasive to allow natural movement of a subject's limbs.

2.1.2 Vision-based motion tracking techniques

Vision-based tracking techniques have the advantage of being non-contacting, and therefore non-invasive and not hindering the hand-arm motion, as such methods only use information acquired from processing images of the human limbs.

2.1.2.1 Limitations of existing marker-based vision techniques

Vision-based techniques often rely on tracking markers in the image that were manually placed at anatomical joint locations. White markers placed at human wrist, index finger and thumb, as shown in Figure 2.1, have been used to track the position and orientation of the human hand [2-3]. In another study, a subject is required to wear tight-fitting clothing, with body parts represented by different coloured markers and symbols to identify body parts to track their motion, as shown in Figure 2.2 [13]. However, it is not always possible to install markers on the human body especially for performing athletes as they may hinder the motion and furthermore they can get occluded during body

motion. A technique that does not require any marker for motion tracking is therefore intended in the current research.

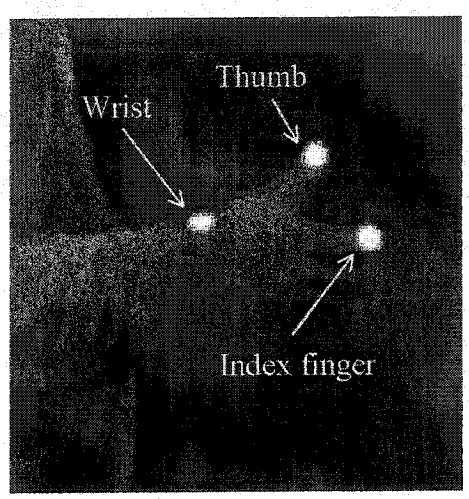


Figure 2.1. Image of hand showing white markers at the wrist, index finger and thumb [from [2]].

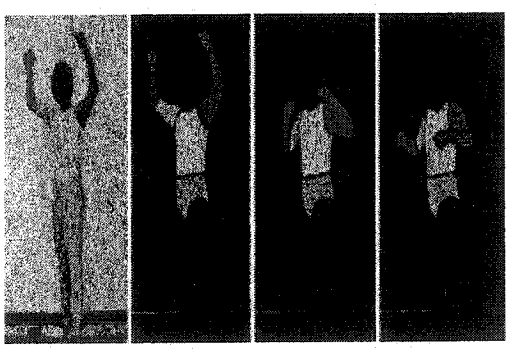


Figure 2.2 Image showing a subject wearing different coloured clothes to identify each body part [adapted from [13]].

2.1.2.2 Non self-initialising techniques

Some techniques require the user to perform prescribed motion sequences to initialise the software [22-24]. In one human-body-pose estimation technique [22], an image of the body is segmented from the background by an intensity histogram-based statistical analysis. Low-level body features obtained from the segmented image are used to train the software using unsupervised clustering via an expectation maximisation neural-network algorithm, to create groups of postures called clusters. The user is required to

perform a predefined sequence of motion to generate the required postures needed for training, as shown in Figure 2.3.

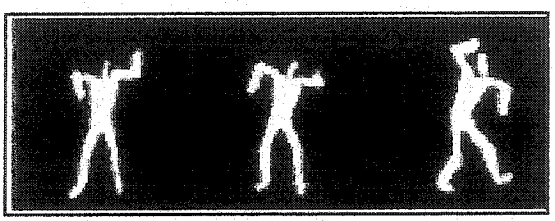


Figure 2.3 Prescribed motion sequence to initialise a tracking system [adapted from [22]].

Another model that requires a prescribed initialisation sequence uses high-curvature points on the silhouette of the body to track body motion [23-24]. Initially, a background model is generated using the background histogram of the image of the empty room. The image is segmented after subtracting the background model from the incoming image. The technique initially assumes the whole image as a single entity and aims to break it into different body parts, based on high curvature points on the silhouette of the body during motion. The largest blob in the image is considered to be the body and the rest of the image is neglected as background. A closed body contour is calculated using the silhouette of the blob in the image. The subject is required to perform a pre-described motion sequence, to reveal various body parts in a manner such that they have high curvature, as shown in Figure 2.4, to facilitate decomposition of the whole body model to individual models for each body part. The model of each body part is then fit to all new images to track the body motion. This technique has limited applications, as it requires pre-defined body information such as the distance of the body from the camera, and color of the body for accurate body modelling.

Both the above-mentioned techniques require the subject to perform a predefined motion sequence to initialise their respective software. As it is intended to develop a motion tracking method that requires minimum effort by an untrained subject with minimum time for initialisation, these techniques are not useful for the current research.

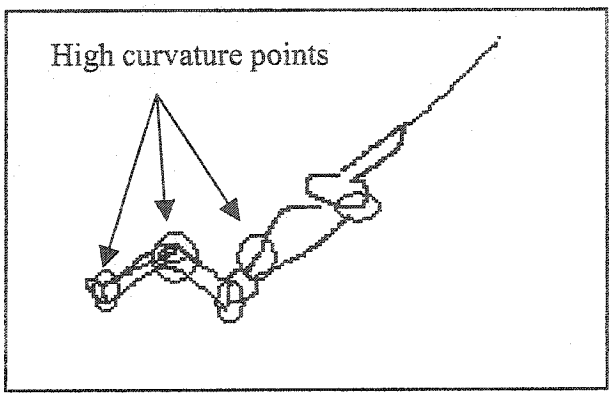
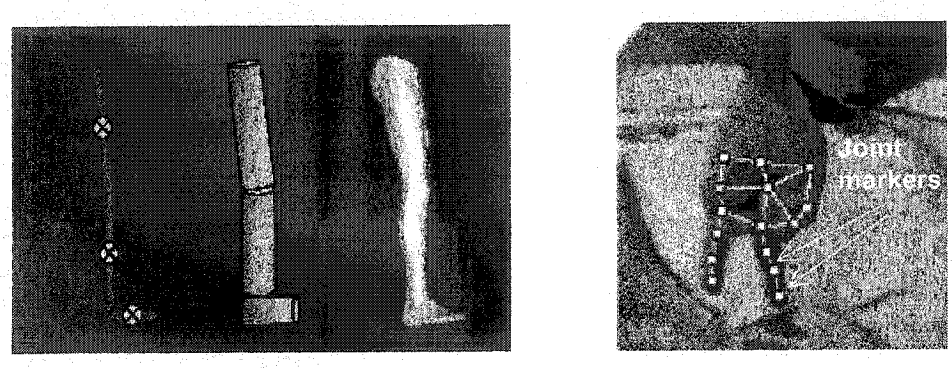


Figure 2.4 High curvature points on the body [adapted from [24]].

2.1.2.3 Tracking techniques requiring user intervention

Many markerless vision-based motion-tracking techniques require user intervention in the initial frames, to locate anatomical points or identify limb segments in the image. In one method, a manually developed 3D model customised to each person is used to track the motion of the human body [13]. The model is made by an operator, who defines the length of the limbs, color and cylindrical geometric primitives for each subject to be tracked, as shown in Figure 2.5a. This model is used to construct synthetic images of the body motion that are later matched to the actual camera images. Another technique performs 3D body motion analysis from a pre-recorded stereo-image sequence [21]. For initialisation the user has to manually identify all the joint positions such as the shoulder, hand, elbow and knee in the stereo-image pair. From the positions identified during initialisation, a body model is developed and used for body tracking. In another technique, manual input from the user has been needed for accurate initialisation of a hand-posture-recognition technique [20]. The technique requires identification of human hand joints by manually placing markers at various joint positions in the image for initialisation, as shown in Figure 2.5b.

The goal of the current research is to develop an automatic technique that requires minimum intervention from the user. As the above techniques require considerable user intervention to identify each body part in the image, these approaches cannot be readily automated and are therefore not applicable to this research.



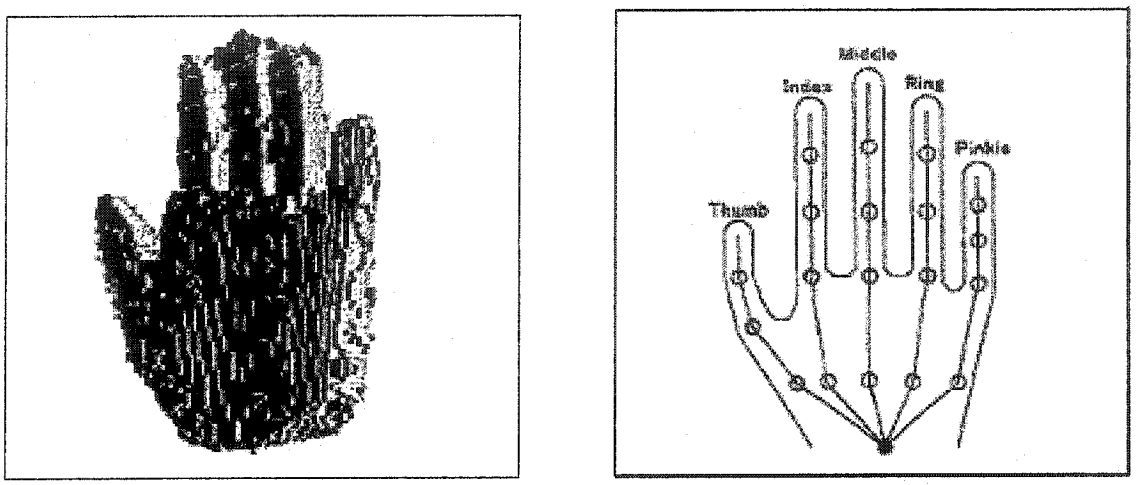
(a) (b)

Figure 2.5 User-defined models requiring manual input of (a) limb length, color and geometric primitives [adapted from [13]], and (b) joint markers [adapted from [20]].

2.1.2.4 Non real-time applications

Some techniques perform tracking not in real-time but as a post process from a recorded video-image sequence. In one of the techniques, human-body motion tracking is performed in 2D using a single un-calibrated video camera [27]. To detect an object in the incoming image, temporal and trajectory predictions are used. The temporal model is an image map of an empty room that finds connectivity between various objects in the image. The trajectory model provides an estimate of each object's position and velocity. Trajectories are predicted using an expectation maximisation neural-network algorithm. As the technique maps the image features using the neural-network, it cannot compute results in real-time. Another technique uses multiple camera images, to determine 3D information about the gesture and position of the hand [28]. The 3D surface points on the hand representing the voxel model, shown in the Figure 2.6a, are calculated using the camera images. A surface skeletal model of the hand (Figure 2.6b) is fit to the voxel model in the image. New hand positions are calculated by fitting the skeletal model recursively to the voxel model, using the data from the previous frame. Despite representing 3D position data of the hand, the technique does not provide any information about the arm motion and it does not work in real-time. Another post-processing technique performs human-body pose estimation [22] using feature matching between the image features and a trained neural network. Low-level body features obtained from the segmented image are used to train the system, using unsupervised clustering via an expectation maximisation neural-network algorithm, to create groups of postures called

clusters. A neural-network technique is then used to map a new image to one of the possible clusters by calculating the learned probability distribution and visual feature similarity between the model and the image. As the technique is based on calculating body pose using results from a neural network, it does not process results in real-time.



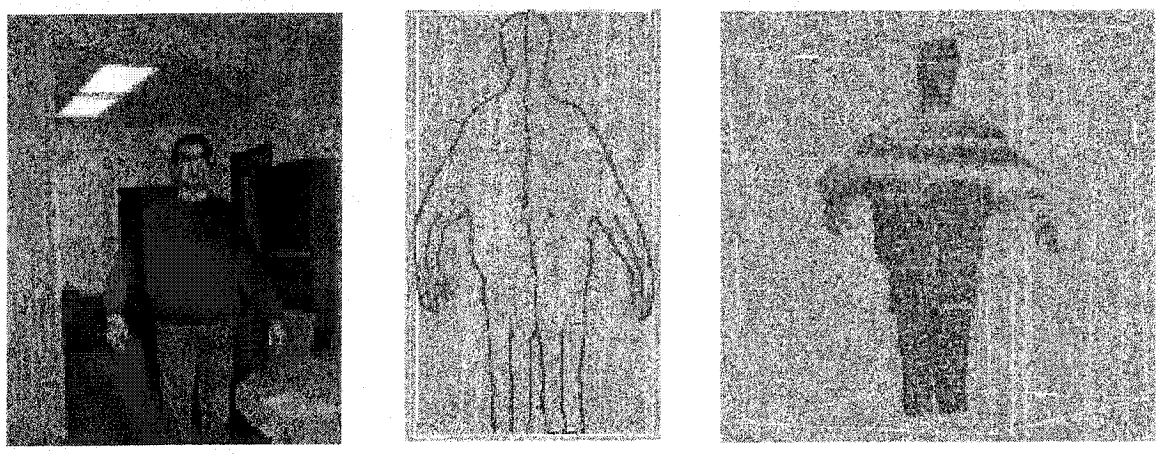
(a) (b)

Figure 2.6 Image of the hand showing (a) voxel model and (b) skeletal model. [adapted from [28]].

As it is intended to develop a technique that computes position and orientation of the hand-arm position in real-time, the above techniques are not suitable.

2.1.2.5 Model-based techniques

Some techniques use a model customised to each user to track the motion of human body parts. In one of the techniques, multiple calibrated cameras installed at fixed positions, are used to track the motion of a moving body [29]. The cameras capture high-resolution images of the subject, as shown in Figure 2.7a. Using epipolar lines from the images, the median lines of the silhouettes are calculated, as shown in Figure 2.7b. Symmetrically located circles are drawn around the median lines, to model the whole body. A similar procedure is repeated for the images from the other camera. A 3D body model is constructed by minimising the epipolar distortion among the 2D body models, as shown in Figure 2.7c. The body model used is an approximation of the actual body shape and it cannot be used for calculating the joint centres of the body parts.



(a) (b) (c)

Figure 2.7 (a) Image of the human, (b) medial lines of the silhouette of the human and (c) body model made by symmetrically located circles [adapted from [29]].

In another technique, markerless tracking of the human hand has been done in real-time using a customised model, based on the geometry of the target hand [30]. The technique assumes that the initial hand configuration is known before local hand tracking begins. The model of the hand is developed from a known hand configuration by fitting cylinders to the fingers and thumb, as shown in Figure 2.8. The technique cannot be used for hand-arm tracking as it uses a customised model for the hand and cannot be extended for the whole arm.

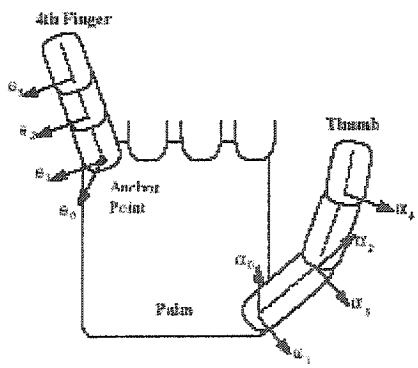
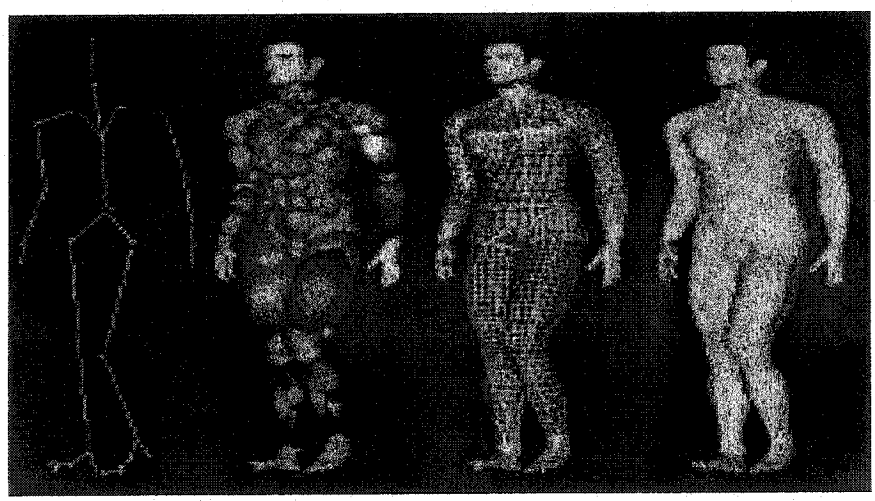


Figure 2.8 Model of the hand using fitted cylinders illustrated for the fourth finger and thumb [adapted from [30]].

In another technique [21], coarse human body tracking is performed for the purpose of animation and surveillance. A skeletal model of the body is created, as shown in Figure 2.9a, by manually identifying all the joint positions such as the shoulder, hand, elbow and knee in the stereo-image pair. Then at these positions, metaballs are fitted using least-squares-fitting criteria, as shown in Figure 2.9b. These layers of metaballs are covered by fitting a polygonal skin surface, and then shaded to depict an animated model, as shown in Figures 2.9c and 2.9d, respectively. From the metaball model of the previous image a new skeletal model is created and fit to the following image. The use of this approximate body model created by fitting symmetrical shapes to the body silhouettes can only be used for coarse tracking of human body parts and therefore only finds possible application in animation and surveillance.

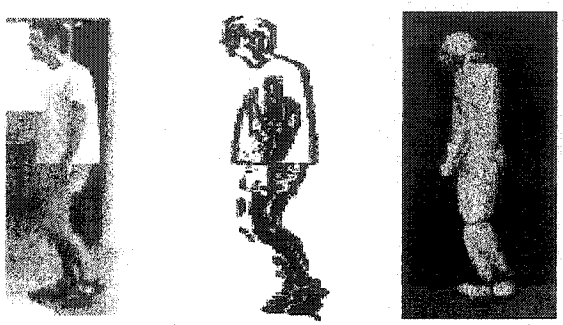


(a) (b) (c) (d)

Figure 2.9 Body model showing (a) skeleton model (b) volumetric primitives used to simulate muscles and fat tissue, (c) polygonal surface representation of the skin and (d) shaded rendering of the body [adapted from [21]].

Another coarse body-motion tracking technique is achieved by fitting a geometric model to an image of a human body (Figure 2.10a) [31]. An ellipsoidal geometric-primitive model (Figure 2.10c) is then fit to the body contour (Figure 2.10b) to represent the tracked body. As this technique is based on an approximate model of the human

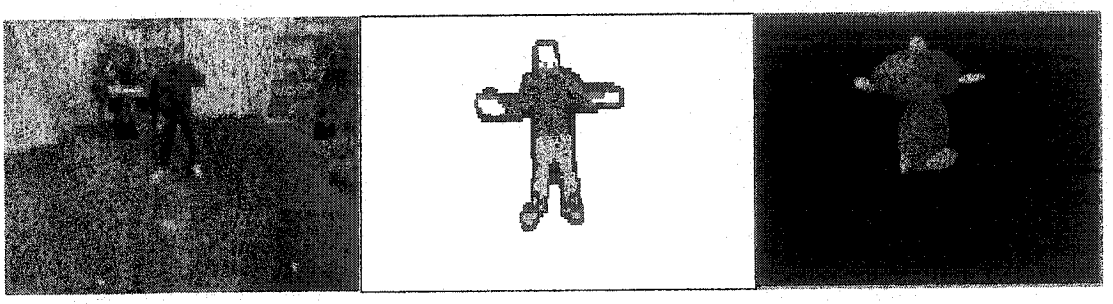
body, it cannot be used for accurate body tracking and joint-centre calculations. Moreover this technique does not compute results in real-time.



(a) (b) (c)

Figure 2.10 (a) Original image of the human, (b) contour of the human body and (c) geometric-primitive based model of the human [adapted from [31]].

One technique also referred to as Pfinder, provides real-time information of the motion of the human body in 2D space [32-33]. It uses a special Silicon Graphics architecture hardware utilising an R4400 INDY processor to enable tracking of human movement in real-time. The image of the human (Figure 2.11a) is segmented (Figure 2.11b) using the background model of the empty room. The biggest blob, which has color different from the initial background model, is considered as the human body. The system tries to identify different sections in the human model, by fitting different blobs based on the contour and color of the body, clothes, skin and head, as shown in Figure 2.11c. This work is extended to perform motion tracking of the whole body in 3D [36]. The extended technique uses a wide-baseline stereo pair of cameras to obtain a 3D model of the human body. It uses stereo information from two 2D models to create an approximate single 3D model. Although this technique provides location of the complete human body in 3D, it fails to provide any information about the 3D location of the different body parts. This is partly due to identification of different body sections based on color rather than on anatomy. This technique only provides an indicator of human presence as a whole and does not aim to perform accurate motion tracking of body parts.



(a) (b) (c)

Figure 2.11 Body modelling showing (a) input image to the system with human against background, (b) image of user segmented into blobs, and (c) model of the human obtained using seven blobs [adapted from [33]].

2.1.2.6 2D tracking techniques

Various techniques that perform body or hand tracking provide data only in a 2D image coordinate system. A three-camera system has been used to provide coarse 2D tracking of a human over a large area [34]. The image is segmented from the background by grouping regions of still-pixels (pixels with unchanged intensity) over time as background. The upper half of the body in the image is used as a blob to create a coarse 2D image model of the human. Image features such as position in the image, intensity and height of the blob are calculated for each frame and used to find a closest match of these features in the adjacent frame. This technique only provides indication of the presence of a human in the scene and does not provide any information about the 3D location of the subject or their specific body part during motion. In another technique, hand gestures are calculated in real-time using a monocular camera view [35]. It uses a partitioned sampling technique to separate the hand from the whole body and compute the gestures. Each step of partitioned sampling is to update the set of image points. This technique only provides 2D image information about the gesture of the hand and cannot be extended to provide joint centre positions in 3D. The techniques that find body parts based on high curvature points on the silhouette of the body [23-24] from a monocular image view, presented earlier, only computes 2D data of the body position in the image and cannot be accurately extended to calculate joint centres of the body parts in 3D coordinates.

2.2 3D markerless hand-arm tracking techniques

In the above section, many recent techniques to measure body motion, hand gesture recognition and human presence detection were discussed. However they all had limitations and cannot be readily used in developing a real-time technique for 3D hand-arm motion tracking. In the following section, techniques [14-19, 25-26, 37] that attempted to solve a similar problem to the current research of 3D hand-arm tracking are discussed.

2.2.1 Arm tracking using a circular-cone based model

Markerless human arm-motion tracking has been performed in 3D using a monocular camera view [18-19]. It uses a simple and approximate model of the arm, in which the upper-arm and forearm are assumed to be right circular cones, while the shoulder and elbow joints are assumed to be spherical joints, as shown in Figure 2.12. Only the arm is modelled for tracking, while it is assumed that the hand is at a fixed distance along the forearm axis. This technique calculates the position of the shoulder, elbow and wrist joint coordinate in 3D; however, the motion is performed along a plane, in front of the camera. The model of the arm is manually fit by the user to the first image and then all the model parameters are calculated from it. In the next frame, there is a direct comparison between the image and the predicted model position. The current estimate of the arm position is used to predict the arm position in the next image. An error is calculated recursively using a Kalman Filter as the difference between the actual and predicted arm position in the image. The error is used to update the arm position while fitting the model to the new image, and to calculate the new model parameters.

The model of arm is an approximate representation of the real arm, as it assumes that the arm is symmetrical and follows a general shape. The coordinates of the arm and the joint centres calculated are therefore not the actual representation of the joint centres for the arm. The positions of the arm joints are calculated in 3D but only within a plane. An accuracy test performed using the technique involved tracing a known path, and it resulted in an error of ± 5 cm in X and Y directions, while the error in Z was kept to zero.

Moreover, the technique requires the assistance of a second person to initialise the algorithm and does not provide any information about the orientation of the hand.

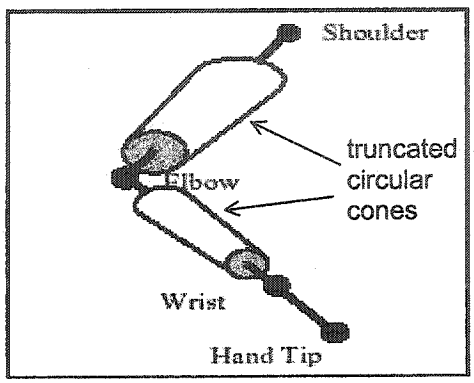


Figure 2.12 Arm modelled by truncated right-circular cones. The elbow and shoulder joints are modelled as spherical joints, and the hand tip is assumed to be along the forearm axis [adapted from [18]].

2.2.2 Arm tracking using a tapered-cylinder based model

Markerless tracking of human arm motion has been performed using a chromatic and physical model of the arm [14-15], as shown in Figure 2.13. The physical model of the arm is created by assuming it as two volumetric tapered cylinders. The arm is earlier divided into many equal-length sub-segments and the color value of each sub-segment is accumulated over a number of images to make a chromatic model of the arm. The system is self-initialising, as it looks for a predefined motion signature in the image. It means that the user will have to move their hand up and down and this predefined motion is matched to a template to initialise the system. During initialisation, the area of the image, which moves up and down, is considered to be the arm and the rest of the image is segmented as the background. By finding the instantaneous centre of rotation of the motion patch, the shoulder position is defined. The rest of the arm is fit to the physical model over many frames. The arm is tracked in the image by comparing the observed images and the predicted state variable of the physical and chromatic arm model.

The technique uses a single camera to track the motion of the arm and does not provide 3D information about the position of the arm. The symmetrical model used to

represent the human arm is an approximate representation of the real arm and cannot be used for finding accurate joint centres. Moreover, the application does not process information in real-time and does not provide any information regarding the orientation of the hand.

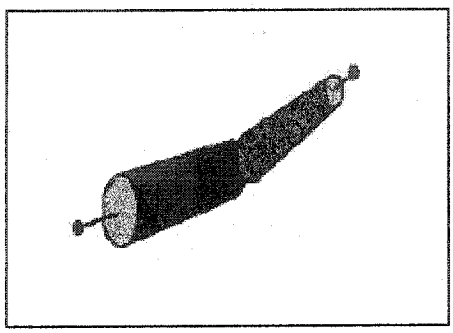


Figure 2.13 Arm modelled by volumetric tapered cylinders [adapted from [14]].

2.2.3 Arm tracking using a deformable model

Motion tracking of the body has been performed using a deformable model, by identifying the shape and calculating the joint-centre positions of the body parts [16-17]. Initialisation of the system requires all different body parts to be identified by the system. Initially, it is assumed that the model is composed of one single deformable body part. Later, based on the information from the new frames, the algorithm tries to break the model into separate body parts: the fixed zone, the bending zone and the relocation zone, as shown in Figure 2.14. This is based on the deformation between parts, due to relative motion between them. The fixed zone is the area that did not move, while the relocation zone is the area that was just translated and rotated rigidly from the previous frame. The bending zone is the area that underwent change of curvature. After the model in broken into two separate models, each model is then defined independently based on information of the fixed and relocation zones. A fuzzy clustering technique based on minimum energy criteria is used to allocate the bending zone data to both models. The 2D joint-centre position for the two models is calculated by assuming it to be a point in the bending area, which has same position in two consecutive frames between the two models. Initialisation requires a user to perform various predefined moves to model all the different body parts. The 2D body tracking is extended to track 3D positions of the joints

by recursively fitting the 2D models generated using three orthogonal camera views, to develop a 3D model of the body [25-26]. The calculated motion parameters can be animated [37] to represent the graphical motion of the 3D model. This display works as a playback from the recorded model parameters.

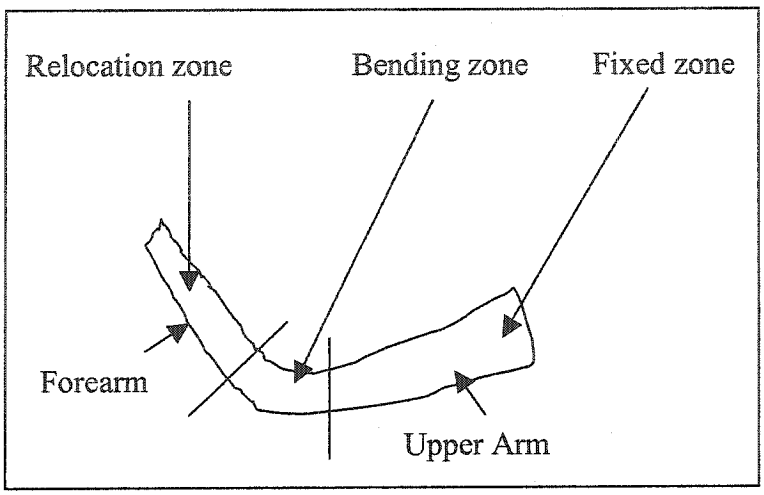


Figure 2.14 Model of the arm with separate parts: fixed, bending and relocation zones.

To initialise the system, the user is required to perform prescribed motion for each body part. The initialisation is time consuming and tedious to perform. The 3D model provides approximate 3D coordinate information, as the 2D models are loosely fit to form a 3D model. Moreover, this technique does not give real-time information therefore it cannot be used for purposes of a human-robot-interface for which the current application is proposed.

2.3. Objectives of the current research

In the above section, a few recent techniques of markerless vision-based arm-motion tracking were discussed. However, all of these techniques had significant limitations. None of the techniques discussed was able to provide a complete solution for accurate real-time tracking of human-arm motion in 3D.

The current research is aimed at developing a non-invasive vision-based technique of tracking human hand-arm motion that avoids hindrance to natural motion

from physical sensors and cables. It is also intended to make the technique markerless, so that there is no limitation due to occlusion of markers. The technique should be self-initialising, hence requiring minimum pre-defined movement by the subject, with no assistance from an extra operator. The technique should be able to compute the 3D hand and arm position and orientation data in real-time, so that the human operator can control the movement of robot using a human-robot interface. It is also intended not to use any approximate or customised model for modelling human arm, so that different shapes of the human arm, arising due to soft tissue and muscles can be accommodated. As well, the position and orientation data should be calculated accurately enough for a human operator to control a robot to perform required tasks. Furthermore, the technique should be person-independent, therefore dynamically adapting to the varied arm length, arm width, distance from the cameras, and texture of the arm of each new subject, without any prior knowledge of the human-arm geometry. The technique developed with these objectives is presented in the following chapter.

CHAPTER 3. MARKERLESS HAND-ARM TRACKING

The main application of the human hand-arm tracking is for a human-robot-interface, described in more detail in Chapter 4. For this purpose a human subject is required to perform the motion of the arm in a volume that has been calibrated for stereo-vision using two fixed stereo cameras. The two camera views for arm tracking are shown in Figures 3.1a-b. Stereo cameras permit 3D coordinates of a point to be computed from 2D images of the same point, when the correspondence between points in the two images is known. Essentially, it maps the 2D image points to 3D world coordinates. The computation of 3D points from 2D points and stereo-camera calibration are described in detail in Section 3.2.9.1. The markerless tracking of the human hand-arm is carried out by continuously processing the monocular images of the hand and arm taken from the two cameras. Human hand and arm motion tracking is achieved by determining the 3D position of the elbow joint, wrist joint, index-finger tip and thumb tip in real-time. The positions of the elbow and wrist joint are used to determine the location of the arm in 3D space, while index finger and thumb tips are used to compute the orientation of the hand. For the main application of the human-robot interface, the 3D position of the wrist joint is used to control the position of the robot arm and the position of wrist joint, index-finger tip and thumb tip are used to control the orientation of the robot gripper, as described in detail in Chapter 4. It is quite challenging to calculate the positions of the elbow joint, wrist joint, index-finger tip and thumb tip by markerless tracking, especially since the thumb can become occluded by the hand or not be visible against the palm, and the shape of the arm changes during motion due to the soft tissue and muscle structure. Markerless human hand-arm tracking is achieved in this research using two main procedures. The first is the initialisation, which is used to obtain initial 2D positions of the elbow and wrist joints, index finger and thumb tips, and the 3D forearm length from the stereo-image pair of the human arm. The second procedure is the real-time tracking, in which the same anatomical points are calculated continuously for each frame in 3D.

3.1 Initialisation sequence

Initialisation is achieved by processing 10 pairs of images of the arm and hand. Each pair of images includes views from the two cameras and each image of the pair is processed separately. The steps used to process a single image from one camera are repeated for the image from the second camera. The aim of the initialisation is to divide the arm into three different parts, namely the hand, forearm and the upper-arm, and to estimate the initial positions of the elbow joint, wrist joint, index-finger tip and thumb tip. The initialisation is performed over 10 pair of images, to compute an average value of the 3D forearm length, which is used later for the correction of the wrist joint position (Section 3.2.10). The position of the elbow joint, wrist joint, index-finger tip and thumb tip from the last (10^{th}) pair of images of the initialisation are used for tracking in the subsequent frame, which is the first frame of the real-time tracking. The complete initialisation process is summarised below and then detailed in the sections that follow.

The first step of initialisation is to capture an image of the arm. The captured image is then segmented to separate the arm from the background. After that, the contour of the arm is extracted from the segmented image by edge detection and the contour of the arm is then divided into three parts, namely, the wrist, forearm and upper arm. Image coordinates of the contour of each arm part are then determined. For the forearm and upper-arm, the midpoints are calculated using the upper edge of the forearm and the lower edge of the forearm, and upper and lower edges of the upper-arm, respectively. A best-fit medial line for the forearm and upper-arm is then calculated using the respective midpoints. The elbow joint is then calculated as the intersection of the forearm and upper-arm best-fit medial lines. The wrist joint is then calculated as the intersection of the forearm medial and dividing line, as explained below. Finally, the initial position of the thumb and index finger tips are computed. The complete process of initialisation is described in detail below.

3.1.1 Capture of arm images

To begin the tracking initialisation, the human subject assumes a simple posture, such that there is an angle between the forearm and upper-arm, with the hand above the

shoulder, as shown in Figure 3.1a from Camera View 1 and Figure 3.1b from Camera View 2. In the 2D camera image, the elbow and the wrist should appear aligned at predefined dividing lines after capture, as shown in Figures 3.1c and 3.1d. These lines are used again later in Section 3.1.4 to divide the arm contour. The hand must be positioned such that the thumb and index finger make a U-shape and are visible in both camera views. The hand shape is assumed as a U-shape to simulate the robot gripper while tracking, as shown in Figure 3.2. This general shape of the hand must remain fixed throughout the tracking process. The camera images of the unclothed arm are taken against a black background in that posture.

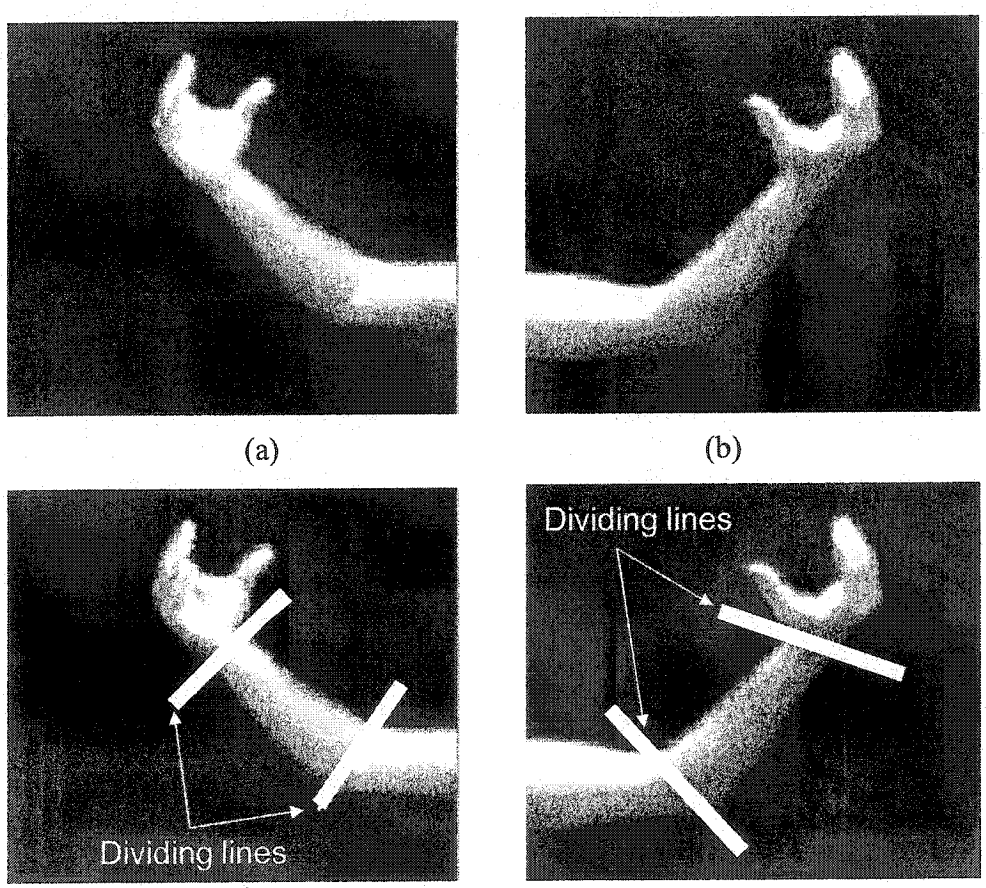


Figure 3.1 Image of the unclothed arm for (a) Camera View 1 and (b) Camera View 2 and shown aligned to the dividing lines for (c) Camera View 1 and (d) Camera View 2.

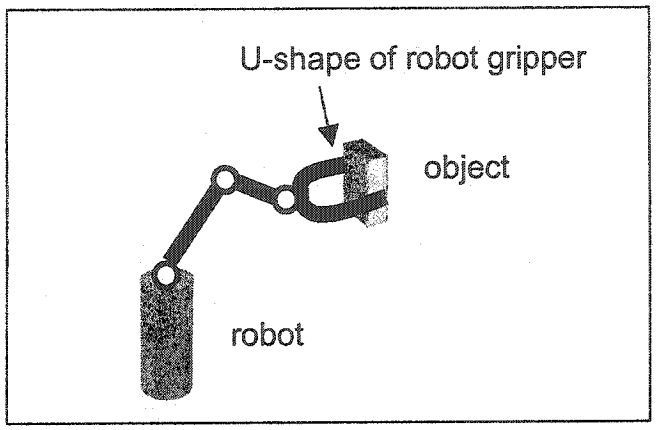


Figure 3.2 Schematic representation of the robot showing the U-shape of the robot gripper, which is imitated by the hand.

3.1.2 Arm segmentation

The arm is distinguished or segmented from the background by processing the image based on a predetermined threshold value of image intensity. The image is thresholded using the following equation.

$$\begin{aligned} &\textit{if} \quad H_1(i,j) \le T_1 \qquad \textit{then} \quad H_2(i,j) = 0 \\ &\qquad\qquad\qquad\quad \textit{else} \quad H_2(i,j) = 255 \end{aligned} \tag{3.1}$$

where,

$H_1(i,j)$ is the original 8-bit image matrix with pixel intensity varying from 0 to 255,

$H_2(i,j)$ is the segmented binarized image matrix with pixel intensity 0 or 255,

i and j are the values of the image coordinates of the image (Figure 1.1), and

T_1 is the threshold intensity.

The intensity threshold value is calculated as a value of image intensity that appears at a valley between the two peaks in the bi-modal intensity histogram of the image, as shown in Figure 3.3. As there is a big plateau between the two peaks of the histogram, the threshold value was manually adjusted to get the best thresholded image. The intensity threshold is calculated separately and only once before the start of the tracking. The threshold is not changed as there is no change in the lighting conditions. All image pixels which have an intensity value lower than or equal to the selected threshold

value T_1 are set to black or 0 value and all the pixels that have an intensity higher than T_1 are set to white or 255. The result of the segmentation process is a binarized image with the entire arm white and the region of the image apart from the arm, set to black and considered as background. The intensity threshold value T_1 is saved to be used for all images throughout the complete initialisation and real-time arm tracking. The image is also processed to remove all blobs that are considered as noise, by setting the intensity of all blobs that have an area below a certain predetermined area threshold (T_A= 4 pixels) to 0 or black. The segmented image of the arm is shown in Figure 3.4.

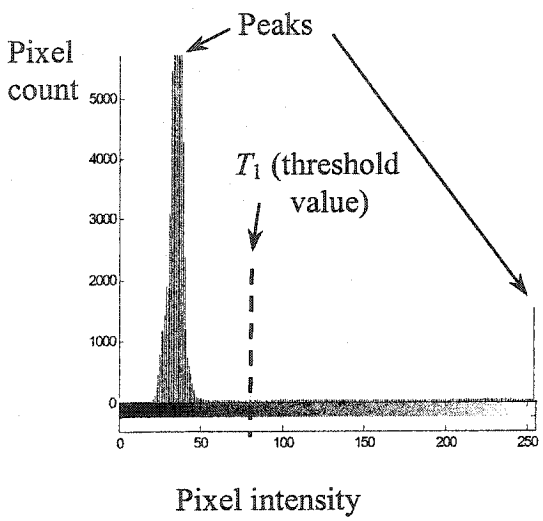


Figure 3.3 Intensity histogram used for image thresholding.

Figure 3.4 Segmented image of the arm.

3.1.3 Determination of the arm contour

To determine the outer contour of the hand and arm, edge detection is performed, by convolving a 3 x 3 mask $H_m(k,l)$, as shown in Figure 3.5a, with the segmented image $H_2(i,j)$, as follows:

For all $H_2(i,j)$:

$$If \sum_{k=-1}^{1} \sum_{l=-1}^{1} H_m(k,l)\, H_2(i+k,l+j) = 2295 \quad then \;\; H_2(i,j) = 0 \qquad (3.2)$$

where, $H_2(i,j)$ is the intensity of the pixel in the segmented image operated on. The convolution operation is further explained below.

A convolution operation of the binarized image is performed, across the image along rows (across columns) from top to bottom of the image, by moving a 3x3 pixel mask, $H_m(k,l)$, as shown in Figure 3.5a. Only if all 9 pixels have an intensity of 255, meaning that they are white, the product of the mask and image gives a value of 2295 (255 x 9) as computed in Equation 3.2. In this case, the intensity of the central pixel of the 3x3 neighbourhood (i,j) is changed to 0 or black, indicating that the pixel is not on the edge of the arm. In all other cases, when the result of the convolution at the mask position is less than 2295, the pixel intensity is kept unchanged. After convolving $H_m(k,l)$ with image $H_2(i,j)$, as shown in Figure 3.5b, the processed image with only the edge of the arm is obtained, as shown in Figure 3.6, and stored as $H_3(i,j)$.

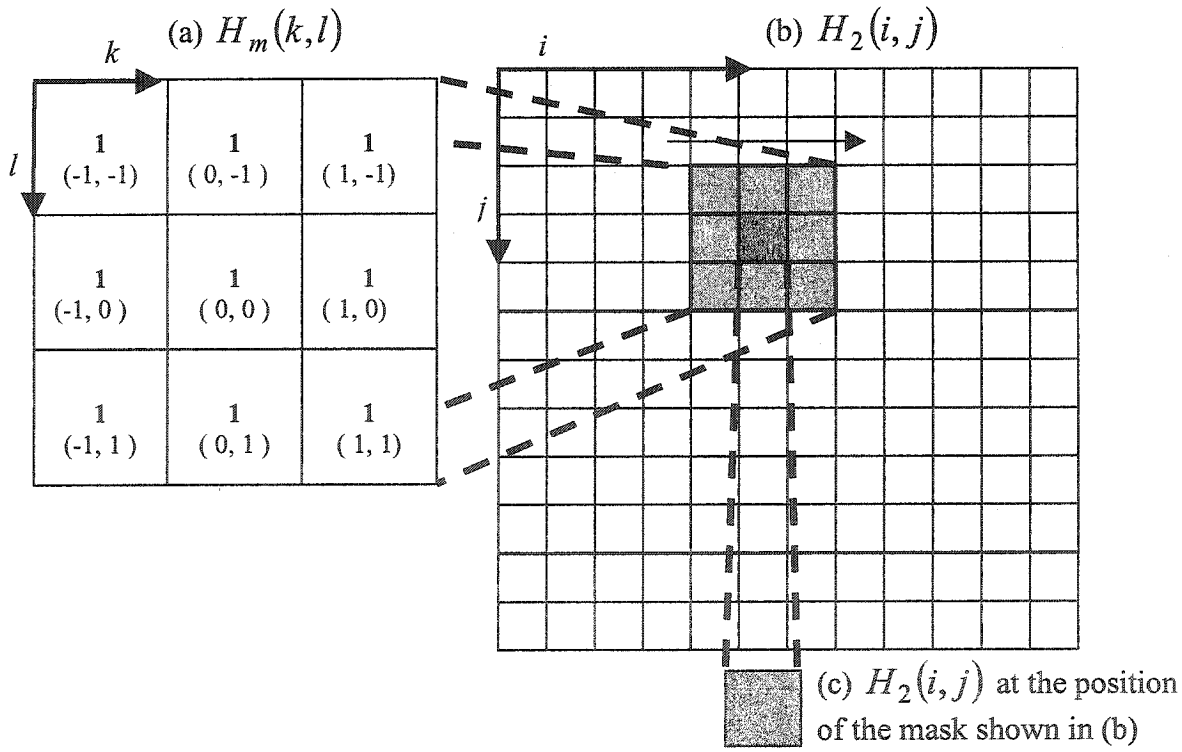


Figure 3.5 Edge detection showing convolution using (a) 3x3 mask, on the (b) segmented image and (c) pixel being operated on. In (b) the value of the multiplier is shown for each pixel with the mask coordinates in the parenthesis.

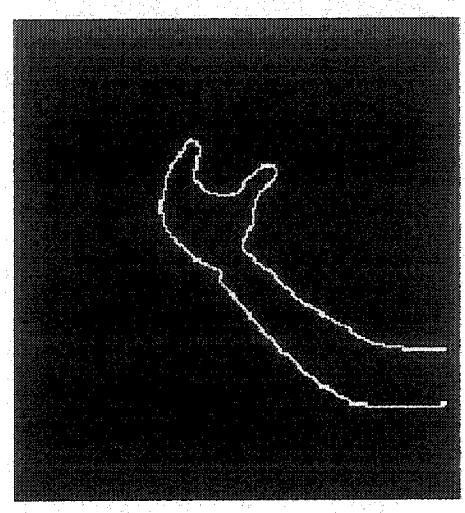
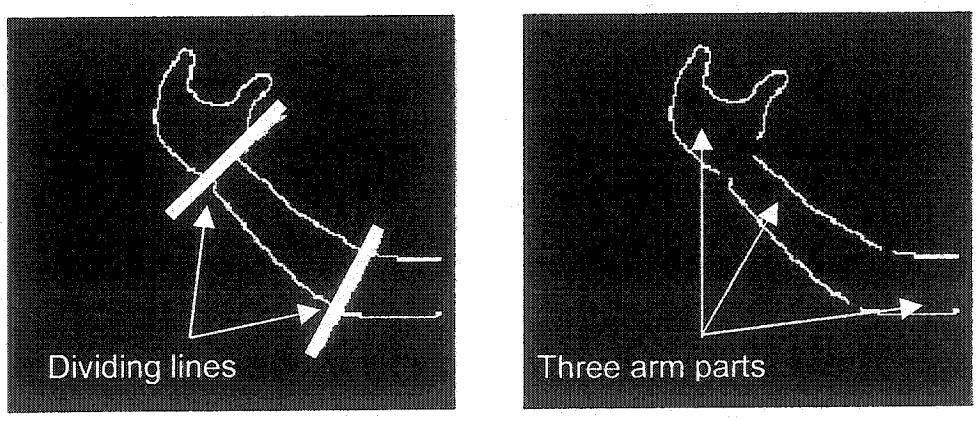


Figure 3.6 Contour of the arm computed by edge detection.

3.1.4 Division of arm into parts

The contour of the arm obtained after edge detection is divided into three parts. The predefined dividing lines, described in Section 3.1.1, are drawn on the image containing the contour of the arm, such that the arm is divided into three parts, namely the hand, forearm and upper-arm. The dividing lines, which are black, are shown in white in Figure 3.7a, for illustrative purposes. The contour of the arm, when the dividing lines are set to black, appears broken into three parts, namely the hand, forearm and upper-arm, shown in Figure 3.7b.



(a) (b)

Figure 3.7 The process of dividing the arm into three parts showing (a) dividing lines and (b) arm divided into three parts: hand, forearm and upper-arm.

3.1.5 Determination of image coordinates of arm parts

A blob identification function from Matrox Software MIL 7.0 is used to identify five different blobs, corresponding to the 1) hand, 2) upper edge of forearm, 3) lower edge of forearm, 4) upper edge of upper-arm and 5) lower edge of upper-arm, respectively, as shown in Figure 3.8. The blobs are collections of connected pixels that have an image intensity of 255. In general, blobs can have any shape, however here they have the shape of a chain or curve (Figure 3.8). The 2D image coordinates of all pixels of each blob are saved as a blob chain, which is an array that represents a connected chain of pixels in the contour of each blob. The centre of gravity of each blob chain is computed for each frame and saved to identify the same blob chain in the next frame and for further processing on an arm part or blob chain. The centre of gravity of each blob chain is indicated by a cross in Figure 3.8.

3.1.6 Computation of midpoints for forearm and upper arm

The midpoints of the forearm are calculated by a numerous distance computation approach, using all the pixel points corresponding to the upper and lower forearm blob chains using the following algorithm:

Algorithm 1: Midpoints calculation:

For all $p = 0, 1, \ldots P\text{-}1$:

For all $r = 0, 1, \ldots R\text{-}1$:

For all $k = 0$ to 1:

$$f_{(R \times p)+r,k} = \frac{\left(\Psi_{p,k} + \Omega_{r,k}\right)}{2} \tag{3.3}$$

where, $f_{(R \times p)+r,k}$ is the forearm midpoint in the image,

$\Psi_{p,k}$ is a image point on the upper edge of the forearm for point p of the upper forearm blob chain for image coordinate k corresponding to i and j,

$\Omega_{p,k}$ is a image point on the lower edge of the forearm for point r of the lower forearm blob chain for image coordinate k corresponding to i and j,

P is the total number of upper-forearm points,

R is the total number of lower-forearm points and

$k = 0,1$ is the index to image coordinate i or j, respectively.

The above algorithm, essentially computes the midpoints of lines joining every upper forearm point to every lower forearm point to obtain all the midpoints for the forearm.

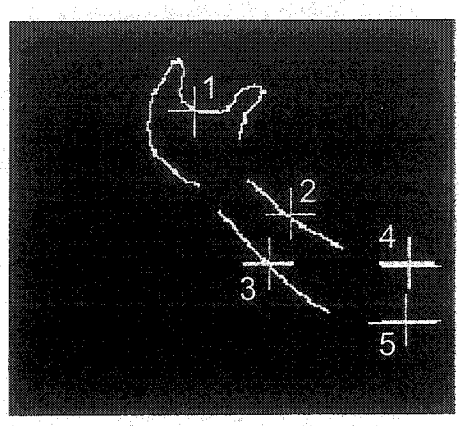


Figure 3.8 Five blob chains representing 1) hand, 2) upper edge of forearm, 3) lower edge of forearm, 4) upper edge of upper-arm and 5) lower edge of upper-arm, respectively.

A similar calculation is used for the upper arm, using all the pixel points corresponding to upper and lower parts of the upper-arm blob chains. This numerous distance computation approach reduces the error due to the irregular shape of the arm in different postures. The midpoints corresponding to the upper-arm and forearm are shown in Figure 3.9.

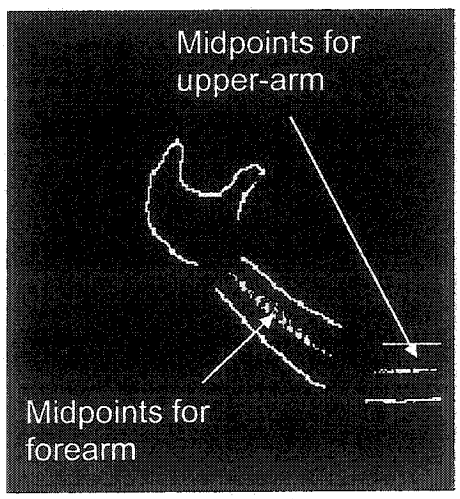


Figure 3.9 Divided arm contour showing midpoints of the forearm and upper-arm.

3.1.7 Computation of best-fit medial lines

A least-squares best-fit medial line is generated for the forearm using the midpoints computed in Section 3.1.6, as shown in Figure 3.10. The least-squares best-fit line is calculated using Algorithm 2, by determining a vector $\varepsilon = [\varepsilon_1 \ \varepsilon_2]^T$, where the ε_1 and ε_2 are the coefficients of the line defined by $y = \varepsilon_1 x + \varepsilon_2$.

Algorithm 2: Least squares best-fit line

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ . & . \\ . & . \\ x_n & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}$$

D S ε

nx1 nx2 2x1

where,

$n = R \times P$ is the number of the midpoints,

D is a n x 1 vector containing the Y coordinates of the midpoints ($y_1, y_2 \ldots y_n$) and

S is a n x 2 vector containing the X coordinates of the midpoints ($x_1, x_2 \ldots x_n$).

Step 1: $D = S\varepsilon$

Step 2: Premultiply both sides by S^T

$$S^T D = S^T S\varepsilon$$

Step 3: Premultiply both sides with $[S^T S]^{-1}$

$$[S^T S]^{-1} S^T D = [S^T S]^{-1}[S^T S]\varepsilon$$

Step 4:
$$\underset{\substack{2\text{x}n \ \ n\text{x}2 \\ 2\text{x}2}}{[S^T \ \ S]^{-1}} \ \underset{2\text{x}n \ \ n\text{x}1}{S^T D} = \underset{2\text{x}1}{\varepsilon} \qquad (3.4)$$

A similar procedure is used to compute the medial line for the upper arm.

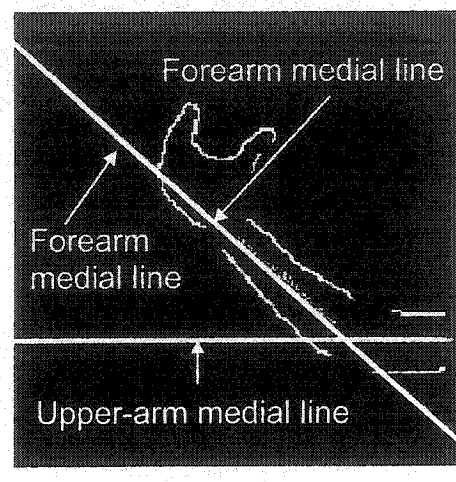


Figure 3.10 Divided arm contour showing medial lines for forearm and upper-arm.

3.1.8 Computation of elbow-joint position

The 2D image coordinates of the elbow-joint centre are estimated from the intersection of the forearm medial line and the upper-arm medial line in each camera view as shown for Camera Views 1 and 2 in Figures 3.11a and 3.11b, respectively. The elbow joint centre position is calculated as explained in the following pseudo code:

Let G_{ex} and G_{ey} be the X and Y coordinates of the elbow joint, respectively, Q_{1c} and Q_{1m} represent the intercept and slope of the forearm medial line, respectively, and Q_{2c} and Q_{2m} represent the intercept and slope of the upper-arm medial line, respectively.

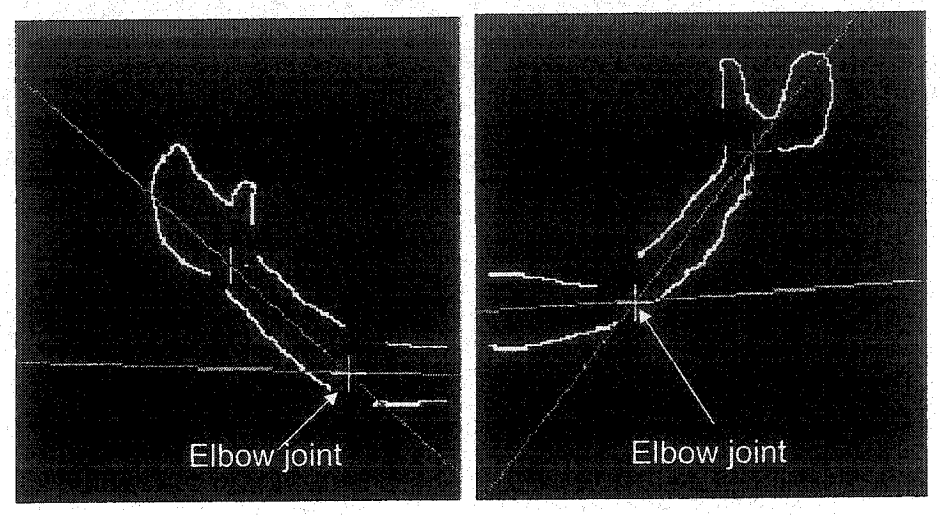
Step 1: Equation of the forearm medial line is given as $G_{ey} = Q_{1m}G_{ex} + Q_{1c}$

Step 2: Equation of the upper-arm medial line is given as $G_{ey} = Q_{2m}G_{ex} + Q_{2c}$

Step 3: The X and Y coordinates of the elbow joint is calculated at the intersection of the forearm and upper-arm medial lines as follows:

$$G_{ex} = \frac{(Q_{2c} - Q_{1c})}{(Q_{1m} - Q_{2m})} \tag{3.5}$$

$$G_{ey} = Q_{1m}G_{ex} + Q_{1c} \tag{3.6}$$



(a) (b)

Figure 3.11 Elbow joint positions computed in (a) Camera View 1 and (b) Camera View 2.

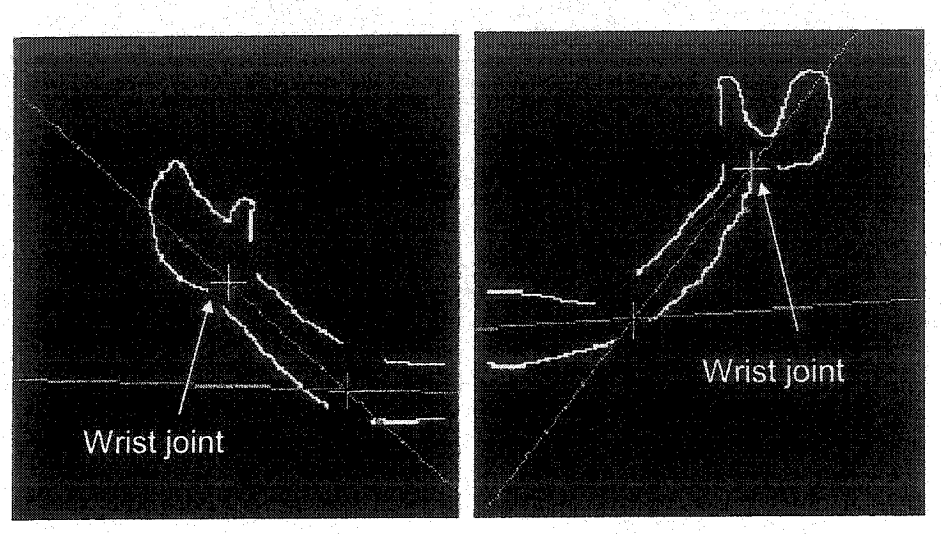
3.1.9 Computation of wrist-joint position

The intersection of the forearm medial line with the dividing line at the wrist, described in Section 3.1.4 (Figure 3.7a), is used to determine the 2D wrist-joint position, as shown for Camera Views 1 and 2 in Figures 3.12a and 3.12b, respectively. The wrist position computed in both images is later used to calculate the 3D wrist-joint position. The 2D wrist-joint centre position is calculated similarly as in Section 3.1.8 and is described below:

$$G_{wx} = \frac{(Q_{3c} - Q_{1c})}{(Q_{1m} - Q_{3m})} \tag{3.7}$$

$$G_{wy} = Q_{1m} G_{wx} + Q_{1c} \tag{3.8}$$

where, G_{wx} and G_{wy} are the X and Y coordinates of the wrist joint, respectively and Q_{3c} and Q_{3m} represent the intercept and slope of the dividing line, respectively.

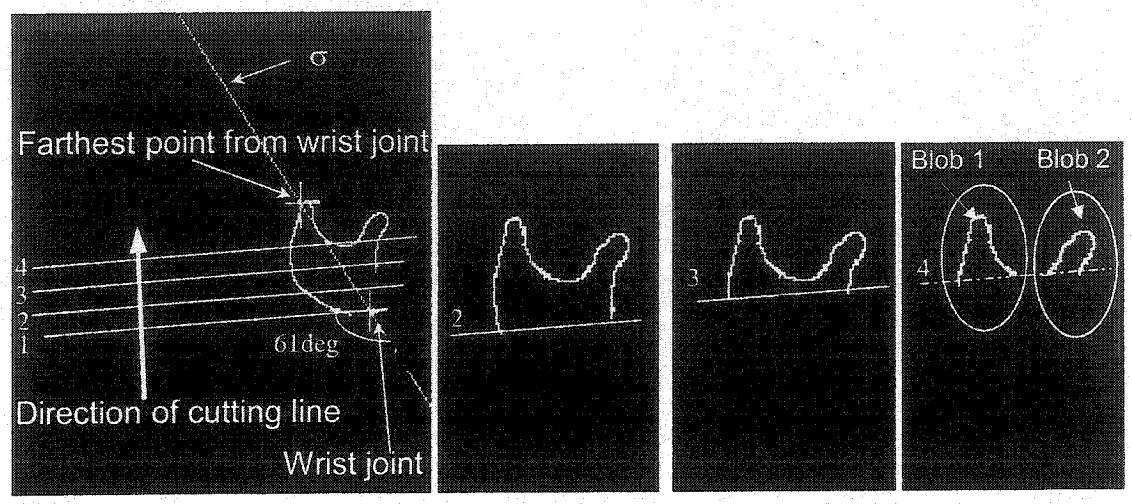


(a) (b)

Figure 3.12 Wrist joint positions computed in (a) Camera View 1 and (b) Camera View 2.

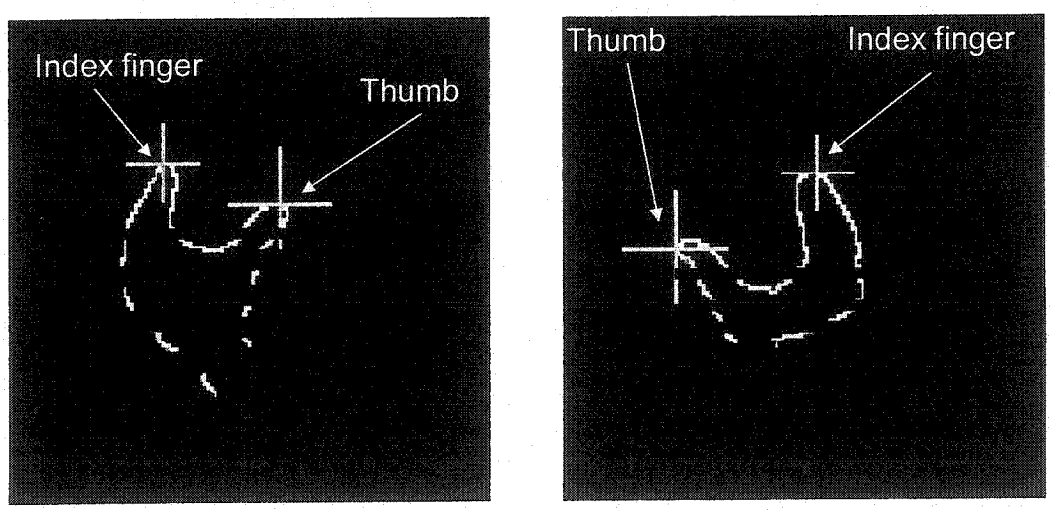
3.1.10 Determination of thumb and index-finger positions

The thumb tip and index-finger tip positions are determined from an image of the hand processed separately from the arm. The image contains only one blob, which represents the contour of the hand, as shown in Figure 3.13a. A line σ through the wrist and farthest point on the hand, and cutting lines numbered 1 to 4 at 61 deg to it, are used to determine the location of the thumb and index finger tips in the image. A cutting line is first drawn through the wrist joint (see line 1 in Figure 3.13a) and then incrementally moved by one-pixel perpendicular to itself toward the fingers, as shown by lines 2, 3 and 4 in Figure 3.13a. The area of the image below the cutting line is set to black, as shown for lines 2, 3, and 4 in Figure 3.13b, 3.13c and 3.13d, respectively. The cutting line is moved until the hand blob is broken into two blobs (Figure 3.13d), which represent the thumb and index finger, respectively. The point farthest from the wrist joint on the two blob chains is determined as the position of the index-finger tip and the point farthest from the wrist on the other blob is determined as the thumb tip. The positions of the index finger and thumb tips from Camera Views 1 and 2 are shown in Figures 3.14a and 3.14b, respectively. The 61 deg angle of the cutting line was found to be optimal to break the hand contour into two blobs.



(a) (b) (c) (d)

Figure 3.13 (a) Determination of the thumb and index-finger tips using cutting lines. The contour of the hand below the cutting line is removed after each incremental move, shown for (b) cutting line 2, (c) cutting line 3 and (d) until two blobs are found by cutting line 4.



(a) (b)

Figure 3.14 Contour of the hand showing the position of the thumb and index-finger tips in (a) Camera View 1 and (b) Camera View 2. The dashed-line contour of the hand was removed in the process, but is shown to illustrate the positions of the index finger and thumb tips relative to the hand.

The process of initialisation, described above is performed to find the initial 2D positions of the elbow joint, wrist joint, index-finger tip, thumb tip and the 3D forearm length. The complete initialisation process is performed by processing 10 pairs of images, where each pair of images includes views from camera 1 and camera 2 and each image of the pair is processed separately. From the wrist and elbow 2D positions in these images, an average value of the 3D forearm length is computed and later used for correction of the wrist-joint position (Section 3.2.10). The 3D forearm length in each frame is determined using the 3D positions of the wrist joint and elbow joint, which are computed from the 2D image positions, using the 3D reconstruction technique explained in detail in Section 3.2.9.2. The 2D positions of the elbow joint, wrist joint, index-finger tip and thumb tip from the last (10^{th}) pair of images is used in the first frame of real-time tracking, which follows immediately after initialisation.

3.2 Real-time tracking

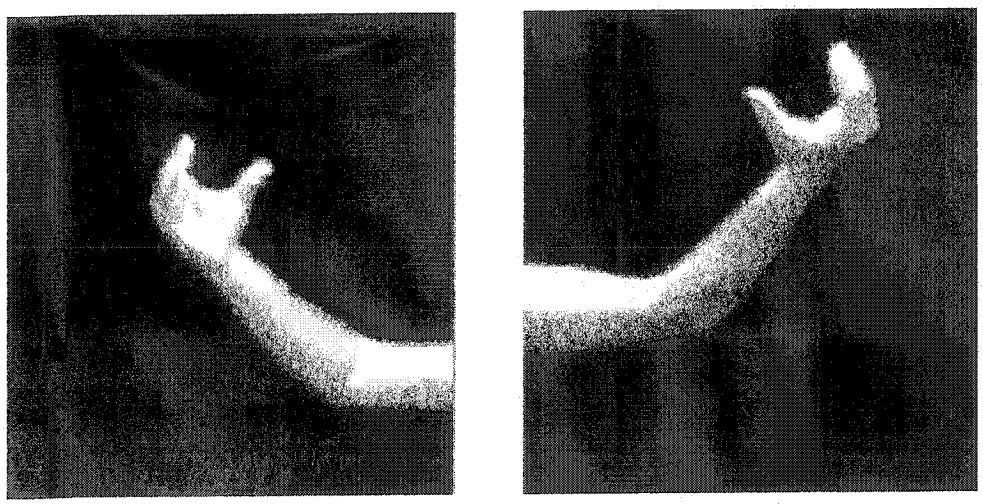
Once the system is initialised, real-time tracking of a sequence of interest can be carried out. The 2D locations of the elbow joint, wrist joint, tips of the thumb and index finger and the 3D forearm length obtained during the initialisation are used to estimate the new positions of the elbow joint, wrist joint, and the tips of the thumb and index finger in each frame. The images are grabbed and processed at approximately 5 frames per second, to maintain real-time processing during hand-arm motion. Steps used in real-time arm tracking are summarised below and explained in detail in the sections that follows. The description that follows is for a single frame of one of the views, from a pair of images from the two cameras. All frames are processed in the same manner.

The first step in real-time tracking is to capture a pair of stereo-camera images of the arm. The image is then segmented by separating the arm from the background. Afterwards, the contour of the arm is extracted from the segmented image using edge detection and then the contour of the arm is divided into three parts, namely the hand, forearm and upper arm. Image coordinates of the contour of each arm part are then determined. For the forearm and upper arm, the midpoints are calculated using the upper edge of forearm and lower edge of forearm, and upper edge of upper-arm and lower edge

of upper-arm, respectively. A best-fit medial line for the forearm and upper-arm is then calculated using the respective midpoints. The elbow joint is then calculated as the intersection of the forearm and upper-arm best-fit medial lines. The wrist joint is then calculated, as a point at a distance equal to the 2D forearm length from the 2D elbow joint position along the forearm medial line. The 3D coordinates of the elbow and wrist joint centres are calculated using the 2D image coordinates of the elbow and wrist joint from the two camera images. The 3D position of the wrist joint is then corrected using the 3D length of the forearm calculated during the initialisation. Finally, the 3D positions of the thumb and index finger tips are computed by processing the image of the hand. The complete process of real-time tracking is described in detail below.

3.2.1 Capture of arm images

During the real-time tracking, the subject is able to move their hand in any direction inside the calibrated control volume from the previous position used in the initialisation. Monochrome images of the unclothed arm against a black background are captured from two cameras. The shape of the hand and the relative orientation between the index finger and the thumb is maintained as explained in Section 3.1.1. The images of the arm from both camera views are shown in Figures 3.15a and 3.15b, respectively.



(a) (b)

Figure 3.15 Image of unclothed arm from (a) Camera View 1 and (b) Camera View 2.

3.2.2 Arm segmentation

The arm is segmented from the background by processing the image based on a pre-calculated threshold value (T_1) of image intensity using Equation 3.1. All image pixels that have an intensity value lower than or equal to T_1 are set to black and all the pixels that have an intensity higher than T_1 are set to white. The result of this thresholding process is a binarized image with the entire arm white and the region of the image apart from the arm, set to black and considered as background. The segmented image of the arm obtained by thresholding is shown in Figure 3.16.

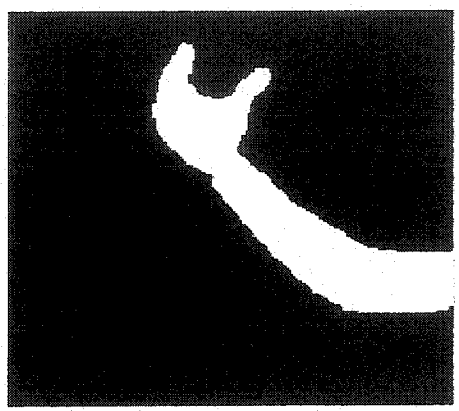


Figure 3.16 Segmented image of arm.

3.2.3 Determination of the arm contour

Edge detection is used to obtain the outer contour of the arm in the images. The edge detection is done using Equation 3.2, as described in Section 3.1.3, resulting in only the arm contour remaining in the image, as shown in Figure 3.17.

3.2.4 Division of arm into parts

A separator consisting of black circular blobs is used to break the contour of the arm into three parts. A separator at the wrist and elbow-joint positions is used to break the contour of the arm at those positions, respectively. The elbow and wrist separator positions for the current frame are calculated using the coordinates of the wrist and elbow-joint positions from the previous frame. The first frame of real-time tracking uses 2D arm joint locations from the last (10^{th}) pair of images in the initialisation. It is assumed that the new position

of the wrist and elbow joint will be near their respective positions in the previous frame because of the sufficiently high frame rate of image acquisition.

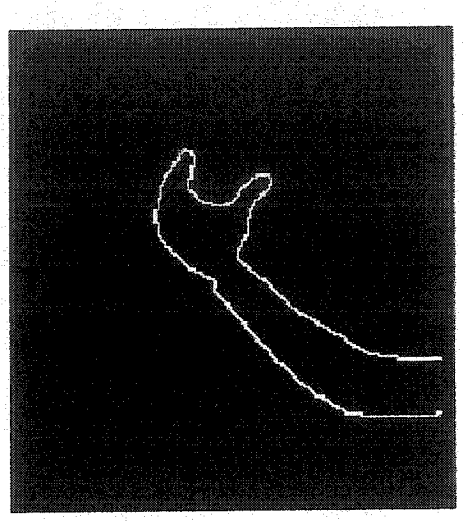


Figure 3.17 Contour of the arm obtained by edge detection.

The elbow separator consists of four circular blobs (Figure 3.18a) that are placed at the upper and lower part of elbow and used to divide the upper-arm from the rest of the arm. The elbow separator is black but is shown white for illustrative purpose in Figure 3.18a. Only three circular blobs of the elbow separator are clearly visible in the image, as blobs 3 and 4 are overlapping. The 2D position of a circular blob of the elbow separator $(\Im_{ex}, \Im_{ey})$ is calculated using a fixed proportion χ_1 of the upper-arm thickness t_u from the 2D position of the elbow joint (G_{ex}, G_{ey}) in the previous frame, along a line perpendicular to the upper-arm medial line, as determined in the following equation.

$$\Im_{ex} = G_{ex} + t_u \chi_1 \cos(\theta) \tag{3.9}$$

$$\Im_{ey} = G_{ey} + t_u \chi_1 \sin(\theta) \tag{3.10}$$

where:

θ is the angle between the line perpendicular to the upper-arm medial line and the *i*-axis,

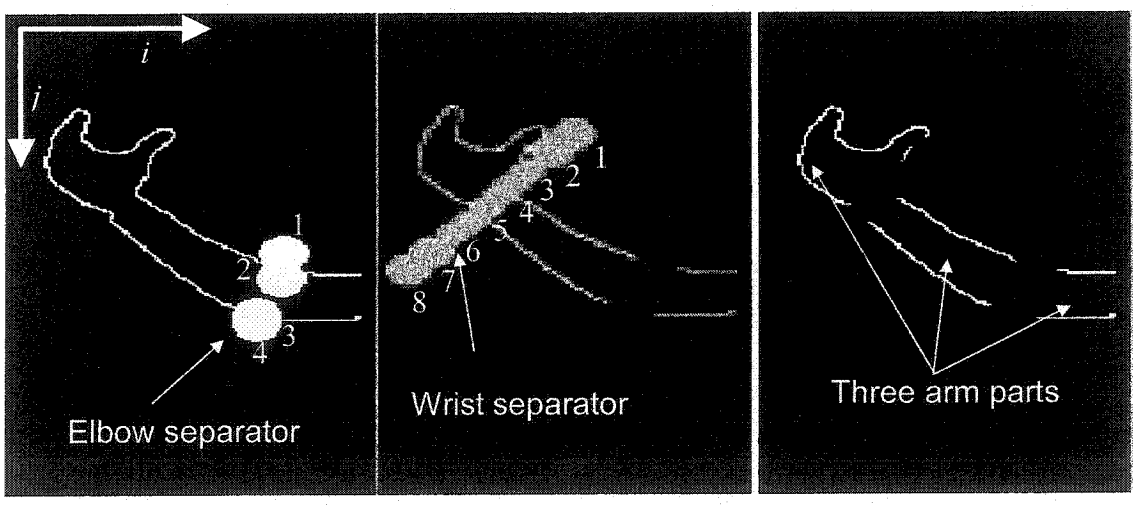
χ_1 is the fixed value equal to 0.8, 0.3, -0.4 and -0.6 for each of the circular blobs in the separator, respectively.

The diameter of the elbow separator γ_e is calculated as a fixed proportion χ_2 of the upper-arm thickness.

$$\gamma_e = t_u \chi_2 \tag{3.11}$$

where, γ_e is the diameter of the separator.

The wrist separator is used to separate the hand from the forearm. The wrist separator consists of eight circular blobs and their positions and diameters are calculated in a similar manner to those for the elbow, using the forearm thickness, forearm medial line and wrist-joint position instead of the upper-arm thickness, upper-arm medial line, and elbow-joint position, respectively. The fixed proportion χ_1 will have values 1.9, 1.2, 0.7, 0.2, -0.2, -0.7, -1.2 and -1.9 for the circular blobs of the wrist separator, respectively. The wrist separator is black but is shown white for illustrative purpose in Figure 3.18b. The image of the divided arm parts is shown in Figure 3.18c.



(a) (b) (c)

Figure 3.18 Separators used to the divide the arm: (a) elbow separator used to divide arm into upper-arm and forearm, (b) wrist separator used to divide arm into hand and forearm, and (c) arm divided into three parts: hand, forearm and upper-arm.

3.2.5 Determination of image coordinates of arm parts

When the separator partitions the arm into three parts, the contour of the arm is broken into five different blobs representing 1) hand, 2) upper edge of forearm, 3) lower edge of forearm, 4) upper edge of upper-arm and 5) lower edge of upper-arm, respectively. The five blobs are shown in Figure 3.19, where a cross is used to indicate the centre of gravity of each blob. Each blob in the current frame is labelled as a blob corresponding to one of the above five arm blobs; the hand, upper edge of forearm, lower edge of forearm, upper edge of upper-arm and lower edge of upper-arm, respectively. The labelling process utilizes a proximity comparison function, that uses the position of the centre of gravity of the blobs from the previous frame to identify and label the blob corresponding to the same body part in the current frame, as explained below in Algorithm 3.

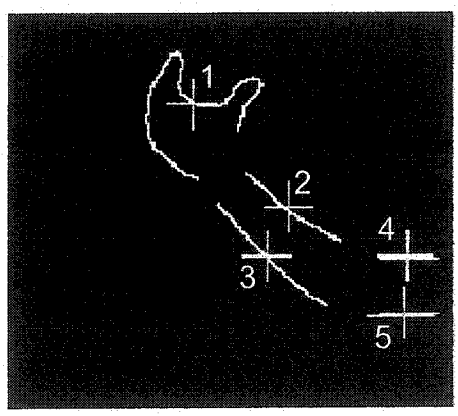


Figure 3.19 Image of the arm contour broken into 5-labelled blobs with the centres of gravity indicated by crosses.

Each of the five identified blobs is given a label and their position of centre of gravity is saved in each camera frame to be used in the next frame. Proximity comparison is done between each blob in the current frame to each blob in the previous frame. For the first frame in the real-time processing the proximity comparison is done with the last frame of the initialisation. Among all the blobs in the previous frame, the one blob which is nearest to a blob in the current frame is chosen, based on the minimum distance between their centres of gravity. The label number of the identified blob in the previous frame is assigned to the blob in the current frame. A blob in the previous frame that is associated once is not used again in the proximity comparison. The remaining four blobs

are labelled similarly using the proximity comparison function. Successful blob labelling by the proximity comparison function is based on the assumption that there is little displacement in the spatial position of the human arm between two consecutive images, as the images are grabbed at a sufficiently high frame rate.

Algorithm 3: Proximity comparison function

The concept of identifying blobs in the new images using the proximity comparison function is described below.

Step 1: The centre of gravity $C(i, j)$ and label number δ_* for each blob m for the current frame n are computed using a built-in Matrox software function (MIL 7.0), and assigned to an array $(E_{n,m}[\delta_*, C(i, j)])$. The label δ_* has to be modified from that assigned automatically by the built-in function which uses no previous-frame information.

Step 2: For each blob m in the current frame n, a corresponding blob m' in the previous frame, n-1, which yields the least distance between the centres of gravity of the two blobs, is identified.

For all $m = 1, 2.....5$:

$$v_m = \min_{m'=1,2..5} \left\| \left(E_{n,m}[\delta_*, C(i, j)]\right) - \left(E_{n-1,m'}[\delta, C(i, j)]\right) \right\| \qquad (3.12)$$

Assign δ to δ_*, for the blob m' which yields minimum v_m.

where, $\| \ \|$ is the Euclidean distance and δ is the label value of the blobs in frame n-1.

3.2.6 Computation of midpoints for forearm and upper arm

The midpoints of the forearm are calculated from the two blob chains corresponding to the upper and lower edges of the forearm, respectively, using Equation 3.3 in the numerous distance computation approach, described in Section 3.1.6. The midpoints of the upper arm are computed in a similar manner from the two blob chains corresponding

to the upper and lower edges of the upper arm, respectively. The midpoints corresponding to the forearm and the upper-arm are shown in Figure 3.20.

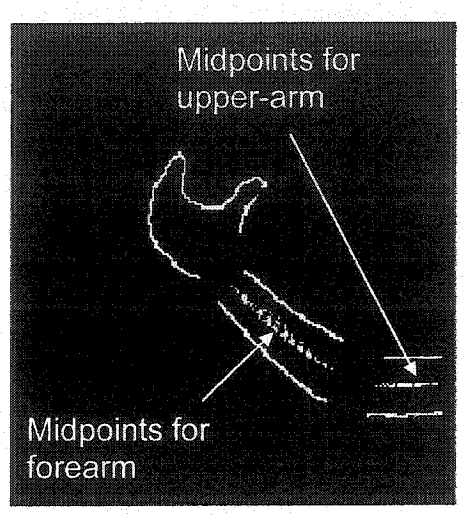


Figure 3.20 Image of the divided arm contour showing midpoints of the forearm and upper-arm.

3.2.7 Computation of medial lines for forearm and upper arm

The best-fit medial lines are determined from the upper-arm and forearm midpoints, respectively, using the least-squared sum approach described in Algorithm 2 in Section 3.1.7. The best-fit medial lines for the forearm and the upper-arm are shown in Figure 3.21.

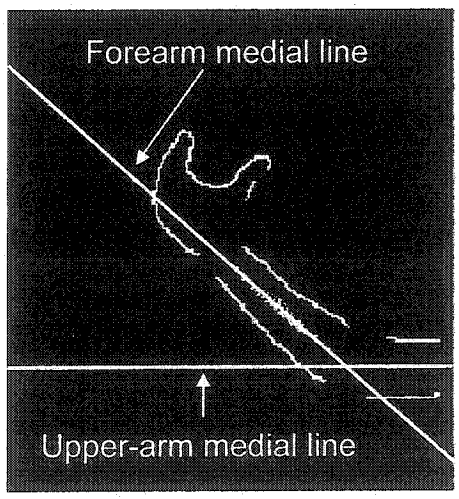


Figure 3.21 Arm contour with medial lines for forearm and upper-arm.

3.2.8 Computation of 2D joint positions

3.2.8.1 Computation of 2D elbow-joint position

The elbow-joint position is determined by the intersection of the forearm and upper-arm medial lines, using Equations 3.5 and 3.6, as explained in Section 3.1.8. The position of the elbow joint is shown in Figures 3.22a and 3.22b for Camera Views 1 and 2, respectively.

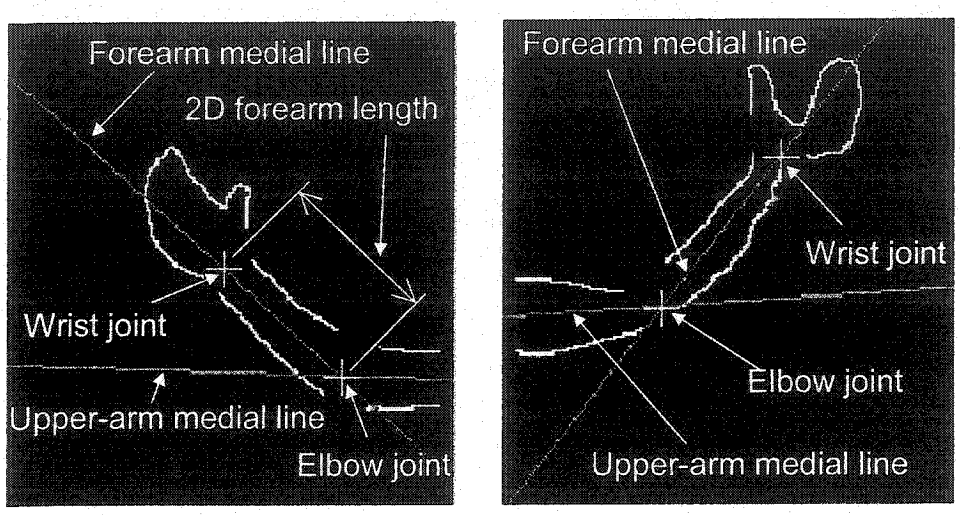
3.2.8.2 Computation of 2D wrist joint position

The wrist joint position in the current frame is calculated as the 2D point along the forearm medial line at a distance equal to the 2D forearm length (Figure 3.22a), computed in the previous frame, from the elbow joint in the current image. The calculated wrist joint positions are shown in Figures 3.22a and 3.22b for Camera Views 1 and 2, respectively. The 2D wrist joint position from both camera views are used to compute the 3D wrist joint position using a 3D reconstruction technique, explained later in Section 3.2.9.2. As the 2D forearm length changes, with the change in orientation of the arm, a correction in the 3D wrist position is therefore performed. This is explained later in Section 3.2.10. The 2D wrist position in the current frame is also updated using the corrected 3D wrist position, as it will be needed in the next frame to compute the new 2D wrist joint position.

3.2.9 Determination of 3D coordinates from 2D image coordinates

The 3D coordinates of all hand and arm points are determined from the 2D image coordinates of these points using the stereo-camera image pair available in each time frame, and employing a 3D reconstruction technique. The 3D reconstruction technique can only be used after stereo-camera calibration is performed. The camera calibration is a technique that relates the 2D information from camera images to 3D object space. There are several techniques of camera calibration, such as the Tsai method [38] and Direct Linear Transformation (DLT) [39]. In the current research, the DLT camera calibration technique is used to calibrate a pair of cameras. The DLT method has the advantage of not requiring calculation of intrinsic and extrinsic camera calibration parameters such as focal length, pixel resolution in the image plane, and position and orientation of each

camera. The DLT methods of stereo-camera calibration and 3D reconstruction are explained below.



(a) (b)

Figure 3.22 Image of the arm showing the elbow-joint centre position at the intersection of the medial lines and the calculated wrist joint centre from (a) Camera View 1 and (b) Camera View 2.

3.2.9.1 DLT-based stereo-camera calibration

Stereo-camera calibration is a technique used to relate 3D coordinates of a point in object space (x, y, z) to the image coordinates of the same point in Image Plane 1 (u_1, v_1) and in Image Plane 2 (u_2, v_2), as shown in Figure 3.23. The determination of the unknown coordinates in 3D space, x, y, z, from image coordinates of the same point from two cameras (u_1, v_1) and (u_2, v_2) is known as *3D reconstruction*. It is typically carried out, by first calibrating the cameras of the stereo-camera system. The calibration is done by capturing camera images of a set of well-defined points in space called *control points*, whose coordinates are known. The camera setup (position and angles) must remain the same once calibrated, as it will be used later in the measurement of the coordinates of points of interest, whose 3D coordinates are not known *a priori* and are thereafter termed *unknown points*. The calibration requires that the 3D positions of the control points be measured accurately and independently from the camera system. These points are therefore also referred to as *known points*. This measurement must be done using an

object-space coordinate reference frame. The control points are typically markers fixed on a calibration frame and the calibrated volume occupied by the control points should ideally be the same as the space required for points whose motion is of interest. In this research, the calibrated space is where motion of the human arm will take place.

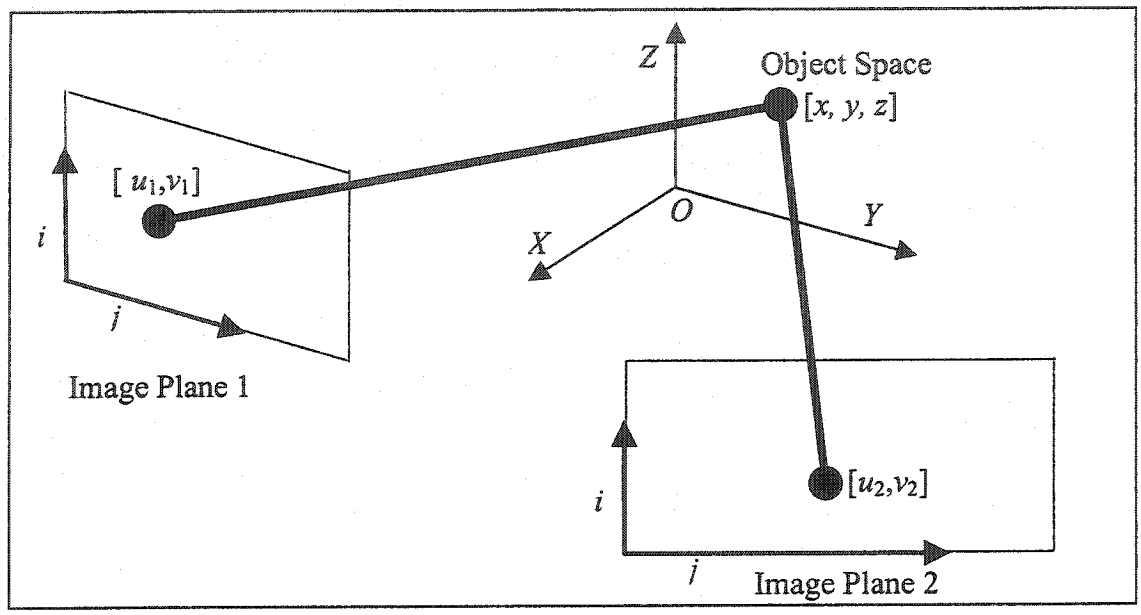


Figure 3.23 Camera image plane mapping to object space.

For the main application of this research in robot teleoperation, two cameras are installed at fixed positions to view a human operator performing arm motion. The calibration volume has dimensions of 1000 x 1000 x 994 mm^3. A calibration plate is positioned at three different vertical locations so that the whole calibration volume is covered, as shown in Figure 3.24. The cameras capture 3 pairs of images of the calibration plate as shown in Figure 3.25. An already developed software [40] is used to process each raw image shown in Figure 3.25, to compute the image coordinate of the centroid of each marker in the image, as shown for Camera View 1 in Figure 3.26a and Camera View 2 in Figure 3.26b, for position 3. After processing each image in Figure 3.25, the image coordinates of the 75 pair of points on the calibration device are stored (25 points at three positions).

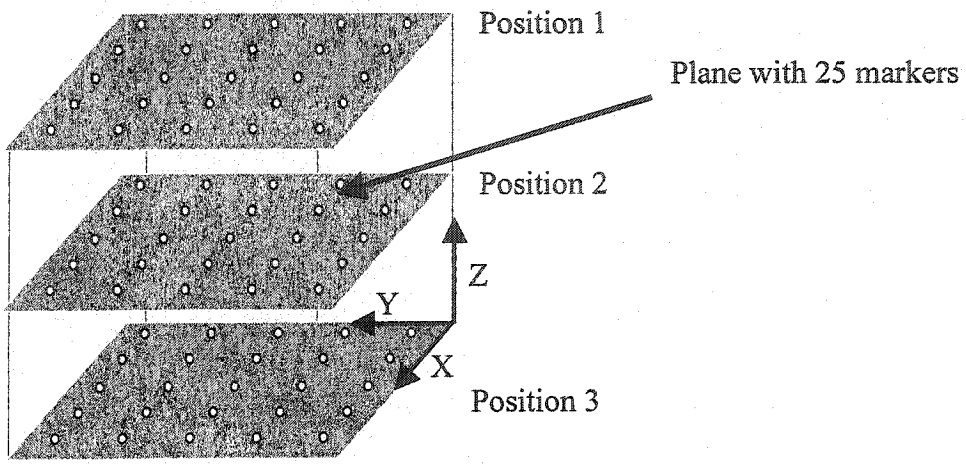


Figure 3.24 Schematic representation of the calibration volume showing three different vertical positions of the calibration plate used for camera calibration. The plate has 25 equally spaced markers and is placed in 3 positions to obtain 75 marker positions.

The 2D image coordinates (u_1, v_1) and (u_2, v_2) of a single object point can be related to the 3D object space coordinates (x, y, z) of the point using the following DLT camera-calibration equations:

$$u_1 = \frac{(L_1 x + L_2 y + L_3 z + L_4)}{(L_9 x + L_{10} y + L_{11} z + 1)} \tag{3.13}$$

$$v_1 = \frac{(L_5 x + L_6 y + L_7 z + L_8)}{(L_9 x + L_{10} y + L_{11} z + 1)} \tag{3.14}$$

$$u_2 = \frac{(L_{12} x + L_{13} y + L_{14} z + L_{15})}{(L_{20} x + L_{21} y + L_{22} z + 1)} \tag{3.15}$$

$$v_2 = \frac{(L_{16} x + L_{17} y + L_{18} z + L_{19})}{(L_{20} x + L_{21} y + L_{22} z + 1)} \tag{3.16}$$

where,

L_1 to L_{22} are the DLT parameters, which implicitly describe camera positions and orientation, camera internal parameters and linear lens distortion factors.

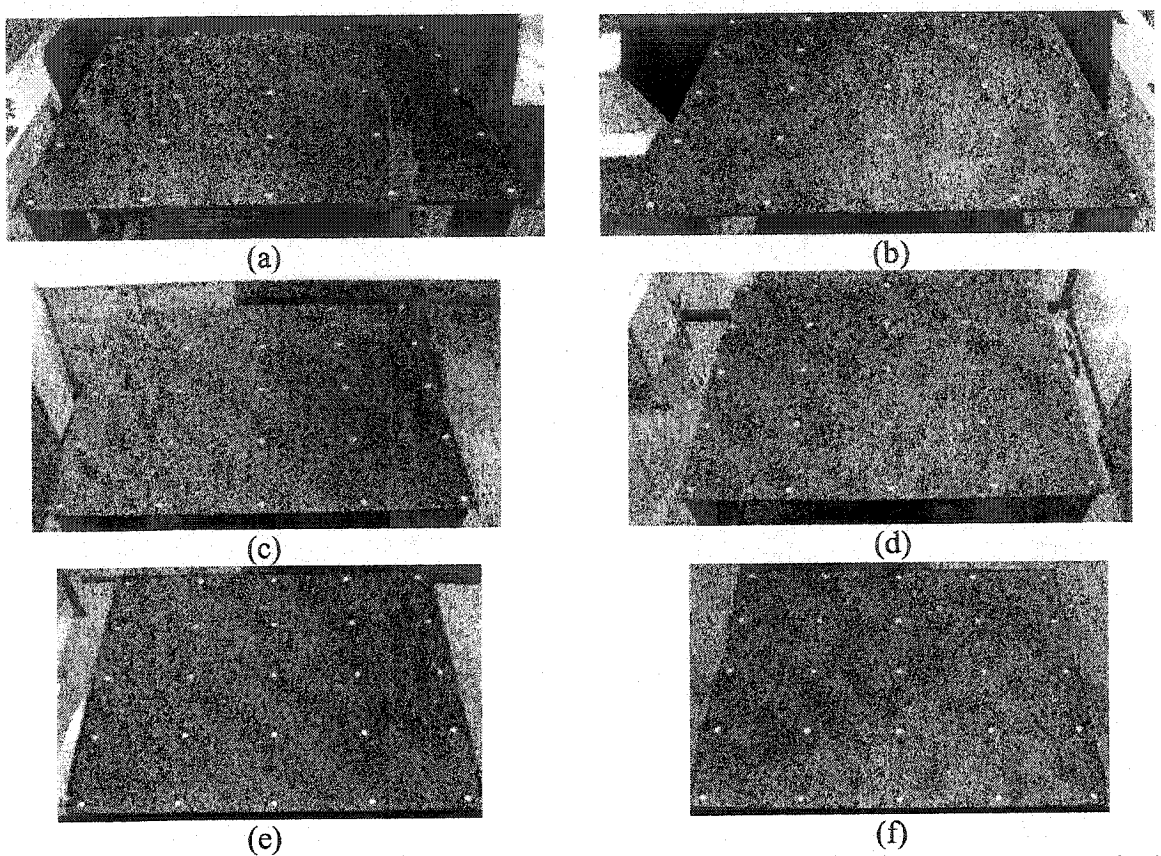


Figure 3.25 Images of the actual calibration device showing three different vertical positions of the calibration plate from two camera views, (a) Position 1, View 1, (b) Position 1, View 2, (c) Position 2, View 1, (d) Position 2, View 2, and (e) Position 3, View 1. The raw image in (f) shows a simulated view, as the actual image was not available, however the actual processed image is shown in Figure 3.26b.

To perform calibration, the 11 unknown parameters for each camera, L_1 to L_{11} for Camera 1 and L_{12} to L_{22} for Camera 2 are calculated using Equations 3.13 and 3.14 and Equations 3.15 and 3.16, respectively. There are 22 unknowns and 4 equations for each pair of image points corresponding to a single object point. To solve for the 22 unknown parameters, at least 6 object points are required to generate the minimum 22 equations required. As there could be errors in the measured coordinates of the 3D control points, as well as in the image coordinates of the control points, a total of 25 object points of the available 75 points are used. A least-squared-sum approach is used to solve for the 22 calibration parameters using 25 pairs of image points (100 equations). The

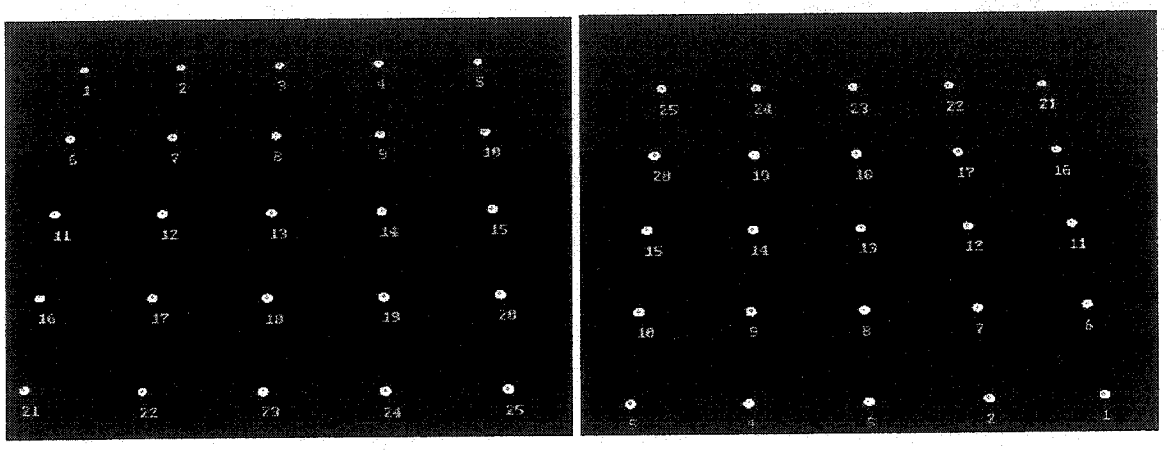
Equations 3.13-3.16 are rearranged as shown below, to solve for the 22 calibration parameters by matrix methods as developed by [40].

$$xL_1 + yL_2 + zL_3 + L_4 - u_1xL_9 - u_1yL_{10} - u_1zL_{11} = u_1 \quad (3.17)$$

$$xL_5 + yL_6 + zL_7 + L_8 - v_1xL_9 - v_1yL_{10} - v_1zL_{11} = v_1 \quad (3.18)$$

$$xL_{12} + yL_{13} + zL_{14} + L_{15} - u_2xL_{20} - u_2yL_{21} - u_2zL_{22} = u_2 \quad (3.19)$$

$$xL_{16} + yL_{17} + zL_{18} + L_{19} - v_2xL_{20} - v_2yL_{21} - v_2zL_{22} = v_2 \quad (3.20)$$



(a) (b)

Figure 3.26 Processed images of the calibration device in position 3 showing the computed centroids of the 25 markers as white dots in (a) Camera View 1 and (b) Camera View 2.

3.2.9.2 3D Reconstruction

The camera calibration is performed to determine the values of the calibration parameters, L_1, L_2.....L_{22}, which relate 3D and the 2D coordinates of a point. Once the calibration is completed, the calibration device can be removed from the object space to allow the motion of the interest, hand-arm motion, to take place. With the 22 calibration parameters determined, image coordinates of the same point from two cameras (u_1, v_1) and (u_2, v_2), respectively, are used in Equations 3.17-3.20, to determine the 3D coordinates x, y and z of the point. Equations 3.17-3.20 are rearranged to solve for 3D coordinates (x, y, z) using the following equations developed by [40]:

$$(L_1 - L_9u_1)x + (L_2 - L_{10}u_1)y + (L_3 - L_{11}u_1)z = u_1 - L_4 \quad (3.21)$$

$$(L_5 - L_9v_1)x + (L_6 - L_{10}v_1)y + (L_7 - L_{11}v_1)z = v_1 - L_8 \quad (3.22)$$

$$(L_{12} - L_{20}u_2)x + (L_{13} - L_{21}u_2)y + (L_{14} - L_{22}u_2)z = u_2 - L_{15} \quad (3.23)$$

$$(L_{16} - L_{20}v_2)x + (L_{17} - L_{21}v_2)y + (L_{18} - L_{22}v_2)z = v_2 - L_{19} \quad (3.24)$$

As there are three unknown values x, y and z and four equations (Equations 3.21-3.24), the three unknowns are solved with four equations, using information from all four cameras by the least-squared-sum method and the following matrix form.

$$\begin{bmatrix} L_1 - L_9u_1 & L_2 - L_{10}u_1 & L_3 - L_{11}u_1 \\ L_5 - L_9v_1 & L_6 - L_{10}v_1 & L_7 - L_{11}v_1 \\ L_{12} - L_{20}u_2 & L_{13} - L_{21}u_2 & L_{14} - L_{22}u_2 \\ L_{16} - L_{20}v_2 & L_{17} - L_{21}v_2 & L_{18} - L_{22}v_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u_1 - L_4 \\ v_1 - L_8 \\ u_2 - L_{15} \\ v_2 - L_{19} \end{bmatrix} \quad (3.25)$$

Using the 3D reconstruction technique above, the 3D coordinates of the wrist and elbow joints are calculated from the 2D image coordinates of the wrist and elbow joints, as calculated in Section 3.2.8. The 3D reconstruction of points on the hand is done in a similar manner from the respective 2D coordinates, the computations of which are discussed later.

3.2.10 Correction of wrist-joint position

The 3D wrist-joint position calculated in Section 3.2.9, is based on the 2D image length of the forearm (Figure 3.22a), from the previous frame. However, the 2D image length does not remain the same from frame to frame, if the angle of the arm changes with respect to the camera. A correction is therefore required for the 3D wrist-joint position. During the initialisation sequence, the 3D forearm length was calculated and saved for this correction of the 3D wrist joint position in each frame. A corrected wrist-joint position is computed by finding a point in each camera image, which (a) yields the least 3D error in forearm length calculated in the current frame based on an assumed true 3D

length calculated in the initialisation, and (b) results in the least error in 3D slope of the forearm-medial line based on the assumed true slope using the uncorrected 2D wrist position. As the wrist joint is located on the forearm medial line in the two 2D images, only its distance from the elbow joint along this line has to be corrected. The method of finding the corrected wrist joint is explained in Algorithm 4.

Algorithm 4: Wrist correction algorithm

Step 1: The true 3D forearm length $\hbar$ is calculated during initialisation, as an average of ten pair of images.

$$\hbar = \frac{1}{10}\sum_{g=1}^{10}\left\| G_{eg}(x,y,z) - G_{wg}(x,y,z) \right\| \tag{3.26}$$

where,

$G_{eg}(x,y,z)$ is the 3D elbow joint coordinate and

$G_{wg}(x,y,z)$ is the 3D wrist joint coordinate.

Step 2: Create two sets of 11 points, $\rho_a(i,j)$ and $\rho_b(i,j)$, in the two camera images, respectively, where a = 1, 2,.....11 and b = 1, 2,.....11, to include five points spaced one pixel apart on either side of the uncorrected wrist joint along the forearm-medial line, as shown in Figure 3.27.

Step 3: Compute the 3D coordinate $\hat{\rho}_k(x,y,z)$ for all 121 combinations of pairs of points $\rho_a(i,j)$ and $\rho_b(i,j)$.

$k = 0$

For a =1 to 11:

For b = 1 to 11:

Increment k

$$\hat{\rho}_k(x,y,z) = \Delta\left[\rho_a(i,j), \rho_b(i,j)\right] \tag{3.27}$$

where, k = 1, 2,.....121, is the index for the point pair combination and

Δ is a 3D reconstruction function based on the technique described in Section 3.2.9.2.

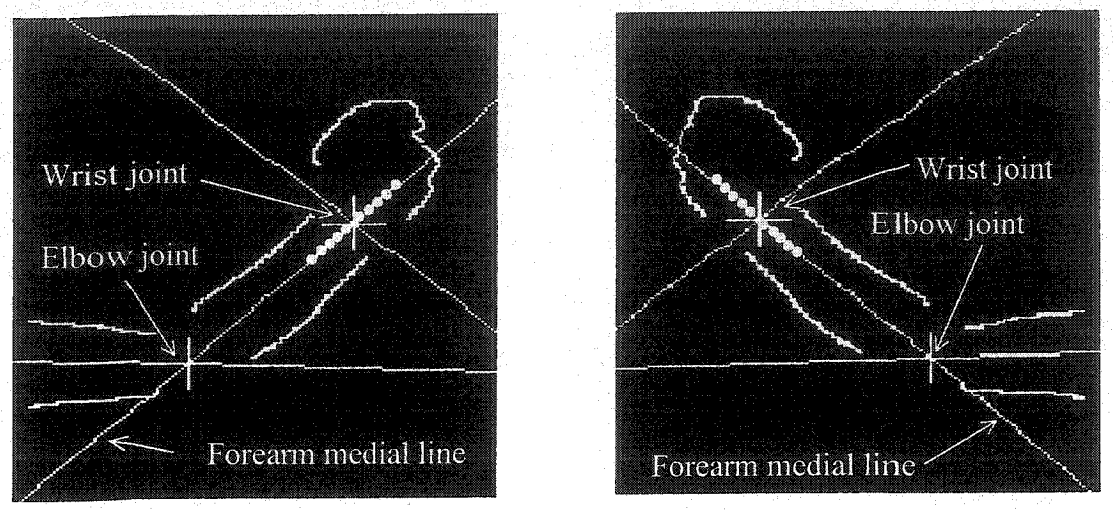


Figure 3.27 Image of the arm showing the use of 11 points lying on the forearm medial line for the wrist-joint correction, for (a) Camera View 1 and (b) Camera View 2. The point size and spacing are enlarged for illustrative purposes.

Step 4: Determine the error Φ in distance between the computed 3D forearm length and the true 3D forearm length $\hbar$.

For $k = 1$ to 121:

$$\Phi_k = \left\| \hbar(x, y, z) - \left[\left\| G_e(x, y, z) - \hat{\rho}_k(x, y, z) \right\| \right] \right\| \qquad (3.28)$$

where, $G_e(x, y, z)$ is the 3D elbow joint coordinate for the current frame.

Step 5: Determine the ten $\bar{\rho}_k(x, y, z)$ from the set of 121 $\hat{\rho}_k(x, y, z)$, which yield the minimum Φ_k. Essentially these $\bar{\rho}_k(x, y, z)$ are the 10 potential wrist points, computed from the 10 best combinations of $\rho_a(i, j)$ and $\rho_b(i, j)$, which yield a minimum error in 3D forearm length.

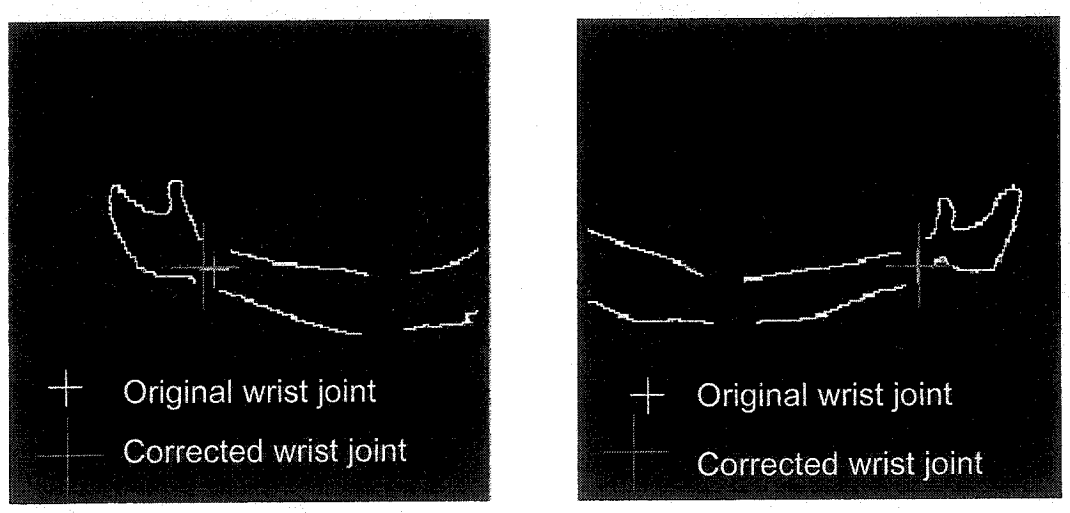
Step 6: The true value of the 3D slope $\hat{\lambda}$ of the forearm line is calculated using the 3D slope-calculation operator ψ, based on the 3D position of the elbow joint $G_e(x, y, z)$ and the uncorrected wrist joint $\tilde{G}_w(x, y, z)$.

$$\hat{\lambda} = \psi\left(G_e(x, y, z), \tilde{G}_w(x, y, z)\right) \qquad (3.29)$$

Step 7: The corrected wrist joint $G_w(x, y, z)$ is calculated as the point having minimum error in slope between the computed 3D slope and true 3D forearm slope $\hat{\lambda}$, for the 10 $\bar{\rho}_k(x, y, z)$.

$$G_w(x, y, z) = \min_{k=1,2...10} \left[\hat{\lambda} - \psi(G_e(x, y, z), \bar{\rho}_k(x, y, z))\right] \tag{3.30}$$

The pair of image coordinates corresponding to the corrected 3D wrist-joint position are used to update the 2D-wrist joint coordinates, as shown in Figure 3.28a and b, as these image coordinates will be required in the next frame.



(a) (b)

Figure 3.28 Image of the arm showing the corrected 2D wrist joint position in (a) Camera View 1 and (b) Camera View 2.

3.2.11 Computation of thumb tip and index-finger tip positions

In the main application of this research, the human-robot interface, the position and orientation information of the hand is required to control the motion of the robot in 3D space. The position data for the robot are provided by the 3D coordinates of the wrist-joint centre. To calculate the hand orientation, 3D coordinates of three points on the human hand are required. As the wrist-joint position can be one of the three points required for the hand orientation, the 3D position of only two other points are required.

The tips of the index finger and thumb are used as the two other points in order to simulate the shape of the robot gripper. The computation of the orientation of the hand based on the 3D positions of the index-finger tip, thumb tip and wrist-joint centre is described later in Section 4.2. To determine the 3D position of the index finger and thumb, the image of the hand is processed for each camera view. The images contain only one blob chain, which represents the contour of the hand. With movement of the hand, three possibilities may occur, as shown by a flowchart in Figure 3.29. The thumb may be: (Case 1) visible in both images, (Case 2) hidden in both images and (Case 3) visible in only one image. The pair of images is processed for all of the possibilities until the position of the thumb and index-finger tips are found, as described in detail below.

3.2.11.1 Case1: Thumb visible in both images

The image is first processed for the possibility of finding the 2D positions of the thumb and index-finger tips in both images, as described in Section 3.1.10 and shown in Figure 3.13. If the 2D image coordinates of the thumb and index finger are located in both images they are used to calculate the 3D coordinates of the thumb and index finger using the 3D reconstruction technique described in Section 3.2.9.2. The 2D positions of the thumb and index-finger tips are shown in Figures 3.30a-b. To verify, which of the two blobs in the current frame corresponds to the index finger and thumb tips position, the 2D distance between the farthest points on both the blobs is compared to the 2D position of the index-finger tip in the previous frame. The blob which yields the least distance from the 2D position of the index-finger tip in the previous frame is reassigned as the index-finger tip in the current frame. The 2D position of the index-finger tip is only used for this verification, as the thumb point can become hidden in many images.

3.2.11.2 Case 2: Thumb hidden in both images

If after processing the images as described in Section 3.2.11.1, the position of the thumb is found to be not visible in both images, the images are processed to determine the position of the index finger and small finger tips. A line σ from the wrist to the farthest point on the hand is first defined, as described in Section 3.1.10 and shown in Figure 3.31a.

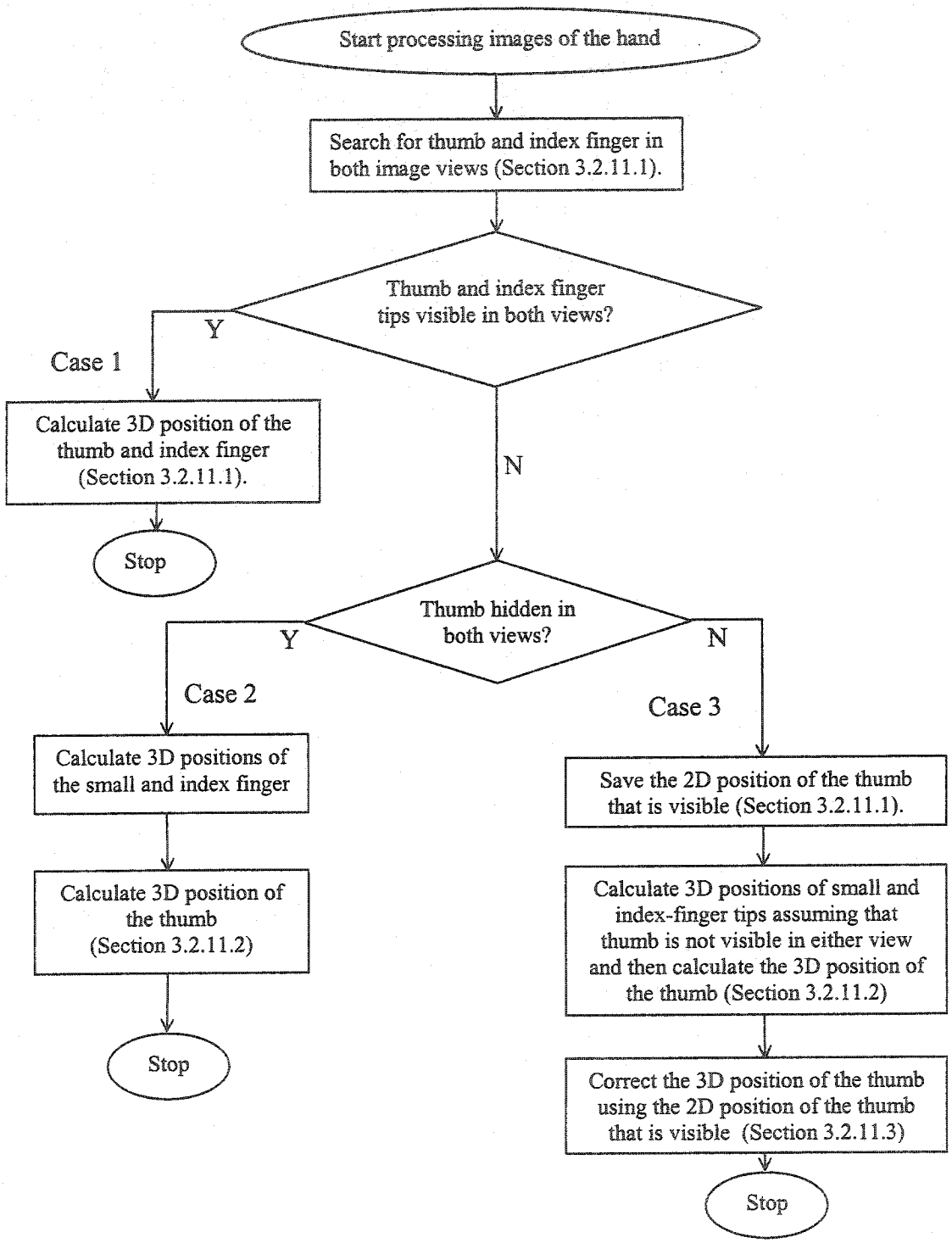
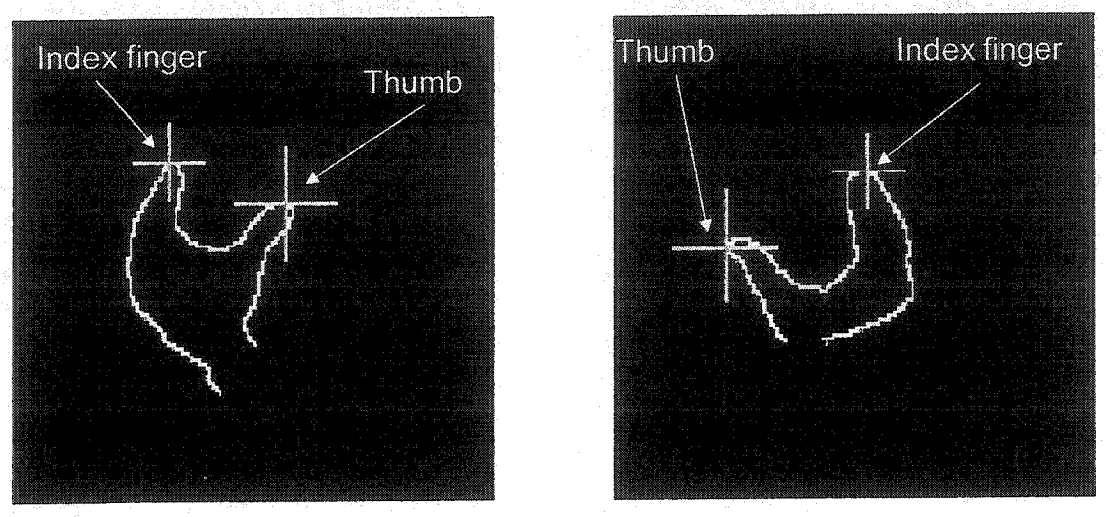


Figure 3.29 Flowchart showing the steps used to determine the 3D position of the thumb and index-finger tips.



(a) (b)

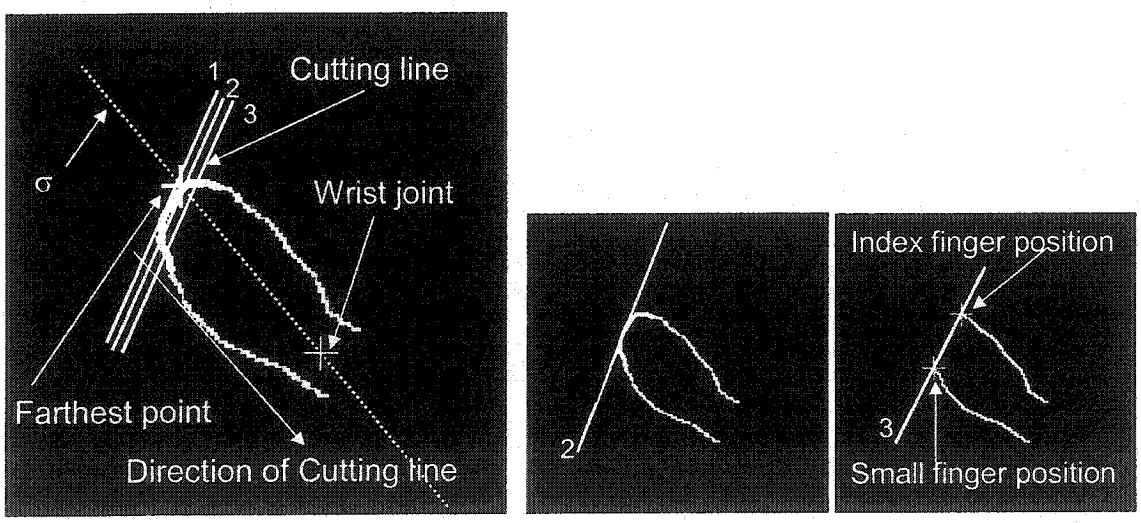
Figure 3.30 Position of thumb and index finger of the hand from (a) Camera View 1 and (b) Camera View 2.

A second line (cutting line) at 61 deg to it shown in Figure 3.31a, is used to determine the location of the index finger and small-finger tips in the image. The cutting line is first drawn through the farthest point on the contour of the hand from the wrist joint (line 1 in Figure 3.31a) and is then incrementally moved by one-pixel perpendicular to itself towards the wrist joint, as shown by lines 2 and 3 in Figure 3.31a, until two blobs are identified for the index finger and small finger, respectively. The area of the image above the cutting line is set to black, as shown for lines 2 and 3 in Figures 3.31b and 3.31c, respectively. The cutting line is moved until the hand blob is broken into two blobs, as shown in Figure 3.31c. The point farthest from the wrist joint on the two blobs is determined as the index-finger tip and the point farthest from the wrist on the other blob is determined as the small-finger tip. To verify, which of the two blobs in the current frame corresponds to the index-finger position, the 2D distance between the farthest points on both the blobs is compared to the 2D position of the index-finger tip in the previous frame. The blob that yields the least distance from the 2D position of the index-finger tip in the previous frame is reassigned as the index-finger tip in the current frame. A similar process is performed on the image from the other camera to compute the image coordinates of the index and small fingers.

The 2D image coordinates of the small finger and index finger from both cameras are used to calculate the 3D coordinates of the small finger and index finger using the 3D reconstruction technique explained in Section 3.2.9.2. From the 3D coordinate positions of the wrist joint, index-finger tip and small-finger tip, the 3D coordinate position of the thumb (x_0, y_0, z_0) is calculated by solving Equation 3.31, which is a general equation of a plane through a given point.

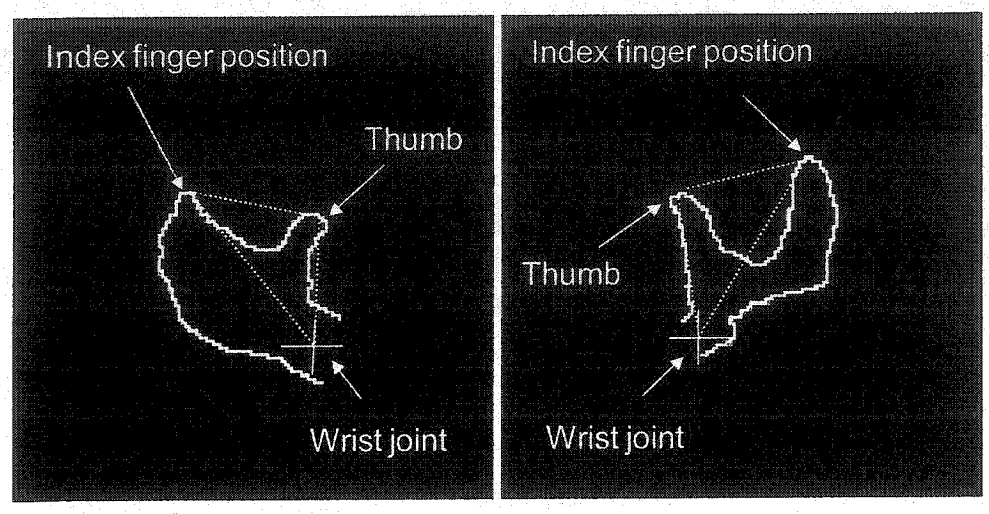
$$A_1 x_0 + A_2 y_0 + A_3 z_0 + A_4 = 0 \tag{3.31}$$

Equation 3.31 represents a plane that passes through the 3D positions of the wrist, index-finger tip and thumb tip. The 3D plane is shown in 2D for Camera View 1 in Figure 3.32a and for Camera View 2 in Figure 3.32b. In the above equation, there are seven unknowns, the four coefficients of a plane A_1, A_2, A_3, A_4 and three values for the coordinates of the thumb (x_0, y_0, z_0). Therefore 7 equations, Equations 3.32-3.38 are solved simultaneously to compute these seven unknowns.



(a) (b) (c)

Figure 3.31 (a) Determination of the index and small finger tips using cutting lines. The contour of the hand above the cutting line is erased after each incremental move, shown for (b) cutting line 2, and (c) cutting line 3 till two blobs are found. The spacing of the cutting line is enlarged for illustrative purposes.



(a) (b)

Figure 3.32 Plane joining thumb tip, index-finger tip and wrist joint from (a) Camera View 1, and (b) Camera View 2.

For the plane passing through the three points, the unknown thumb point (x_0, y_0, z_0), known index-finger point (x_1, y_1, z_1) and known wrist joint (x_2, y_2, z_2), the coefficients can be expressed in the following equations:

$$A_1 = y_1 (z_2 - z_0) + y_2 (z_0 - z_1) + y_0 (z_1 - z_2) \tag{3.32}$$

$$A_2 = z_1 (x_2 - x_0) + z_2 (x_0 - x_1) + z_0 (x_1 - x_2) \tag{3.33}$$

$$A_3 = x_1 (y_2 - y_0) + x_2 (y_0 - y_1) + x_0 (y_1 - y_2) \tag{3.34}$$

$$A_4 = -1 [x_1 (y_2 z_0 - y_0 z_2) + x_2 (y_0 z_1 - y_1 z_0) + x_0 (y_1 z_2 - y_2 z_1)] \tag{3.35}$$

The equation relating the 3D distance ℓ_1 between the unknown thumb point and the known index-finger tip position is given by

$$\ell_1 = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \tag{3.36}$$

The equation relating the 3D distance ℓ_2 between the unknown thumb point and the known small-finger tip position is given by

$$\ell_2 = \sqrt{(x_0 - x_3)^2 + (y_0 - y_3)^2 + (z_0 - z_3)^2} \tag{3.37}$$

where, x_3, y_3 and z_3 are the 3D coordinates for the small finger.

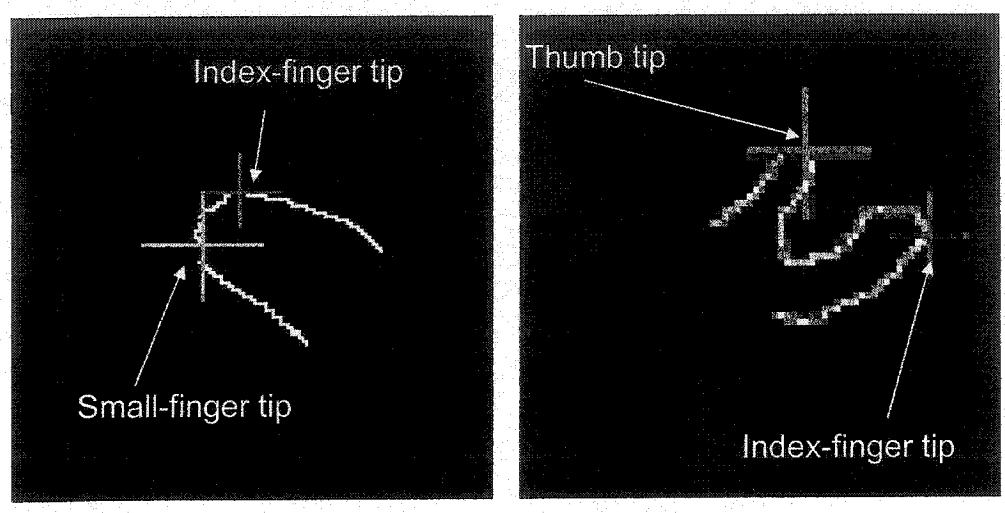
The equation relating the 3D distance ℓ_3 between the unknown thumb point and the known wrist position is given by

$$\ell_3 = \sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2 + (z_0 - z_2)^2} \tag{3.38}$$

The value of 3D lengths ℓ_1 and ℓ_3 are estimated using the 3D coordinates of the known thumb, and index finger and wrist positions from the previous frame. ℓ_2 is estimated to be equal to ℓ_1, based on the assumption that the distance from the index-finger to thumb and small finger to thumb are equal. The 3D thumb position is thus computed from the above equations.

3.2.11.3 Case 3: Thumb visible in only one image

If the thumb is visible in only one image (Figure 3.33), the images are first processed as in Section 3.2.11.2 to provide the 3D position of the index-finger tip and an initial estimate of the 3D position of the thumb tip. The computed position of the thumb tip is later corrected, using the 2D position of the thumb that is visible in one of the views, as described below. This 2D position of the thumb tip, shown in Figure 3.33b, is calculated in Section 3.2.11.1. The initial estimate of the 3D position of the thumb tip is then used in Equations 3.13-3.16 to determine the 2D image coordinates of the thumb tip from both camera views. The 2D coordinate of the thumb tip, which was visible in one of the camera views, is then used with the 2D image coordinate of the thumb tip from the other camera, in the 3D reconstruction (Section 3.2.9.2) to calculate the corrected 3D coordinate of the thumb tip. The corrected 3D thumb-tip position is used to update the 3D lengths, ℓ_1, ℓ_2 and ℓ_3, as these could be used in processing the next image, if necessary.



(a) (b)

Figure 3.33 Stereo-image pair of the hand showing (a) 2D positions of small finger and index finger in Camera View 1 and (b) 2D positions of the thumb and index finger in Camera View 2.

3.2.12 Summary of real-time tracking

The markerless human hand-arm tracking is performed in real-time by repeating the 11 steps described in Sections 3.2.1 to 3.2.11 for each stereo-pair of images by computing the 3D coordinates of the wrist joint, elbow joint, index-finger tip and thumb tip positions. The 2D positions of the elbow joint, wrist joint, index-finger tip and thumb tip calculated for the current frame are used for processing the next pair of images. For the application of robot teleoperation, the 3D positions of the wrist joint, index-finger tip and thumb tip for each frame are sent in real-time to a remote robot-site computer as described in more detail in Chapter 4.

3.3 Software and hardware used

The software used for developing markerless tracking was written in a Visual C++ compiler (version 6.0), using built-in functions of Matrox Imaging Library (version 7.0). The two cameras used were WAT-902H (Watec America Corporation) 1/2" CCD with 570 TV-line resolution. The frame-grabber board was a Matrox Meteor_II PCI frame grabber. The computer used was a Pentium 4, 1.6 GHz processor with 256 MB Random Access Memory (RAM).

CHAPTER 4. HUMAN-ROBOT INTERFACE

The method of markerless human-arm motion tracking is developed mainly for application in a human-robot interface between a human operator and a robot-manipulator arm. The human-robot interface is intended, to allow an operator to control the robot by performing natural hand and arm motions for a task, and having the human motion copied by the robot in real-time. This interface would thus allow robot teleoperation whereby a human can perform motion for a task at a local site and a robot would perform the task at a remote site. An already developed interface [2-3] is used, that allows the user to control the robot by marker-based tracking of the human hand motion. In this thesis, this interface is extended to use markerless tracking instead of marker-based tracking to control the motion of the robot. Tracking by a markerless technique provides an advantage that the user can move their hand in a natural manner without the constraint of avoiding marker occlusion. In the robot teleoperation application, the position of the robot manipulator arm is controlled by the 3D position of the wrist joint, and the orientation of the robot gripper is controlled by the positions of the wrist joint, index-finger tip and thumb tip. The position of elbow joint is currently not used for robot teleoperation; however, it would be implemented in further research at the HMIIS Lab to control the motion of each joint in a seven-axis robot. The details of the human-robot interface are explained in detail below.

4.1 Description of the human-robot interface

The human-robot-manipulator interface for robot teleoperation consists of three main systems, namely, the markerless motion-tracking system at the local site, the robot manipulator at a remote site and a visual-display-based feedback system of the robot's motion, as shown in Figure 4.1.

4.1.1 Motion tracking system at the local site

A stereo-camera-based markerless motion tracking system at the local-site is used to track the hand-arm motion of a human operator, to obtain the 3D coordinates of the wrist,

index finger and thumb. This 3D hand-arm position data is sent to the robot computer at the remote site in real-time. In the current setup, all the 3D computations are performed at the local site, as compared to previous setup [2-3], where only 2D data was sent by the local computer thus requiring all 3D computations to be performed at the remote site.

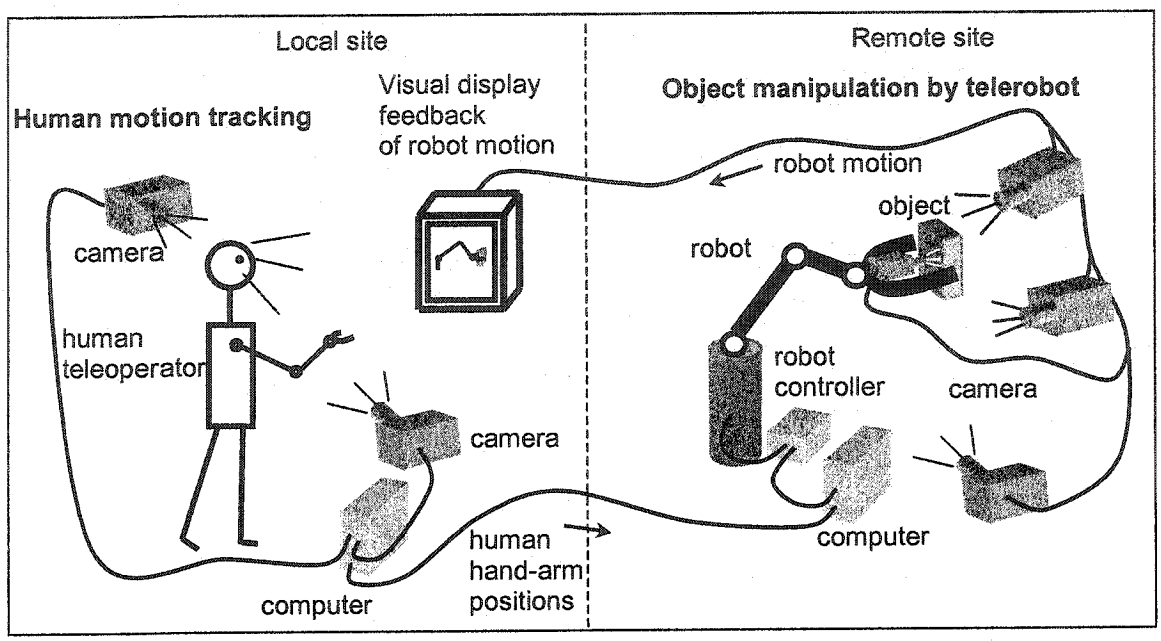


Figure 4.1 Schematic representation of the human-robot-manipulator interface used for robot teleoperation [from [3] Kofman et al. 2003].

4.1.2 Robot manipulator at the remote site

The robot manipulator used is a six-axis Thermo CRS A465 robot (Thermo CRS, Ltd.) with axes shown in Figure 4.2. The robot arm is equipped with a two-finger servo-gripper end-effector. The robot computer receives 3D coordinate information of the human hand-arm motion from the computer at the local site and provides the robot with 3D information of position and orientation in robot's reference frame. The data is sent in real-time, thus enabling the robot to copy the motion performed by the human operator, simultaneously. During robot teleoperation, the human operator performs hand-arm motions that would naturally be used to complete an object manipulation task.

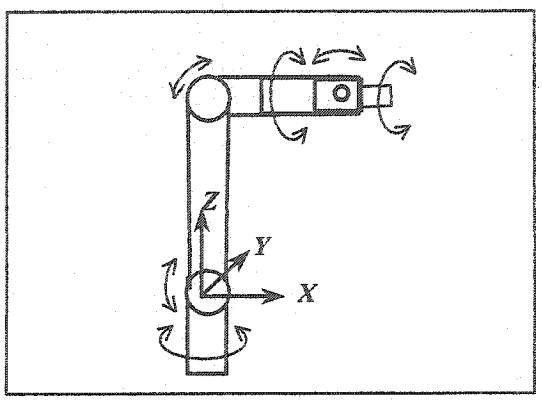


Figure 4.2 Schematic representation of robot manipulator arm showing 6-axes of rotation [adapted from [3] Kofman et al. 2003].

4.1.3 Visual feedback system

A visual feedback system is used to provide the operator at the local-site with information about the robot interaction with objects at the remote site, so that the operator can make necessary adjustments to their actions to complete the task at hand. The system is composed of four cameras at the robot site (Figure 4.3), which obtain continuous images of the robot interaction with the objects in its environment, and a television monitor at the local site where the images are displayed for the human operator (Figure 4.4). Two cameras are mounted in front of the robot manipulator at different heights, one on the side, while Camera 4 is mounted on the robot end-effector. The four views of the cameras simultaneously displayed for the human operator are shown in Figure 4.4. The feedback monitor at the local site is positioned in such a way that the operator can view their own arm at the same time as the views of the robot at the remote site [3].

4.2 Robot teleoperation

At the start of the robot teleoperation, the interface is initialised by storing the first 3D position of each of the thumb tip, index-finger tip and wrist joint, that are sent by the markerless tracking system. These stored positions correspond to the 3D positions of the 30^{th} frame of the real-time tracking described in Chapter 3, and are used as teleoperation reference positions, discussed below. The initial position and orientation of the robot end-effector in the starting position, shown in Figure 4.2, is also stored as the reference position for the robot.

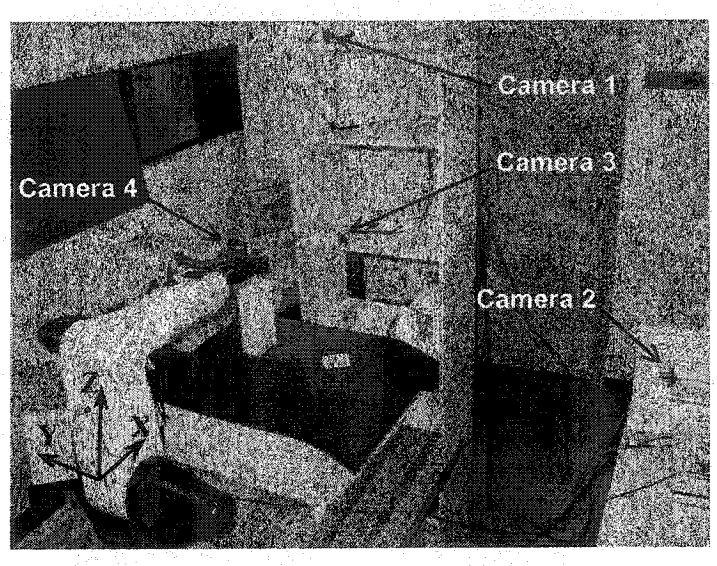


Figure 4.3 Remote-robot-site showing the location of the four cameras used for feedback to the human operator [from [3] Kofman et al. 2003].

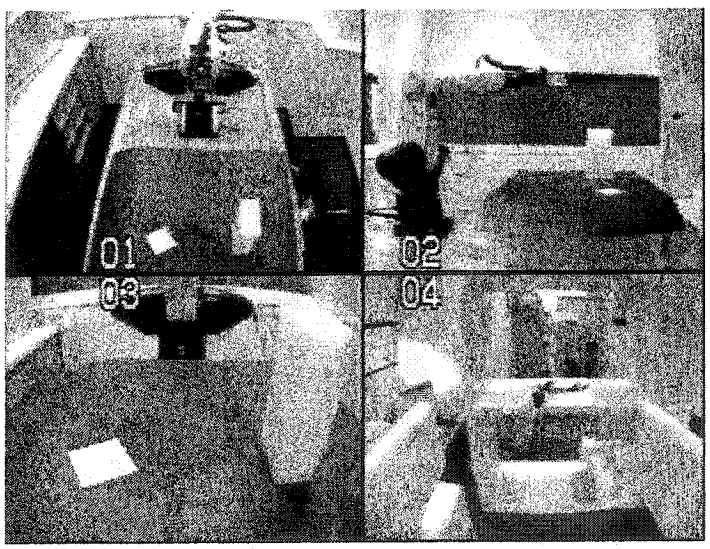


Figure 4.4 Simultaneous display on the monitor of views from the four cameras at the remote-robot-site for visual feedback to the human operator [from Kofman et al. 2003].

The position of the end-effector is controlled by the position of the wrist joint of the human operator and the orientation of the gripper is calculated using the 3D positions of the index-finger tip, thumb tip and wrist joint. The orientation of the hand is calculated from the angles formed by projecting the thumb tip q, index-finger tip h, wrist joint w, and the midpoint of the line joining the thumb and index finger, qh, c' (Figure 4.5a) onto

the 2D planes, XY, XZ and YZ, as shown in Figures 4.5b, 4.5c and 4.5d [3], respectively. The angles α, β, and γ determine the orientations of the hand about the Z, Y and X axes, respectively.

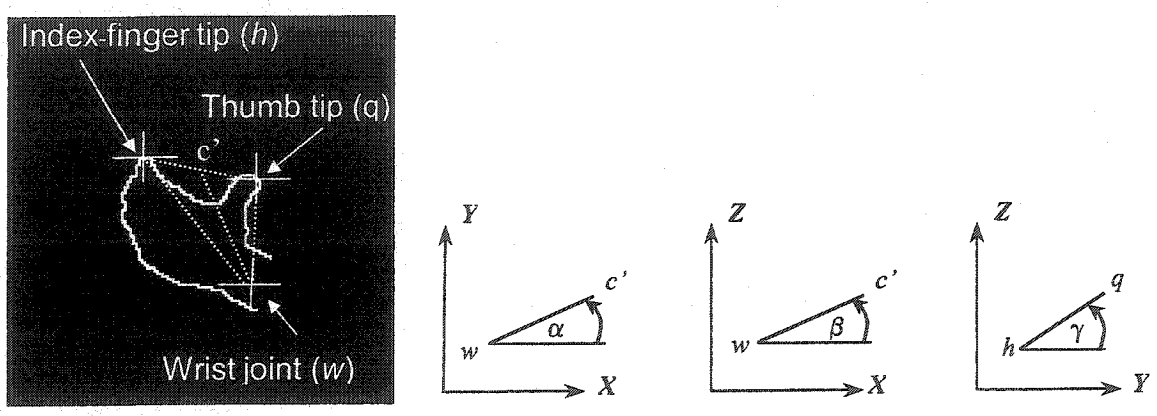
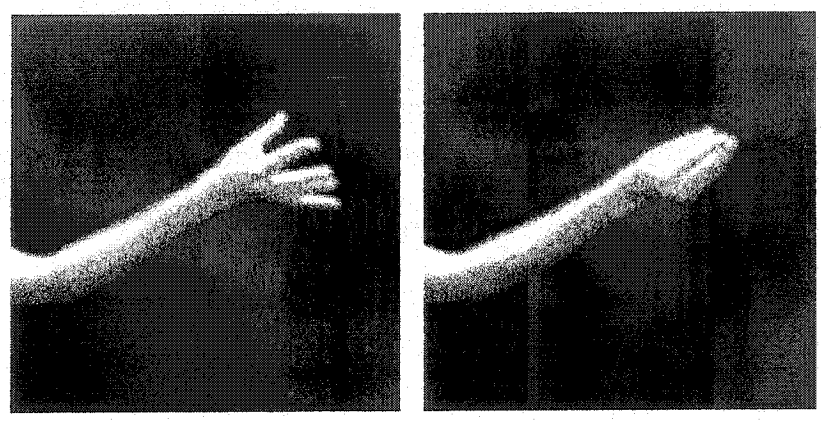


Figure 4.5 Determination of hand-orientation angles from a) the index-finger tip, thumb tip, and wrist positions on the hand; and angles b) α, c) β and d) γ [adapted from [3] Kofman et al. 2003].

During teleoperation, the relative human wrist position from its reference position is applied as a relative robot end-effector motion from its reference position in the robot world-coordinate system, as described in [2-3]. In a similar manner, the relative orientation of the hand from its reference orientation is applied as a relative rotation of the end-effector from its reference orientation in the robot world-coordinate system.

In order to perform a gripping task to pick up objects, the two-finger gripper must be sent a command to open. The opening command is given by the user opening their hand, such that the fingers are individually visible in at least one of the camera views, as shown in Figure 4.6a. When the object is within reach, the gripper can be signalled to close. To perform a closing in natural manner, the user first opens their hand as a windup and then closes their hand to indicate a closing of the gripper, as shown in Figures 4.6a and 4.6b. However, the image-processing algorithm only processes the opening of the hand as a signal to close the robot gripper. Therefore, opening of the hand once commands the robot gripper to open, and opening the hand again when the gripper is

already open closes the gripper. The closing of the gripper is carried out under force control in order to avoid crushing an object and to prevent exceeding the maximum allowable gripping force of the robot [3].



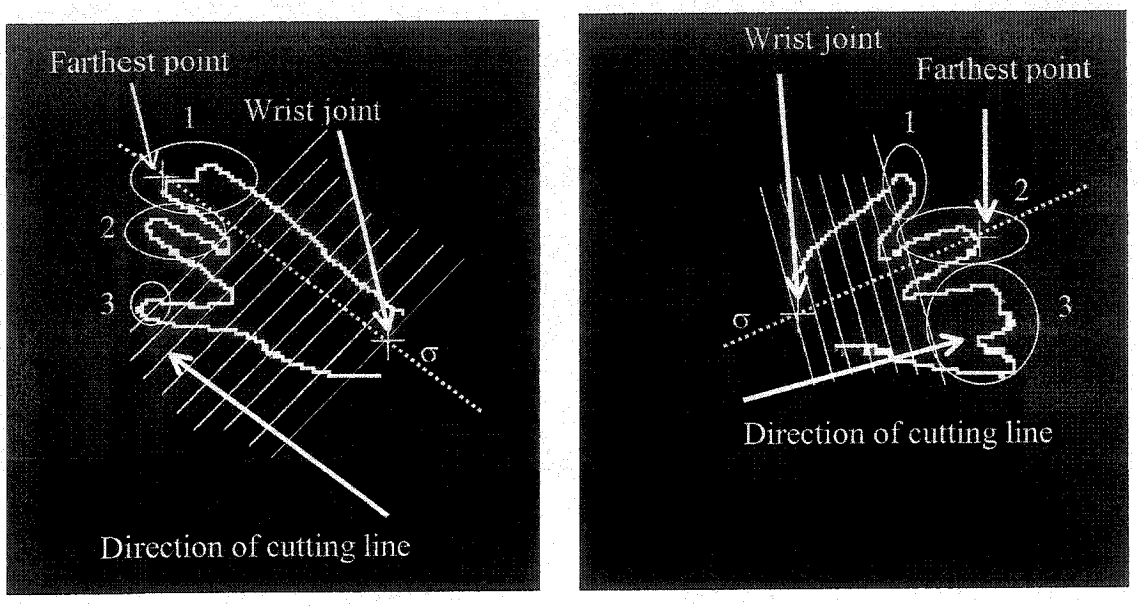
(a) (b)

Figure 4.6 Image of the hand used to control gripper opening/closing, showing (a) fingers open and (b) fingers closed.

To identify opening and closing commands, the hand contour images from both camera views are processed (Figure 4.7). A line σ from the wrist to the farthest point on the hand is first defined, as described in Sections 3.1.10 and 3.2.11.2 and shown in Figures 4.7a and 4.7b. A second line (cutting line) at 61 deg to σ, as shown in Figures 4.7a and 4.7b, is then drawn through the wrist joint and incrementally moved by one-pixel perpendicular to itself toward the fingers, until more than two blobs are identified (similar to thumb and finger identification described in Section 3.1.10). If more than two blobs are found then the hand is considered to be open and the robot is signalled to open or close the gripper depending on the current gripper state.

4.3 Software and hardware used

All commands to the Thermo CRS 6-axis robot are made via the Thermo CRS robot controller from a user-written Visual C++ program, using in-built functions of the CRS Active Robot software library. The computer used at the remote-site is a Pentium 4, 1.8 GHz processor with 384 MB RAM [3].



(a) (b)

Figure 4.7 Hand contour images from (a) Camera View 1 and (b) Camera View 2, showing processing of the image to identify opening and closing commands.

CHAPTER 5. EXPERIMENTS

To evaluate the accuracy of the markerless tracking method, three different experiments were performed. An instantaneous centre of rotation test, explained in Section 5.1, was used to determine the accuracy of the computed elbow and wrist joint centre. A direct comparison between markerless and marker-based tracking techniques, as explained in Section 5.2, was used to determine the accuracy of the computed wrist joint, index-finger tip and thumb tip positions. A teleoperation test, as explained in Section 5.3, was used to evaluate the ability of the markerless-tracking technique to control a robot manipulator using the human-robot interface, to perform object manipulation tasks.

5.1 Instantaneous centre-of-rotation test

5.1.1 Elbow-joint-accuracy test

A test was performed, to determine the accuracy of the elbow joint position computed by the markerless tracking technique, by comparing this position with $E_{ICR}(x, y, z)$, the position of the elbow joint computed by an Instantaneous Centre of Rotation (ICR) technique, which essentially calculates the ICR of the forearm. A marker was first installed on the end of index finger (Figure 5.1) A subject then rotated their forearm in 3D space by an angle of 90 deg from the starting position, while keeping the upper arm stationary. The path of the marker was tracked using marker-based tracking and the 3D coordinates of the index-finger markers in the image sequence were used to find the ICR of the forearm, $E_{ICR}(x, y, z)$. The calculated elbow joint coordinates by the markerless human arm tracking technique, $E_{ML}(x, y, z)$, averaged over ten frames, were then compared to $E_{ICR}(x, y, z)$, computed for the same ten frames. The average of the elbow joint coordinates calculated by the markerless human arm tracking technique was calculated using the following equation:

$$\overline{E}_{ML}(x, y, z) = \frac{1}{10}\sum_{n=1}^{10} E_{ML}(x_n, y_n, z_n) \tag{5.1}$$

where, $\overline{E}_{ML}(x,y,z)$ is the mean elbow joint centre coordinate for markerless tracking, and n is the frame index for the 10 frames used in the computation.

$E_{ICR}(x,y,z)$ was calculated as the centre of the best-fit sphere, obtained using the 3D coordinates of the index-finger marker positions over the same ten frames as in the computation of $\overline{E}_{ML}(x,y,z)$ just described. $E_{ICR}(x,y,z)$ was calculated using the following equation:

$$E_{ICR}(x,y,z) = \Theta[M(x_1,y_1,z_1), M(x_2,y_2,z_2).....M(x_{10},y_{10},z_{10})] \quad (5.2)$$

where: $M(x_n,y_n,z_n) = (M_x, M_y, M_z)$ is the position of the index-finger marker for frame n, and Θ is the best-fit-sphere operator that computes the centre of the sphere, based on a brute force optimization that iteratively modifies the coordinates of the centre of the sphere x_0, y_0, z_0 and the radius of the sphere r_S until the sum of the squared error e is minimized.

$$e = \sum_{n=1}^{10}\left[(M_{xn}-x_0)^2 + (M_{yn}-y_0)^2 + (M_{zn}-z_0)^2 - r_S^2\right] \quad (5.3)$$

An illustration of the elbow joint accuracy test is found in Figure 5.1. The position of the index-finger marker in the adjacent frames is shown with enlarged spacing for clarity. The figure shows 11 positions of markers. From these 11 positions, 2 sets of 10 consecutive positions can be obtained as shown. The test described above was carried out 65 times using 65 sets of 10 consecutive positions from a total set of 75 marker positions.

The error in elbow joint position computed by markerless tracking, denoted as $E_e(x,y,z)$ was determined by the absolute difference between $E_{ICR}(x,y,z)$ and $\overline{E}_{ML}(x,y,z)$ using the following equation:

$$E_e(x,y,z) = \left| E_{ICR}(x,y,z) - \overline{E}_{ML}(x,y,z) \right| \quad (5.4)$$

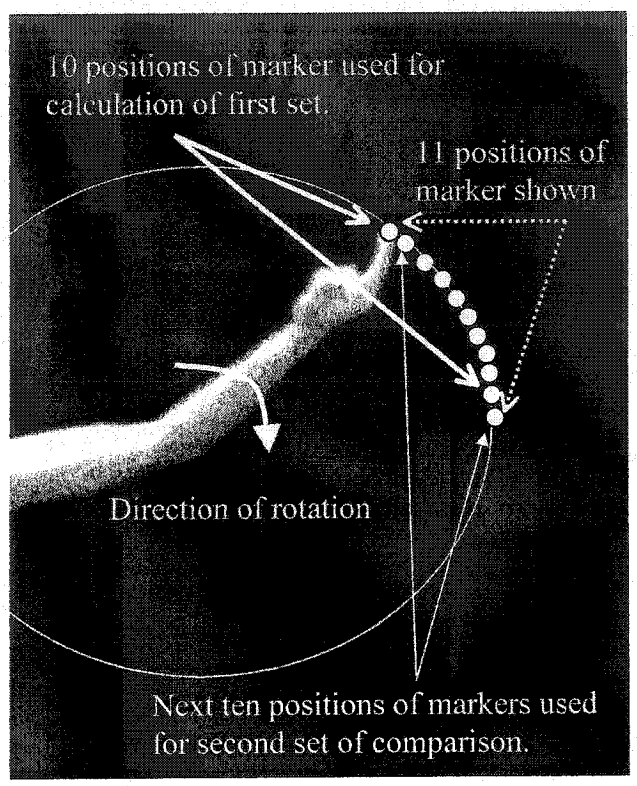


Figure 5.1 Illustration of the arm with a marker installed on the index finger, as used to evaluate the accuracy of the elbow joint based on the instantaneous centre of rotation of the forearm.

5.1.2 Wrist-joint-accuracy test

An experiment similar to the one explained in Section 5.1.1 was performed to determine the accuracy of the wrist joint position computed by the markerless tracking technique. This computed position was compared to the position of the wrist joint calculated by the Instantaneous Centre of Rotation (ICR) technique, which in this case, was performed to calculate the ICR of the hand. For this test, a subject rotated their hand within a plane by an angle of 90 deg, while keeping the upper-arm and forearm stationary. The path of the index finger marker was tracked using marker-based tracking. The calculated wrist joint coordinates by the markerless tracking technique, averaged over ten frames were compared to the ICR of the hand for the same ten frames. Sixty-five repeated tests for the wrist joint were carried out as for the elbow accuracy test. An illustration of the wrist joint accuracy test is found in Figure 5.2 with the positions of the index-finger marker in adjacent frames shown with enlarged spacing for clarity.

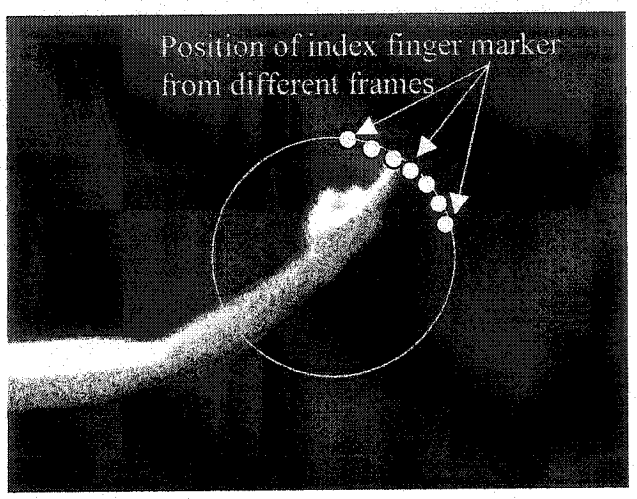


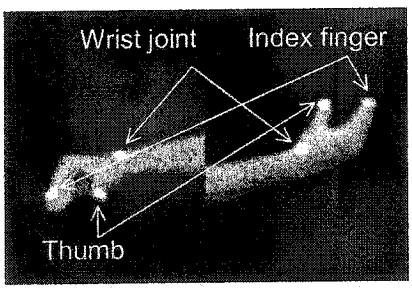
Figure 5.2 Illustration of the arm with a marker installed on the index finger, as used to evaluate the accuracy of the wrist-joint position based on instantaneous centre of rotation.

5.2 Comparison between markerless and marker-based tracking

An experiment was performed to determine the accuracy of the wrist joint centre, index-finger tip and thumb tip, calculated using the markerless hand-arm tracking, by comparing these positions with the centroids of white markers placed on the radial side of the wrist and index-finger, and dorsal side of the thumb (Figure 5.3), computed by marker-based tracking [2,3]. A subject moved their hand through the stereo-camera calibrated volume as described in Sections 5.2.1-5.2.3 that follow. The 3D positions of the centroids of the wrist, index finger and thumb markers were calculated using stereo-camera images of the hand, captured during the hand motion. The same sequence of images was used to compute the 3D positions of the wrist-joint centre, index-finger tip and thumb tip using the markerless hand-arm tracking technique. The differences in 3D positions of the wrist, index finger and thumb, for the two methods as described above, were calculated for each image frame.

It is important to note the differences in 3D positions calculated by the markerless and marker-based techniques. The tracking with markers computes a position based on the centre of the marker placed on top of the thumb and index finger, respectively, and the markerless tracking aims to find the tips of the finger and thumb, as shown in Figure 5.4. Therefore a difference in the 3D vector positions of the index finger and

thumb calculated by marker-based and markerless tracking is expected to be approximately 14.9 mm and 15.6 mm, respectively. The value of the expected difference for the index finger, 14.9 mm, is calculated as the vector sum of the distance (also referred to as the total distance) between the tip of the finger and the centroid of the marker along the Z'-axis, $R_{IF}^{z'} = 11.0\,\text{mm}$, and along the X'-axis, $R_{IF}^{x'} = 10.0\,\text{mm}$, respectively, as shown in Figure 5.4. The value of the expected difference for the thumb, 15.6 mm, is calculated as the vector sum of the distance between the tip of the thumb and the centroid of the marker along the Z'-axis, $R_T^{z'} = 12.0\,\text{mm}$, and along the X'-axis, $R_T^{x'} = 10.0$ mm, respectively, as shown in Figure 5.4. As the marker-based tracking estimates the centroid of the marker placed on the radial side of the wrist, and the markerless tracking determines the wrist centre, the difference between marker-based and markerless tracking positions is expected to be approximately 33 mm, which is half the radial-ulnar width of the arm at the wrist, $R_w = 66.0\,\text{mm}$. (Figure 5.4).



(a) (b)

Figure 5.3 Position of markers placed on the hand during the combined marker-based and markerless tracking test shown from (a) Camera View 1 and (b) Camera View 2.

Three separate tests were performed in which the subject performed a rotation of the hand about one axis, while translating the arm from one side of the calibrated volume to the opposite side, as explained below.

5.2.1 Rotation in *YZ* plane

The subject rotated their hand in *YZ* plane and translated it in 3D in the calibrated volume simultaneously, such that markers on the index finger and thumb were always visible. At

the start of the experiment, the hand was placed near one side of the calibrated volume and slowly moved to the opposite side, as shown in Figure 5.5a. The approximate path moved by the subject is depicted for illustrative purposes. While translating the hand, the hand was repeatedly rotated about the wrist in positive and negative directions approximately about the *X* axis in the *YZ* plane, as shown for part of the whole path in Figure 5.5b. All images of the human hand-arm motion were saved and then processed by both marker-based-tracking and markerless-tracking techniques to compare the positions of the wrist, index finger and thumb.

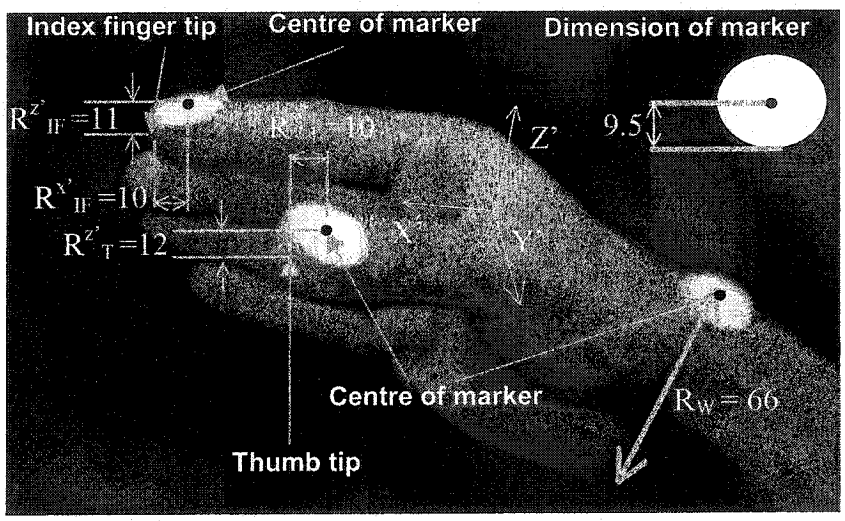


Figure 5.4 Position and orientation of markers on the hand used for the combined marker-based and markerless tracking test. All dimensions are in millimetres.

5.2.2 Rotation in *XY* plane

The subject rotated their hand in *XY* plane and translated it in 3D in the calibrated volume simultaneously, such that markers on the index finger and thumb were always visible. The hand was slowly moved across the calibrated volume on a similar path to the test described in the previous section, as shown in Figure 5.5a. While translating the hand, the hand was repeatedly rotated about the wrist in positive and negative directions approximately about the *Z* axis in the *XY* plane, as shown in Figure 5.5c. All images were processed by the marker-based-tracking and markerless-tracking techniques to compare positions of the wrist, index finger and thumb.

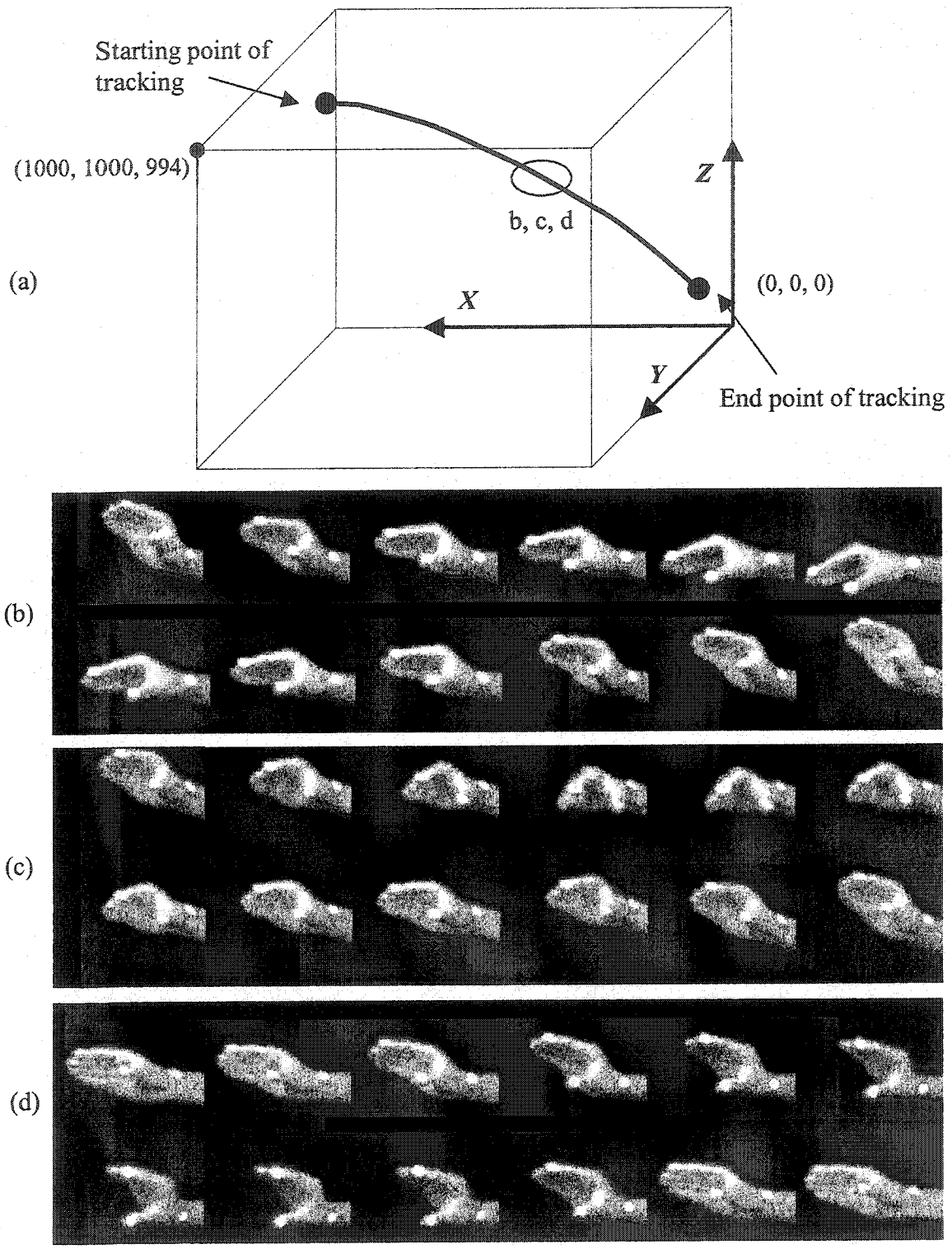


Figure 5.5 (a) Path of the hand moved in the calibration volume, shown as an approximation of the real path. A small portion of the path is exploded to show details of the rotation in the (b) *YZ* plane, (c) *XY* plane, and (d) *XZ* plane.

5.2.3 Rotation in *XZ* plane

A similar test to those described in Sections 5.2.1-5.2.2, was performed in which the hand rotation was approximately about the *Y* axis in the *XZ* plane while the hand translated across the calibrated volume, as shown in Figures 5.5a and 5.5d.

5.3 Teleoperation experiments

A series of tests were carried out to evaluate the ability of the vision-based markerless tracking to interface with the robot-manipulator in order to permit a human operator to control the movements of the robot in real-time in performing object manipulation tasks. Essentially the tests were not only evaluating, whether the markerless tracking accuracy was sufficient for teleoperation, but also the ability of the robot manipulator to copy human hand-arm motions, and the ability of the operator to use the human-robot-manipulator interface using the vision-based feedback system.

During each test, a subject moved their arm in the stereo-camera calibrated volume, in order to control the motion of the robot for a simple object manipulation task. The task involved picking up an object from a specified starting position and placing a specified corner of the object at a predefined corner location on a target with the object and target edges aligned, as shown in Figure 5.6. The object was a rigid plastic foam block, 200 mm in height, 100 mm wide, and 50 mm thick and the target was a 76.5 mm x 76.5 mm square paper. The specified corners of the object and target are shown in Figures 5.6-5.7. Two series of tests of the same object manipulation task were repeated five times. In the first series of tests (Teleoperation Test Series 1), the tool roll of the end-effector was kept fixed. In the second series of tests (Teleoperation Test Series 2), all the orientations of the robot end-effector were allowed. All robot manipulation tasks were carried out in real-time with the markerless tracking of the human hand. The position and orientation of the human hand computed based on the markerless tracking software developed in the current research and the position and orientation of the robot end-effector were recorded during the object manipulation task. This was used to determine how well the robot end-effector followed the hand-arm motion of subject in performing the desired task. The position of the object after it was placed on the target was also

recorded. The error in translation and rotation in placing the object on the target was also recorded.

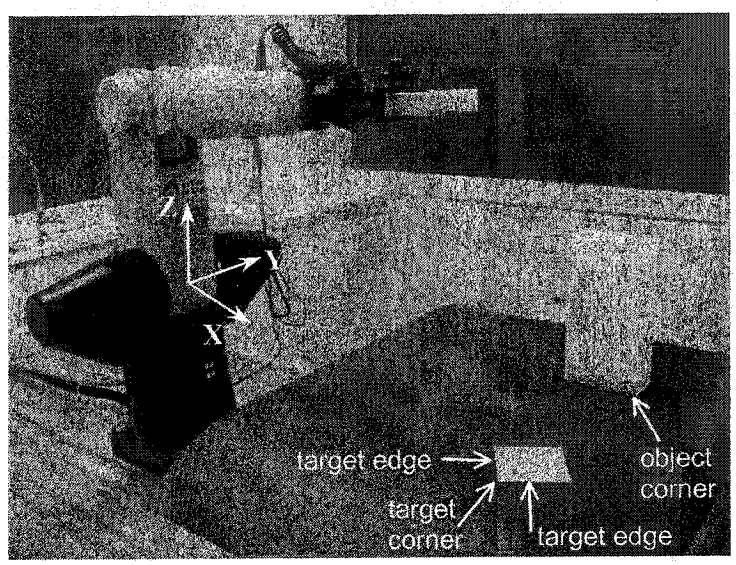


Figure 5.6 Experimental setup at the remote robot site showing the robot-manipulator, object starting position and target corner location [from [3] Kofman et al., 2003].

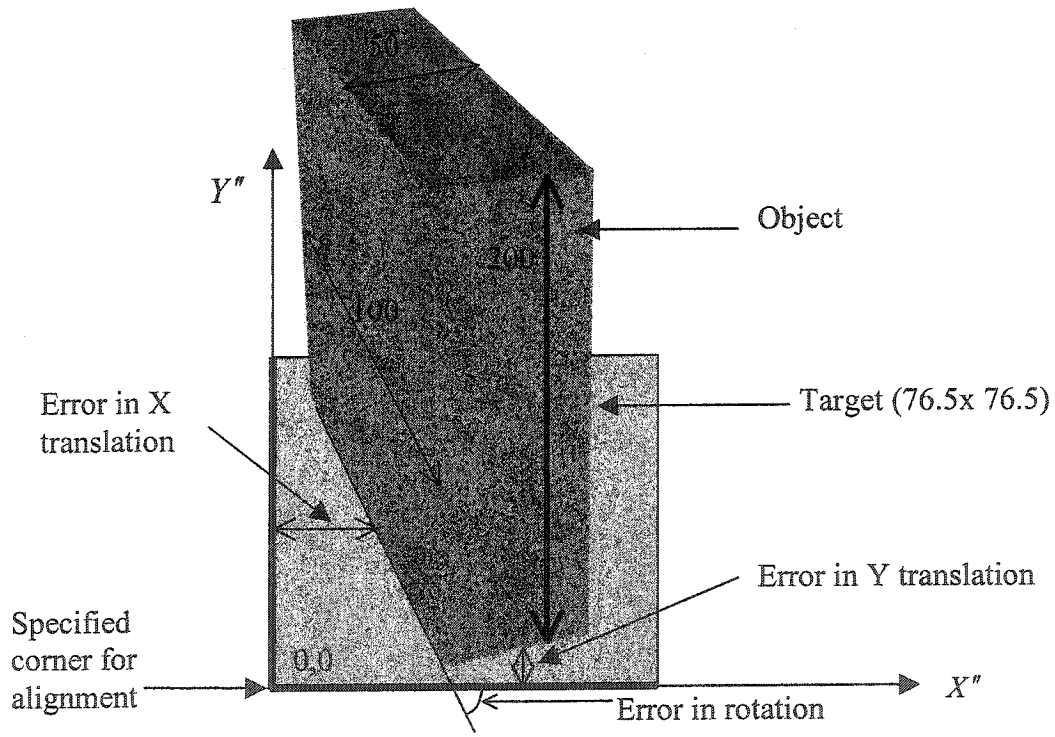


Figure 5.7 Schematic representation of object and target used in teleoperation experiment showing the errors in translation and rotation. All dimensions are in millimetres.

CHAPTER 6. RESULTS

6.1 Instantaneous-centre-of-rotation test

6.1.1 Elbow-joint-accuracy test

The results of the experiment performed to determine the accuracy of the elbow joint centre computed by the markerless tracking technique, by comparing it to the elbow joint centre computed by the ICR technique, assumed as ground truth, is shown in Table 6.1 and Figure 6.1. The results were based on 75 images. The centre of the best-fit circle $E_{ICR}(x, y, z)$, calculated using a set of 10 points of the index-finger position, as shown in Equation 5.2, was compared to the average position of the elbow joint over the same 10 frames, calculated using the markerless tracking technique $\overline{E}_{ML}(x, y, z)$, as shown in Equation 5.1. The absolute error $E_e(x, y, z)$ was calculated using Equation 5.4.

The average absolute error over the 65 tests was highest in *X*, 10.8 mm, compared to those for *Y*, 7.8 mm, and *Z*, 8.5 mm, respectively. The maximum value of the absolute error was highest for *Z*, 24.9 mm, compared to those for *X* and *Y*, 20.5 mm and 19.0 mm, respectively. The minimum values of the absolute error were only 0.1 mm, 0.2 mm and 0.1 mm in *X*, *Y* and *Z*, respectively. The average total error was 16.9 mm with a standard deviation of 6.8 mm. The maximum and minimum values of the total errors were 33.2 mm and 5.4 mm, respectively.

Table 6.1 Results of ICR test for the elbow joint based on 65 joint-centre computations.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute error $E_e(x, y, z)$	10.8	7.8	8.5	16.9
Standard Deviation (absolute error)	5.3	4.7	5.8	6.8
Maximum absolute error	20.5	19.0	24.9	33.2
Minimum absolute error	0.1	0.2	0.1	5.4

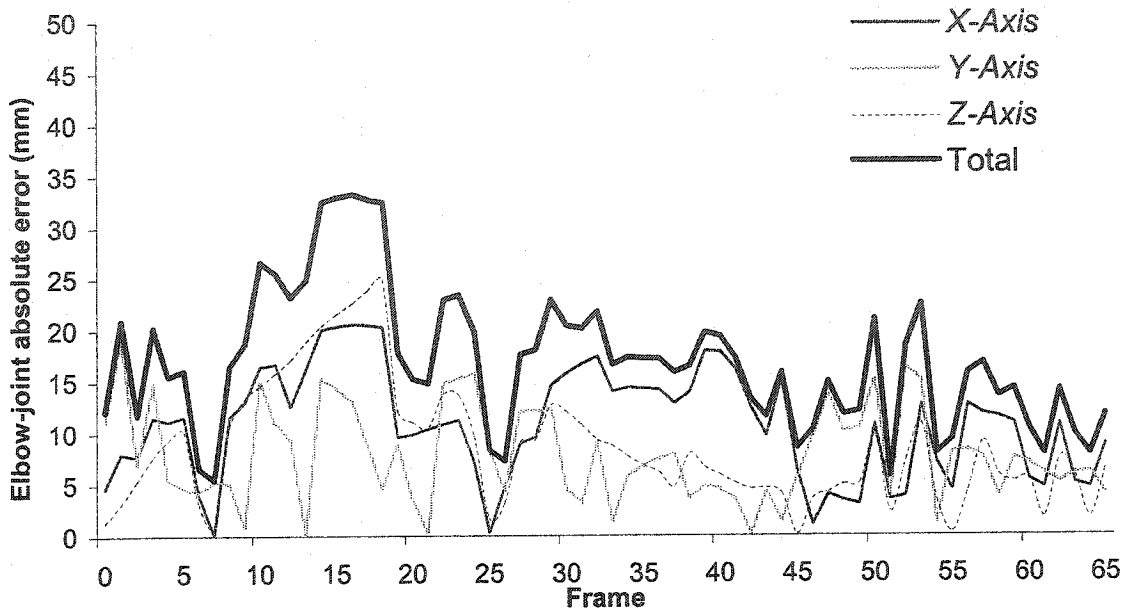


Figure 6.1 Elbow-joint absolute errors $E_e(x, y, z)$ computed by differences between $\overline{E}_{ML}(x, y, z)$ and $E_{ICR}(x, y, z)$ over 65 joint-centre tests.

6.1.2 Wrist-joint-accuracy test

The results of the experiment performed to determine the accuracy of the wrist-joint centre computed by the markerless tracking technique, by comparing the value to the wrist-joint centre computed by the ICR technique, assumed as ground truth, is shown in Table 6.2 and Figure 6.2. The errors were computed in similar manner to the elbow joint errors, based on 75 images.

The average absolute error over the 65 tests was highest in *X*, 13.6 mm, compared to those for *Y*, 7.6 mm, and *Z*, 12.0 mm, respectively. The maximum value of the absolute error was highest for *Y*, 32.1 mm, compared to those for *X* and *Z*, 27.2 mm and 30.6 mm, respectively. The minimum values of the absolute error were only 0.1 mm in *X*, *Y* and *Z*, respectively. The average total error was 21.7 mm with a standard deviation of 9.7 mm. The maximum and minimum values of the total errors were 41.5 mm and 2.0 mm, respectively.

Table 6.2 Results of ICR test for the wrist joint based on 65 joint-centre computations.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute error	13.6	7.6	12.0	21.7
Standard Deviation (absolute error)	7.5	6.7	5.5	9.7
Maximum absolute error	27.2	32.1	30.6	41.5
Minimum absolute error	0.1	0.1	0.1	2.0

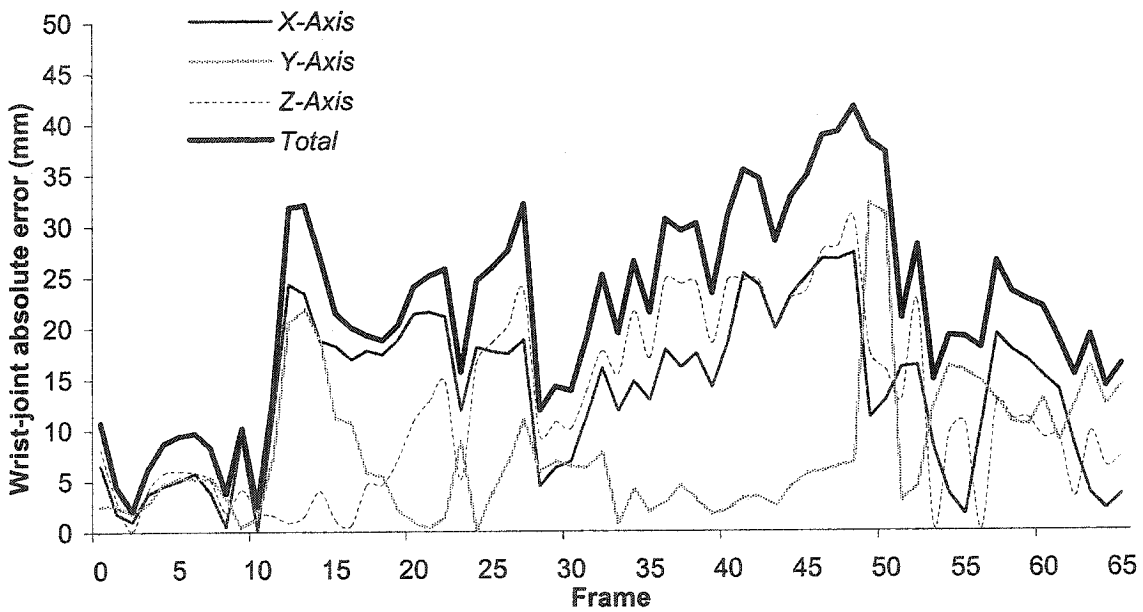


Figure 6.2 Wrist-joint absolute errors determined from differences between the wrist positions computed by markerless and ICR techniques, over 65 joint-centre tests.

6.2 Comparison between markerless and marker-based tracking

6.2.1 Rotation in *YZ* plane

The results of the experiment performed to evaluate the accuracy of the computation of the wrist joint, index-finger tip and thumb tip, using the markerless hand-arm tracking technique, in comparison with the marker-based tracking, are shown in Tables 6.3-6.5 and Figures 6.3-6.6, respectively. (Note the different scale used for position differences). The 3D positions of the hand-arm movement are plotted for the markerless tracking, marker-based tracking and the difference between markerless and marker-based tracking

on the same graph, for the wrist in *X*, *Y*, and *Z* in Figures 6.3a-c, respectively; for the index finger in *X*, *Y*, and *Z* in Figures 6.4a-c, respectively; and for the thumb in *X*, *Y*, and *Z* in Figures 6.5a-c, respectively. The total differences (vector sum of *X*, *Y* and *Z*) for the wrist, index finger and thumb are shown in Figures 6.6a-c, respectively.

The average absolute differences over 60 frames for the wrist joint were greatest in *X*, 26.7 mm, compared to 9.1 mm and 4.5 mm in *Y* and *Z*, respectively (Table 6.3). The maximum absolute difference was greatest in *X*, 33.3 mm, compared to 25.0 mm and 11.9 mm, in *Y* and *Z*, respectively. The minimum absolute differences were smallest in *Y* and *Z*, 0.3 mm and 0.1 mm, respectively, compared to 18.7 mm in *X*. As explained in Section 5.2, a high value of the difference between the computed value of the wrist joint (markerless based) and the ground truth-value (position of the wrist marker), is expected to be equal to half the width of the operator wrist, 33 mm. This expected difference value of 33 mm is usually distributed along the three axes. The parameter that can most readily be compared to these expected differences for any orientation of the hand and arm is the total difference (vector sum) between markerless and marker-based computed positions. The average total difference over 60 frames for the wrist joint was 29.3 mm (Table 6.3), which is very close to the expected value, 33 mm, described in Section 5.2. The maximum value of the total difference was 36.3 mm, while the minimum value of the total difference was 21.9 mm.

Table 6.3 Results of wrist joint computation for hand rotation in the *YZ* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	26.7	9.1	4.5	29.3
Standard Deviation	3.0	5.7	3.2	3.2
Maximum absolute difference	33.3	25.0	11.9	36.3
Minimum absolute difference	18.7	0.3	0.1	21.9

The average absolute differences over 60 frames for the index-finger joint were greatest in *Y*, 17.6 mm, compared to 10.9 mm and 5.8 mm in *X* and *Z*, respectively

(Table 6.4). The maximum absolute difference was greatest in *Y*, 32.6 mm, compared to 14.6 mm and 13.3 mm, in *X* and *Z*, respectively. The minimum absolute difference was smallest in *Z*, 0.3 mm, compared to 7.2 mm and 5.5 mm in *X* and *Y*, respectively. As explained in Section 5.2, the difference between the computed value of the index-finger tip (markerless based) and the ground truth-value (position of the index-finger marker) is expected to be equal to 14.9 mm. The average total difference over 60 frames for the index-finger joint was 22.3 mm (Table 6.4), which represents a 7.4 mm error from the expected value, 14.9 mm, described in Section 5.2. The maximum value of the total difference was 36.8 mm, while the minimum value of the total difference was 13.8 mm.

Table 6.4 Results of index-finger joint computation for hand rotation in the *YZ* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	10.9	17.6	5.8	22.3
Standard Deviation	2.0	7.3	3.5	5.7
Maximum absolute difference	14.6	32.6	13.3	36.8
Minimum absolute difference	7.2	5.5	0.3	13.8

The average absolute differences over 60 frames for the thumb joint were greatest in *X* and *Z*, both 18.6 mm, compared to the similar value of 16.5 mm in *Y* (Table 6.5). The maximum absolute difference was greatest in *Y* and *Z*, 26.0 mm and 27.9 mm, respectively, compared to 23.3 mm in *X*. The minimum absolute difference was smallest in *Y*, 0.2 mm, compared to 7.6 mm and 10.6 mm in *Z* and *X*, respectively. As explained in Section 5.2, the difference between the computed value of the thumb tip (marker-based) and the ground truth-value (position of the thumb marker) is expected to be 15.6 mm. The average total difference over 60 frames for the thumb joint was 31.6 mm (Table 6.5), which is more than the expected value, 15.6 mm, described in Section 5.2. The maximum total difference was 43.4 mm, while the minimum total difference was 15.3 mm.

Table 6.5 Results of thumb joint computation for hand rotation in the *YZ* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	18.6	16.5	18.6	31.6
Standard Deviation	2.7	6.1	5.0	6.6
Maximum absolute difference	23.3	26.0	27.9	43.4
Minimum absolute difference	10.6	0.2	7.6	15.3

6.2.2 Rotation in *XY* plane

The results of the experiment performed to evaluate the accuracy of the computation of the wrist joint, index-finger tip and thumb tip, using the markerless hand-arm tracking technique, in comparison with the marker-based tracking, for rotation in the *XY* plane are shown in Tables 6.6-6.8 and Figures 6.7-6.10, respectively. (Note the different scale used for position differences). The 3D positions of the hand-arm movement are plotted for the markerless tracking, marker-based tracking and the difference between markerless and marker-based tracking on the same graph, for the wrist in Figure 6.7, for the index finger in Figure 6.8, and for the thumb in Figure 6.9. The total differences for the wrist, index finger and thumb are shown in Figures 6.10a-c, respectively.

The average absolute differences over 60 frames for the wrist joint were greatest in *X*, 25.9 mm, compared to 7.0 mm and 10.8 mm in *Y* and *Z*, respectively (Table 6.6). The maximum absolute difference was greatest in *X*, 33.9 mm, compared to 15.1 mm and 18.0 mm, in *Y* and *Z*, respectively. The minimum absolute difference was smallest in *Y*, 0.9 mm, compared to 3.3 mm in *Z* and 17.9 mm in *X*, respectively. The average total difference over 60 frames for the wrist joint was 29.5 mm, which is very close to the expected value, 33 mm, described in Section 5.2. The maximum value of the total difference was 37.9 mm, while the minimum value of the total difference was 25.2 mm.

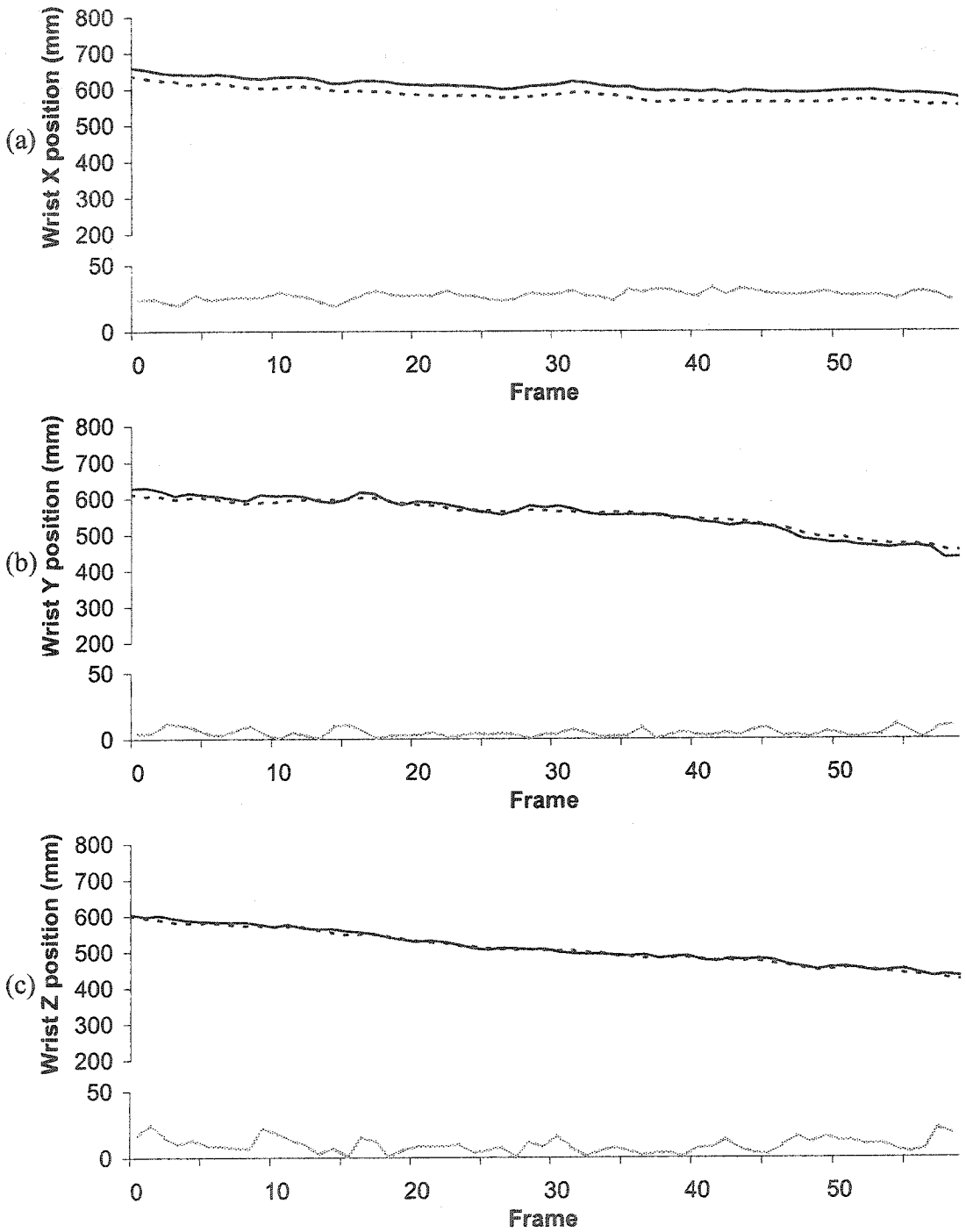


Figure 6.3 Wrist position for motion in *YZ* plane for (———) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z* axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

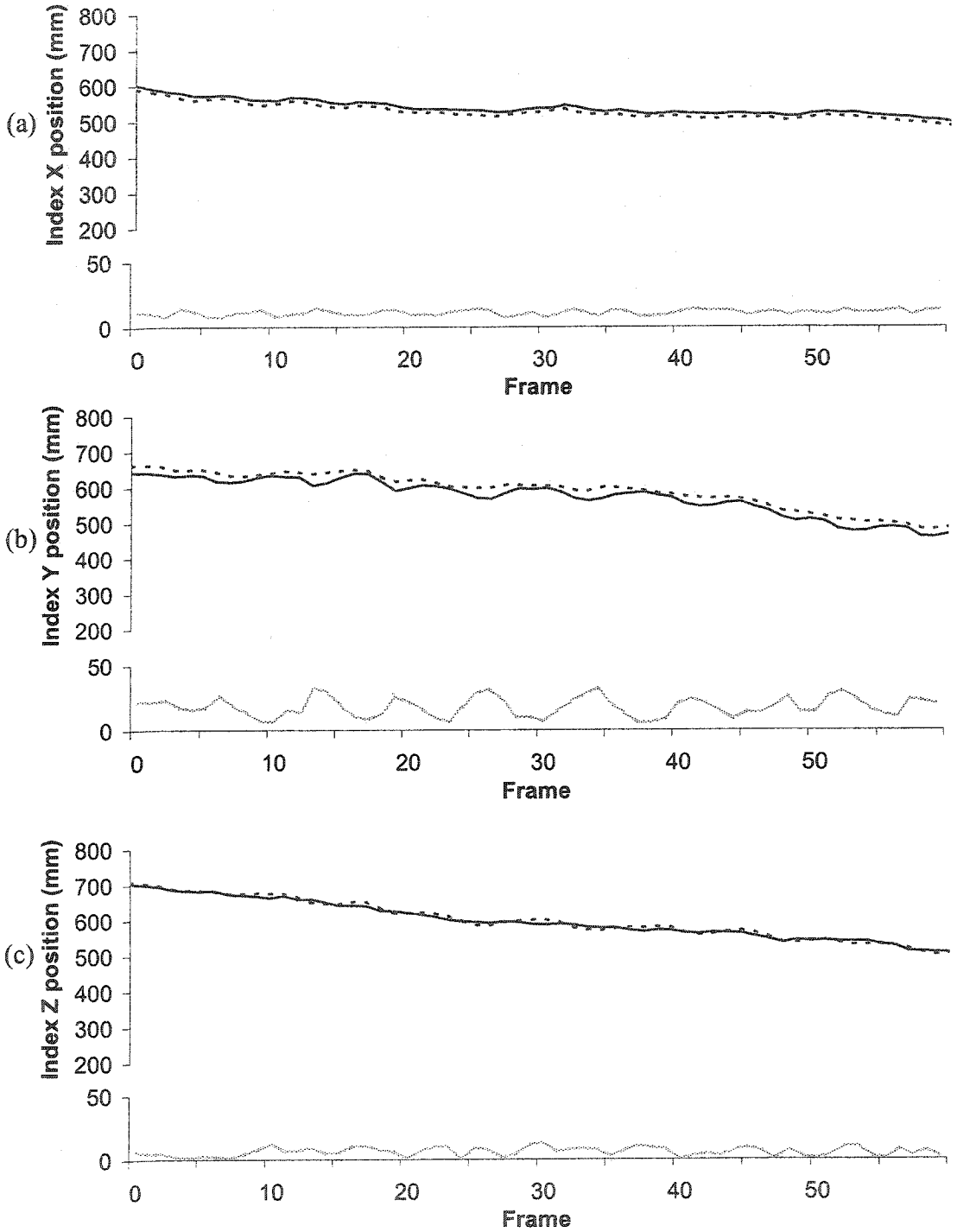


Figure 6.4 Index-finger position for motion in *YZ* plane for (——) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z* axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

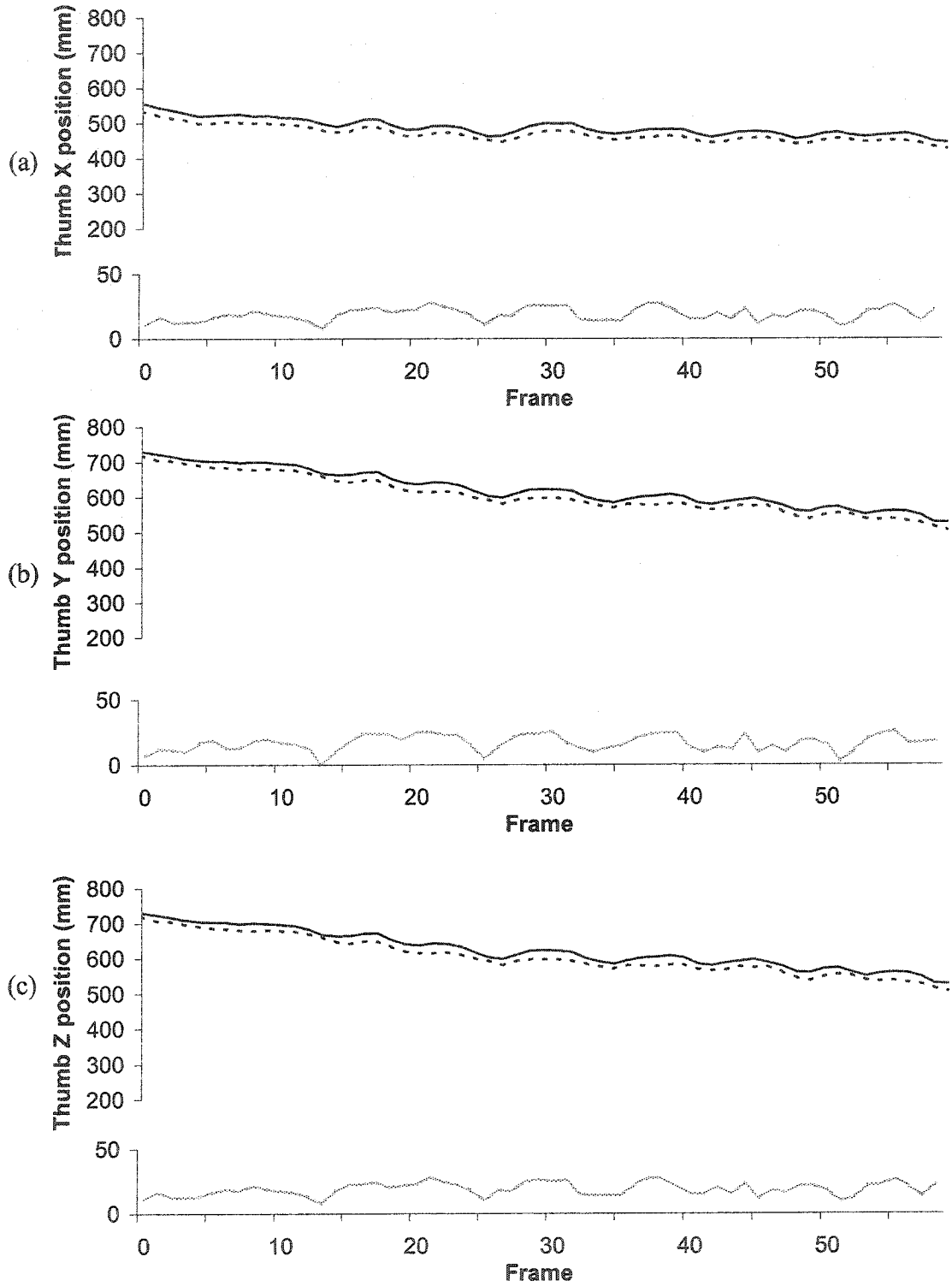


Figure 6.5 Thumb position for motion in *YZ* plane for (————) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z*-axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

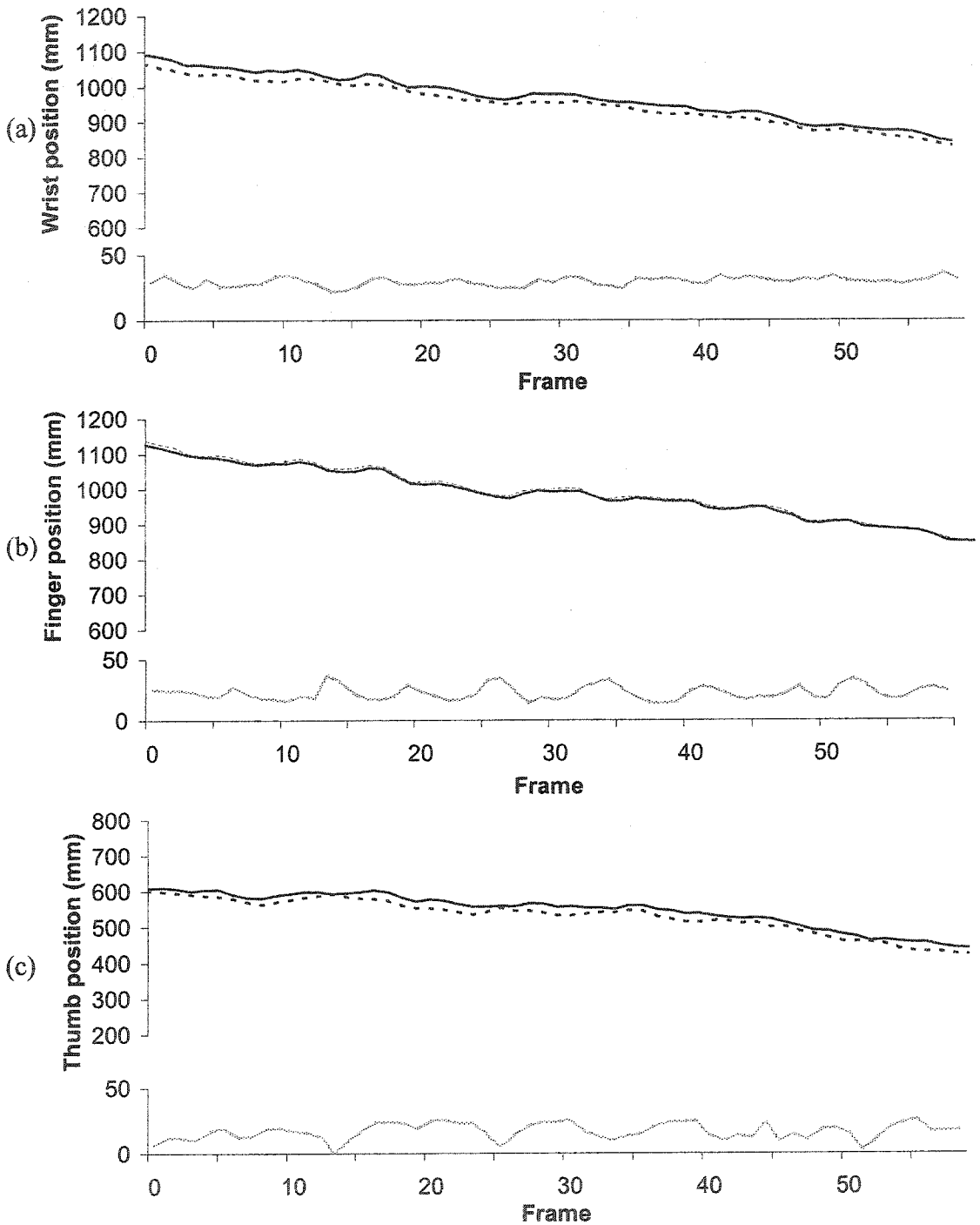


Figure 6.6 Total (vector) position of points of the hand in *YZ* plane for (——) marker-based tracking, (------) markerless tracking and (·····) total difference between marker-based and markerless tracking for: (a) wrist, (b) index finger and (c) thumb. Points for the marker-based and markerless tracking are not identical and are expected to have differences (see Sections 5.2 and 7.2.2).

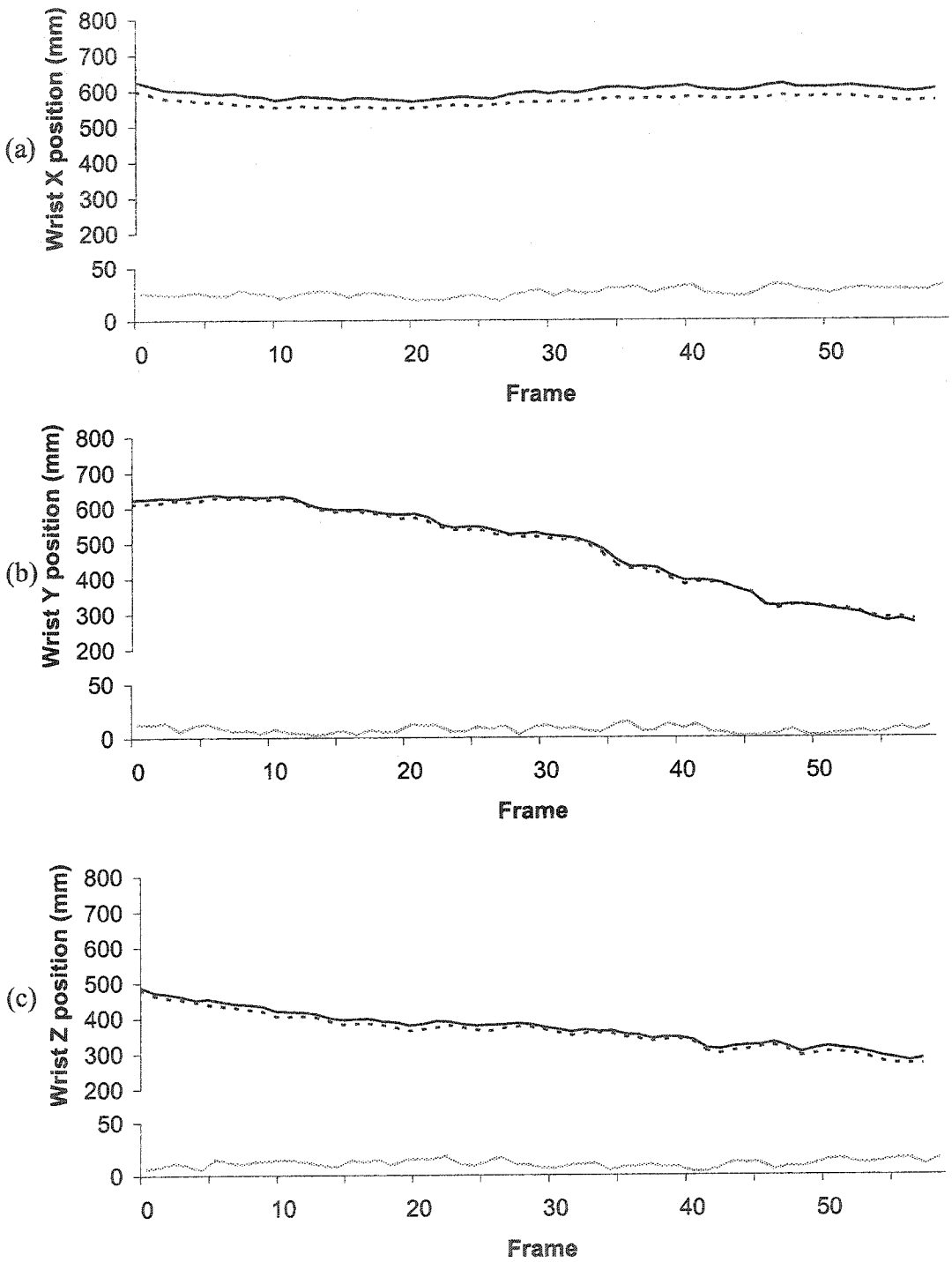


Figure 6.7 Wrist position for motion in *XY* plane for (——) marker-based tracking, (----) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z* axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

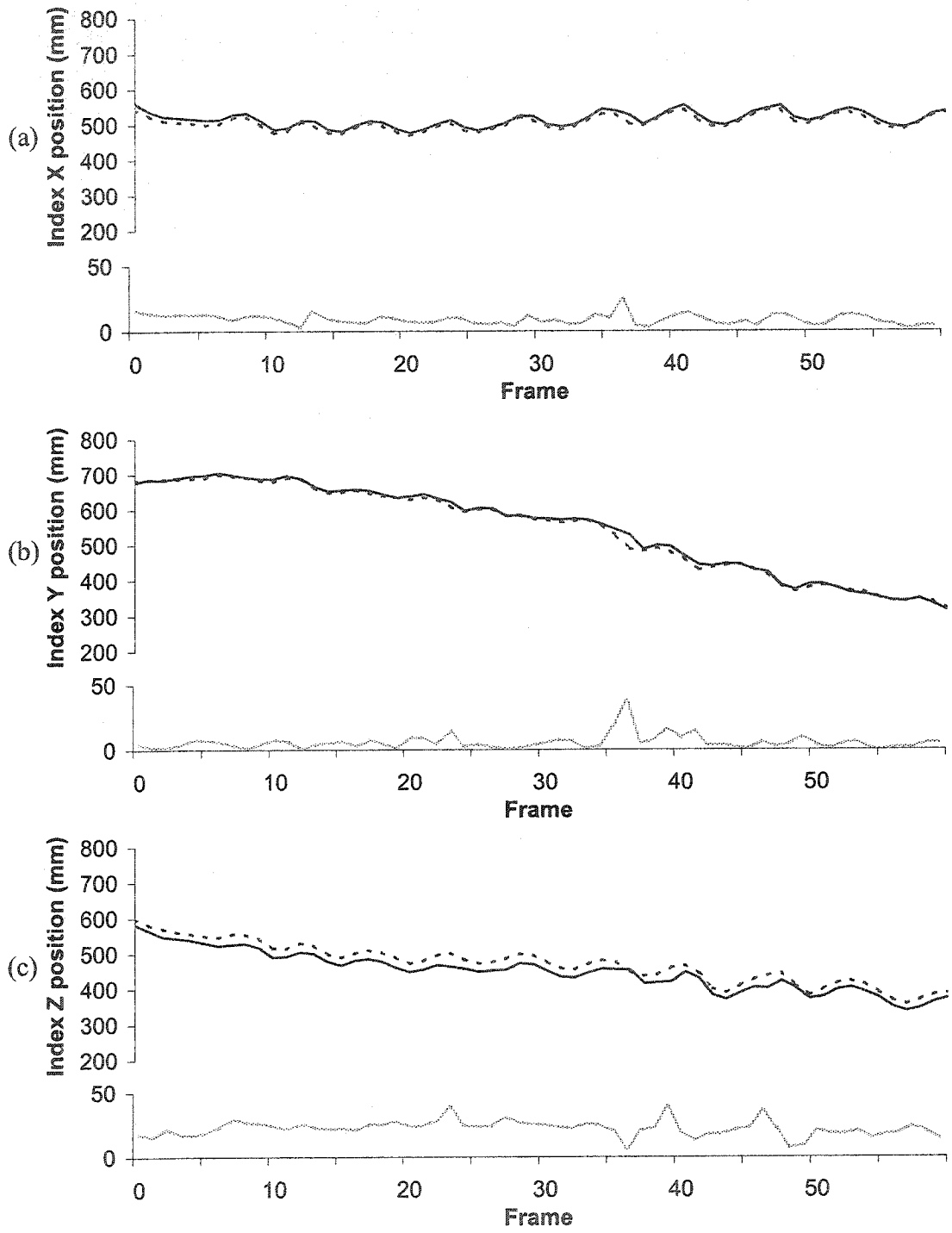


Figure 6.8 Index-finger position for motion in *XY* plane for (———) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z* axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

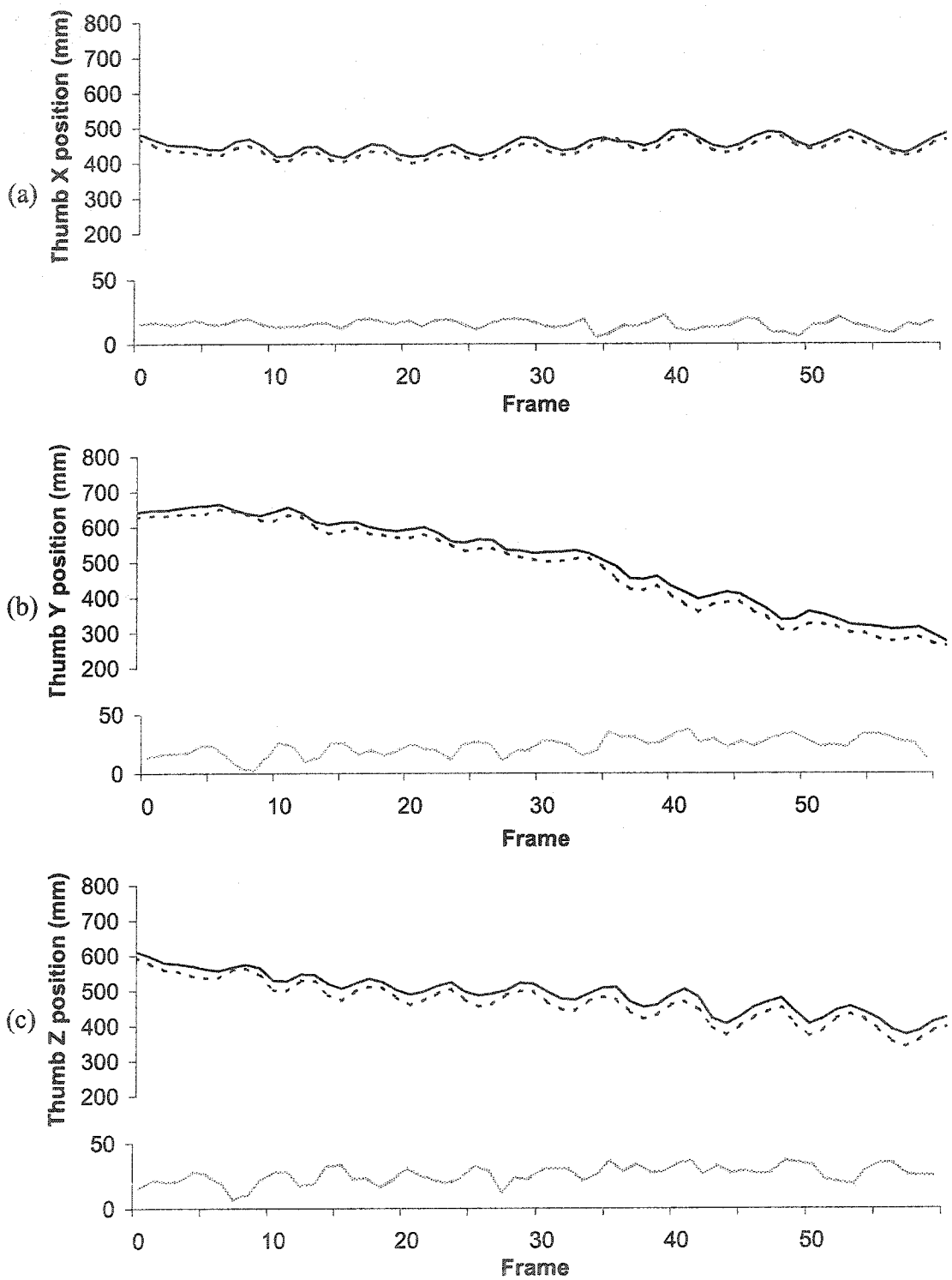


Figure 6.9 Thumb position for motion in *XY* plane for (———) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z* axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

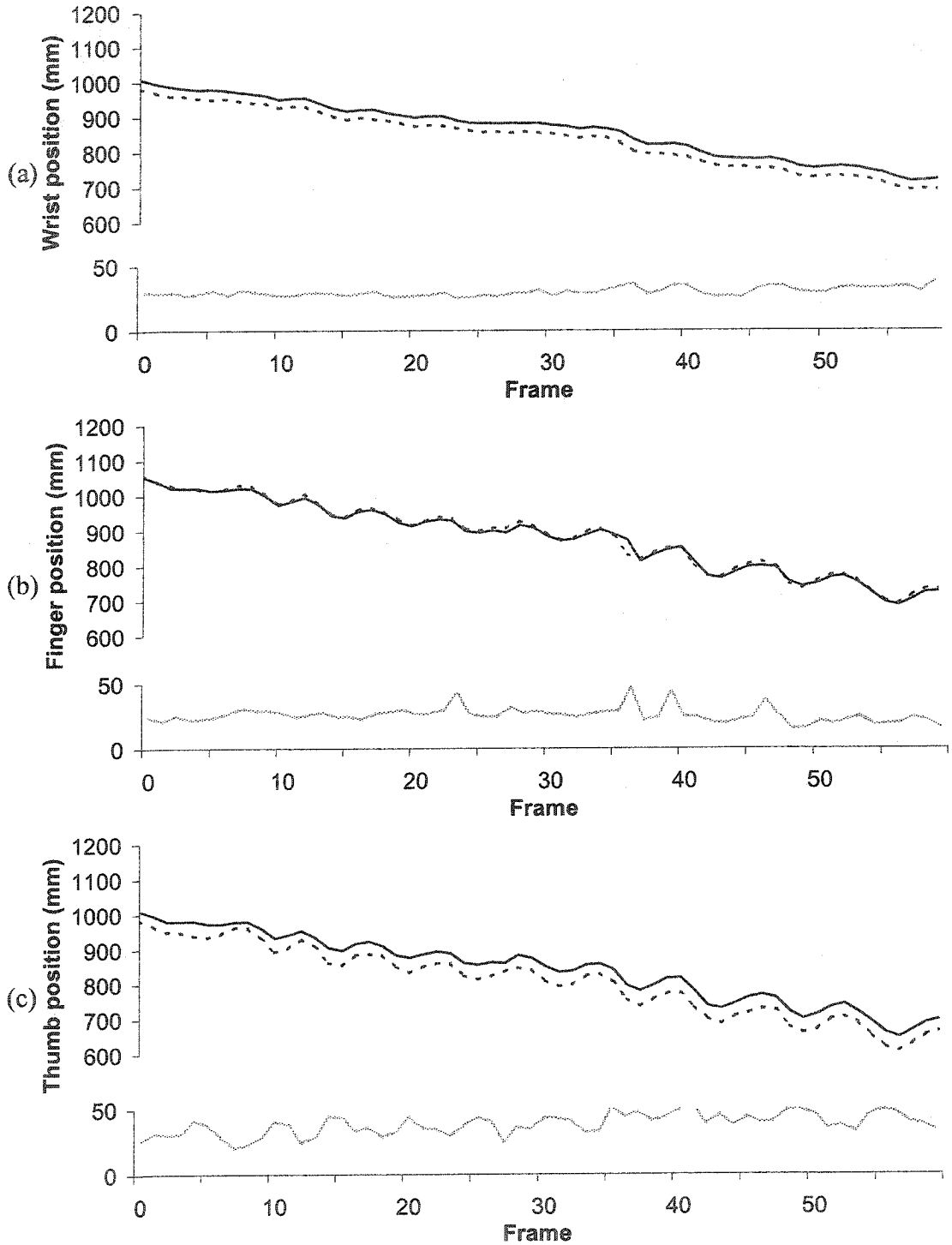


Figure 6.10 Total position of points of the hand in *XY* plane for (——) marker-based tracking, (------) markerless tracking and (·····) total difference between marker-based and markerless tracking for: (a) wrist, (b) index finger and (c) thumb. Points for the marker-based and markerless tracking are expected to be different (Secs. 5.2 and 7.2.2).

Table 6.6 Results of wrist joint computation for hand rotation in the *XY* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	25.9	7.0	10.8	29.5
Standard Deviation	3.8	3.7	3.5	2.9
Maximum absolute difference	33.9	15.1	18.0	37.9
Minimum absolute difference	17.9	0.9	3.3	25.2

The average absolute differences over 60 frames for the index-finger joint were greatest in *Z*, 21.9 mm, compared to 8.8 mm and 5.5 mm in *Y* and *Z*, respectively (Table 6.7). The maximum absolute differences were greatest in *Z* and *Y*, 39.4 mm and 38.8 mm, respectively, compared to 24.8 mm in *X*. The minimum absolute difference was smallest in *Y*, 0.1 mm, compared to 1.4 mm and 6.2 mm in *X* and *Z*, respectively. The average total difference over 60 frames for the index-finger joint was 25.4 mm, which represents a 10.5 mm error from the expected value, 14.9 mm, described in Section 5.2. The maximum value of total difference was 46.4 mm, while the minimum value of the total difference was 15.2 mm.

Table 6.7 Results of index-finger joint computation for hand rotation in the *XY* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	8.8	5.5	21.9	25.4
Standard Deviation	4.2	5.8	6.2	5.8
Maximum absolute difference	24.8	38.8	39.4	46.4
Minimum absolute difference	1.4	0.1	6.2	15.2

The average absolute differences over 60 frames for the thumb joint were greatest in *Z*, 25.8 mm, compared to 14.9 mm and 22.7 mm in *X* and *Y*, respectively (Table 6.8). The maximum absolute difference was greatest in *Y* and *Z*, 37.4 mm and 36.7 mm, respectively, compared to 22.4 mm in *X*. The minimum absolute difference was smallest in *Y*, 2.0 mm, compared to 5.2 mm and 7.1 mm in *X* and *Z*, respectively. The average

total difference over 60 frames for the thumb joint was 38.3 mm, which is more than the expected value, 15.6 mm, described in Section 5.2. The maximum total difference was 49.5 mm, while the minimum value was 20.2 mm.

Table 6.8 Results of thumb joint computation for hand rotation in the *XY* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	14.9	22.7	25.8	38.3
Standard Deviation	3.7	7.5	6.6	7.6
Maximum absolute difference	22.4	37.4	36.7	49.5
Minimum absolute difference	5.2	2.0	7.1	20.2

6.2.3 Rotation in *XZ* plane

The results of the experiment performed to evaluate the accuracy of the computation of the wrist joint, index-finger tip and thumb tip, using the markerless hand-arm tracking technique, in comparison with the marker-based tracking, for rotation in *XZ* plane are shown in Tables 6.9-6.11 and Figures 6.11-6.14, respectively. (Note the different scale used for position differences). The 3D positions of the hand-arm movement are plotted for the markerless tracking, marker-based tracking and the difference between markerless and marker-based tracking on the same graph, for the wrist in Figure 6.11, for the index finger in Figure 6.12, and for the thumb in Figure 6.13. The total differences for the wrist, index finger and thumb are shown in Figures 6.14a-c, respectively.

The average absolute differences over 60 frames for the wrist joint were greatest in *X*, 33.6 mm, compared to 5.6 mm and 8.6 mm in *Y* and *Z*, respectively (Table 6.9). The maximum absolute difference was greatest in *X*, 45.5 mm, compared to 16.8 mm and 21.4 mm, in *Y* and *Z*, respectively. The minimum absolute differences were smallest in *Y* and *Z*, 0.2 mm and 0.1 mm, respectively, compared to 20.9 mm in *X*. The average total difference over 60 frames for the wrist joint was 36.2 mm, which is very close to the expected value, 33 mm, described in Section 5.2. The maximum value of total difference was 46.1 mm, while the minimum value of the total difference was 26.6 mm.

Table 6.9 Results of wrist joint computation for hand rotation in the *XZ* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	33.6	5.6	8.6	36.2
Standard Deviation	6.2	4.0	6.4	4.6
Maximum absolute difference	45.5	16.8	21.4	46.1
Minimum absolute difference	20.9	0.2	0.1	26.6

The average absolute differences over 60 frames for the index-finger joint were greatest in *Z*, 20.6 mm, compared to 9.5 mm and 5.3 mm in *Y* and *Z*, respectively (Table 6.10). The maximum absolute difference was greatest in *Z*, 33.7 mm, compared to 21.9 mm and 27.5 mm, in *X* and *Y*, respectively. The minimum absolute difference was smallest in *Y* and *X*, 0.2 mm and 0.6 mm, respectively, compared to 9.5 mm in *Z*. The average total difference over 60 frames for the index-finger joint was 24.2 mm, which represents a 9.3 mm error from the expected value, 14.9 mm, described in Section 5.2. The maximum value of the total difference was 40.2 mm, while the minimum value of the total difference was 12.7 mm.

Table 6.10 Results of index-finger joint computation for hand rotation in the *XZ* plane in markerless / marker-based tests over 60 frames.

	X (mm)	*Y* (mm)	*Z* (mm)	Total (mm)
Average absolute difference	9.5	5.3	20.6	24.2
Standard Deviation	4.8	5.0	5.1	6.0
Maximum absolute difference	21.9	27.5	33.7	40.2
Minimum absolute difference	0.6	0.2	9.5	12.7

The average absolute differences over 60 frames for the thumb joint were similar in *X*, *Y*, and *Z*. The values were greatest in *Z*, 13.5 mm, compared to 10.2 mm and 12.5 mm in *X* and *Y*, respectively (Table 6.11). The maximum absolute difference was greatest in *Y* and *Z*, 36.0 mm and 33.7 mm, respectively, compared to 18.6 mm in *X*. The

minimum absolute difference was smallest in Y, 0.1 mm, compared to 2.9 mm and 0.8 mm in X and Z, respectively. The average total difference over 60 frames for the thumb joint was 22.1 mm, which is more than the expected value, 15.6 mm, described in Section 5.2. The maximum value of the total difference was 47.0 mm, while the minimum value of the total difference was 6.1 mm.

Table 6.11 Results of thumb joint computation for hand rotation in the XZ plane in markerless / marker-based tests over 60 frames.

	X (mm)	Y (mm)	Z (mm)	Total (mm)
Average absolute difference	10.2	12.5	13.5	22.1
Standard Deviation	3.8	8.3	9.4	10.4
Maximum absolute difference	18.6	36.0	33.7	47.0
Minimum absolute difference	2.9	0.1	0.8	6.1

6.3 Teleoperation experiments

During the teleoperation test, a subject moved their arm in the calibrated volume to control the robot's motion, in order to perform the task of moving an object from a predefined location to a target, using the vision-based feedback system. The task was to pick an object from a known position and place the specified corner of the object at a specified corner location on the target with the object and target edges aligned, as shown in Figures 5.6 and 5.7.

The results for positioning of the object on the target were computed for both teleoperation tests series, Teleoperation Test Series1, in which tool roll of the end-effector was kept fixed and Teleoperation Test Series 2, in which all orientations of the end-effector were allowed. The results for first and second teleoperation series of tests are shown in Tables 6.12 and 6.13. The average absolute errors over the 5 tests for Teleoperation Test Series 1 were 11.6 mm, 11.4 mm, and 11.0 deg, in X'' and Y'' translation, and Z'' rotation, respectively, relative to the target, as shown in Figure 5.7. The maximum errors in placing the object on the target along X'' and Y'' axes were

19.0 mm and 25.0 mm, respectively, with the maximum error in rotation 18.0 deg. The minimum errors in placing the object on the target were only 5.0 mm and 3.0 mm along X'' and Y'' axes, respectively, with the minimum error in rotation 4.5 deg. The average absolute errors over the 3 tests for the Teleoperation Test Series 2 were 33.7 mm, 29.0 mm, and 30.0 deg, in X'' and Y'' translation, and Z'' rotation, respectively, relative to the target (Figure 5.7). The maximum errors in placing the object on the target along X'' and Y'' axes were 48.0 mm and 42.0 mm, respectively, with the maximum error in rotation 33.0 deg. The minimum errors in placing the object on the target were 9.0 mm and 10.0 mm along X'' and Y'' axes, respectively, with the minimum error in rotation 28.0 deg.

Table 6.12 Teleoperation Test Series 1 - Positioning errors for combined object translation and rotation with tool roll of the end-effector fixed.

	Error X'' (mm)	**Error Y'' (mm)**	**Error Z'' (deg)**
Test 1	15.0	25.0	9.0
Test 2	5.0	10.0	4.5
Test 3	8.0	3.0	18.0
Test 4	11.0	10.0	9.5
Test 5	19.0	9.0	14.0
Mean absolute error	**11.6**	**11.4**	**11.0**
SD (absolute error)	**5.5**	**8.1**	**5.1**

Table 6.13 Teleoperation Test Series 2 - Positioning errors for combined object translation and rotation with all end-effector rotations allowed.

	Error X'' (mm)	**Error Y'' (mm)**	**Error Z'' (deg)**
Test 1	44.0	10.0	28.0
Test 2	9.0	42.0	33.0
Test 3	48.0	35.0	29.0
Mean absolute error	**33.7**	**29.0**	**30.0**
SD (absolute error)	**21.5**	**16.8**	**2.6**

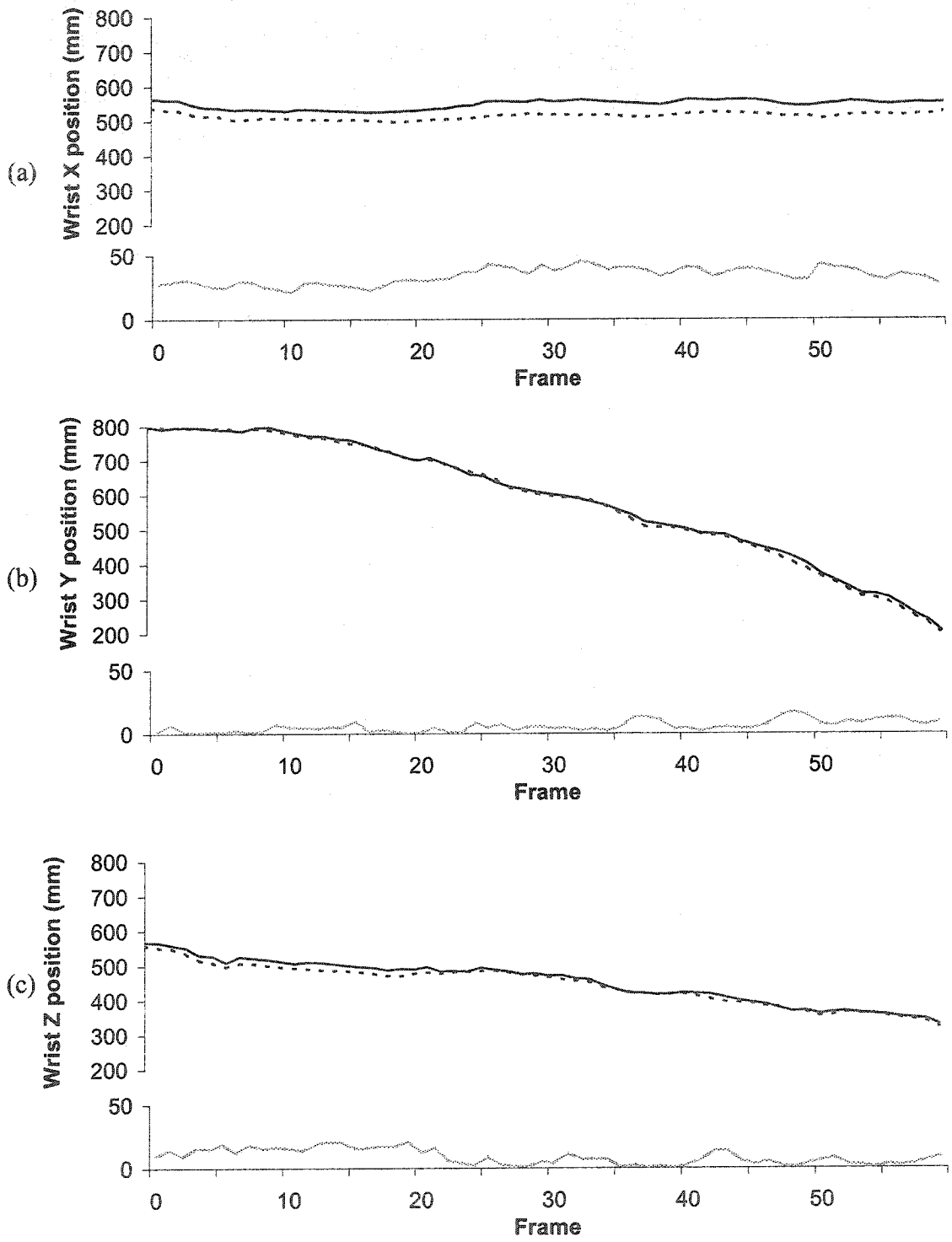


Figure 6.11 Wrist position for motion in *XZ* plane for (————) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z*-axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

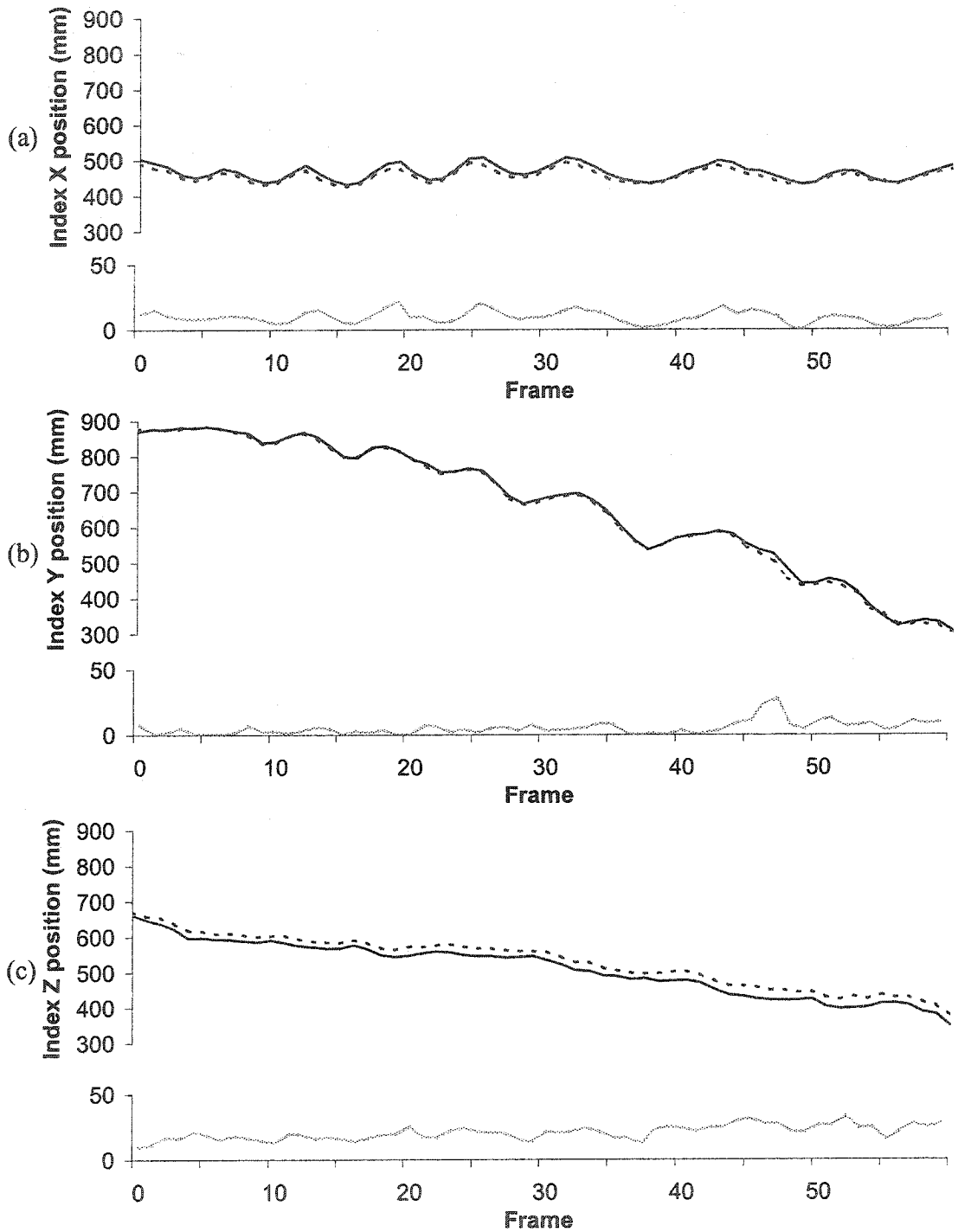


Figure 6.12 Index-finger position for motion in *XZ* plane for (———) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z*-axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2).

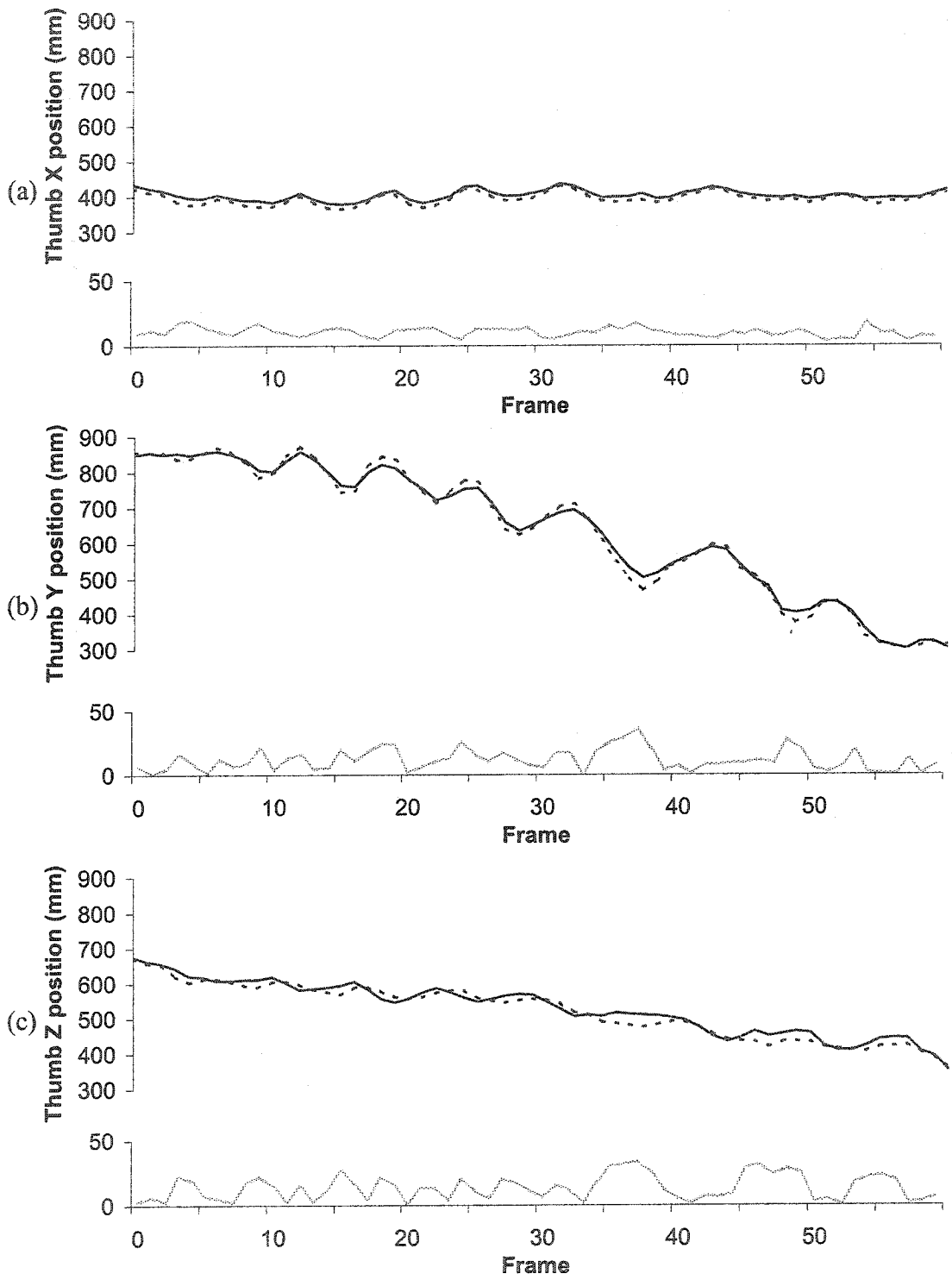


Figure 6.13 Thumb position for motion in *XZ* plane for (———) marker-based tracking, (------) markerless tracking and (·····) difference between marker-based and markerless tracking for: (a) *X*, (b) *Y* and (c) *Z*-axes. Points for the marker-based and markerless tracking are not identical and are expected to have differences (Sections 5.2 and 7.2.2)

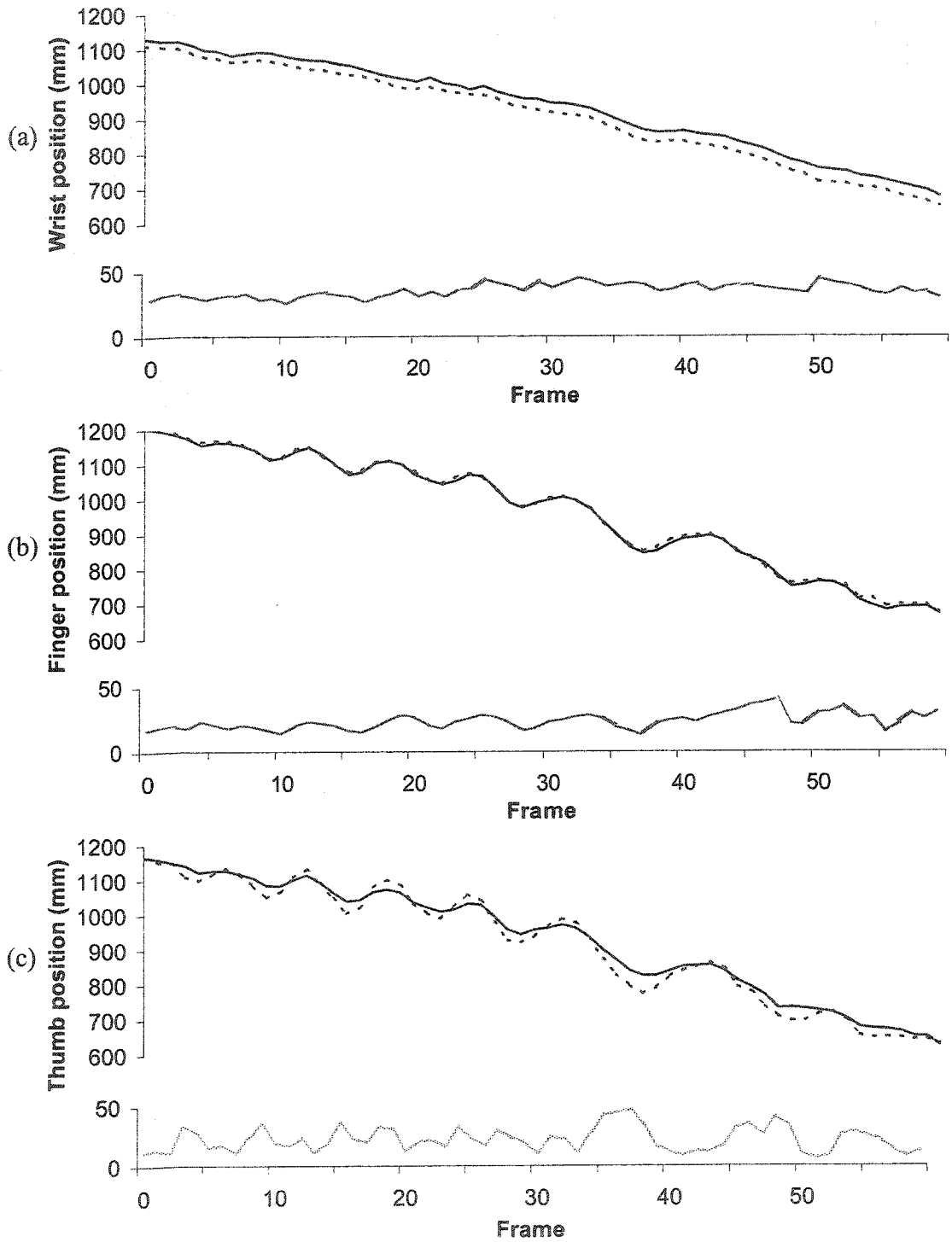


Figure 6.14 Total position of points of the hand in *XY* plane for (——) marker-based tracking, (------) markerless tracking and (·····) total difference between marker-based and markerless tracking for: (a) wrist, (b) index finger and (c) thumb. Points for the marker-based and markerless tracking are expected to be different (Secs. 5.2 and 7.2.2).

In the robot teleoperation by markerless tracking, the relative position and orientation of the human hand with respect to the hand starting position, are sent to the robot as relative positions and orientations to which the robot end-effector should move to, from the robot starting position. The actual position of Tool Control Point (TCP) on the robot manipulator arm and the actual orientation of the end-effector, as determined by the robot controller, were recorded during one of the teleoperation tests. The position and orientation of the human hand (input) and robot manipulator (output) for each frame, for Test 3 of Teleoperation Test Series 1 (tool-roll fixed), are shown in Figures 6.15 and 6.16, respectively. The graphs firstly indicate that the profile of the input data sent to the robot and the actual position and orientation reached by the robot match very closely. However, the profile of the robot lags compared to the input data profile sent, due to the processing time of the robot controller, the actual time required by the robot to reach the specified position and orientation, and the time for the user-written robot controlling program to received the actual robot position and orientation data.

The path of the hand used by the operator and that of the robot end-effector along the *X*, *Y* and *Z*-axes for the task of picking and placing the object can be seen in Figure 6.15. It can be observed, in Figures 6.15b and 6.15c, that for frames 1-80, the subject moved their hand towards the object while simultaneously moving down, from the robot starting position, to position the robot end-effector to grip the object. During the gripping process the subject overshot the distance in *Y* at around frame 25 and went below the intended height in *Z*, at frames 30-50. It can be observed in Figures 6.15b and 6.16a-b, for frames 40-90, the subject adjusted the position and orientation of the end-effector, respectively, in order to grip the object. The subject gripped the object at approximately frame 90 and then moved the end-effector up and towards the target until frame 120. For frames 120-160, the operator tried to adjust the position and orientation of the object over the target for final placement of the object on the target.

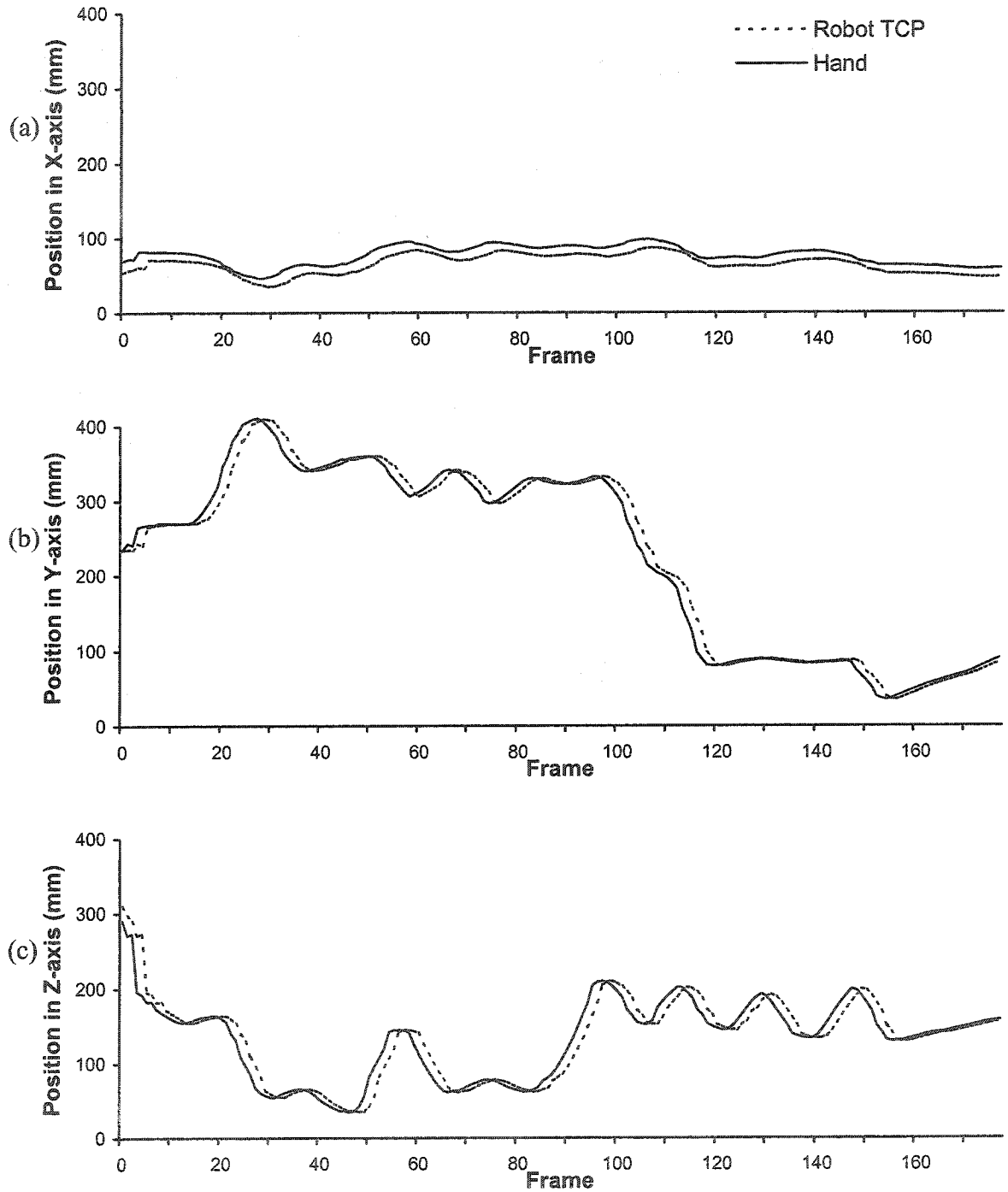


Figure 6.15 Relative positions of the human-hand and actual robot TCP in (a) *X*, (b) *Y* and (c) *Z*.

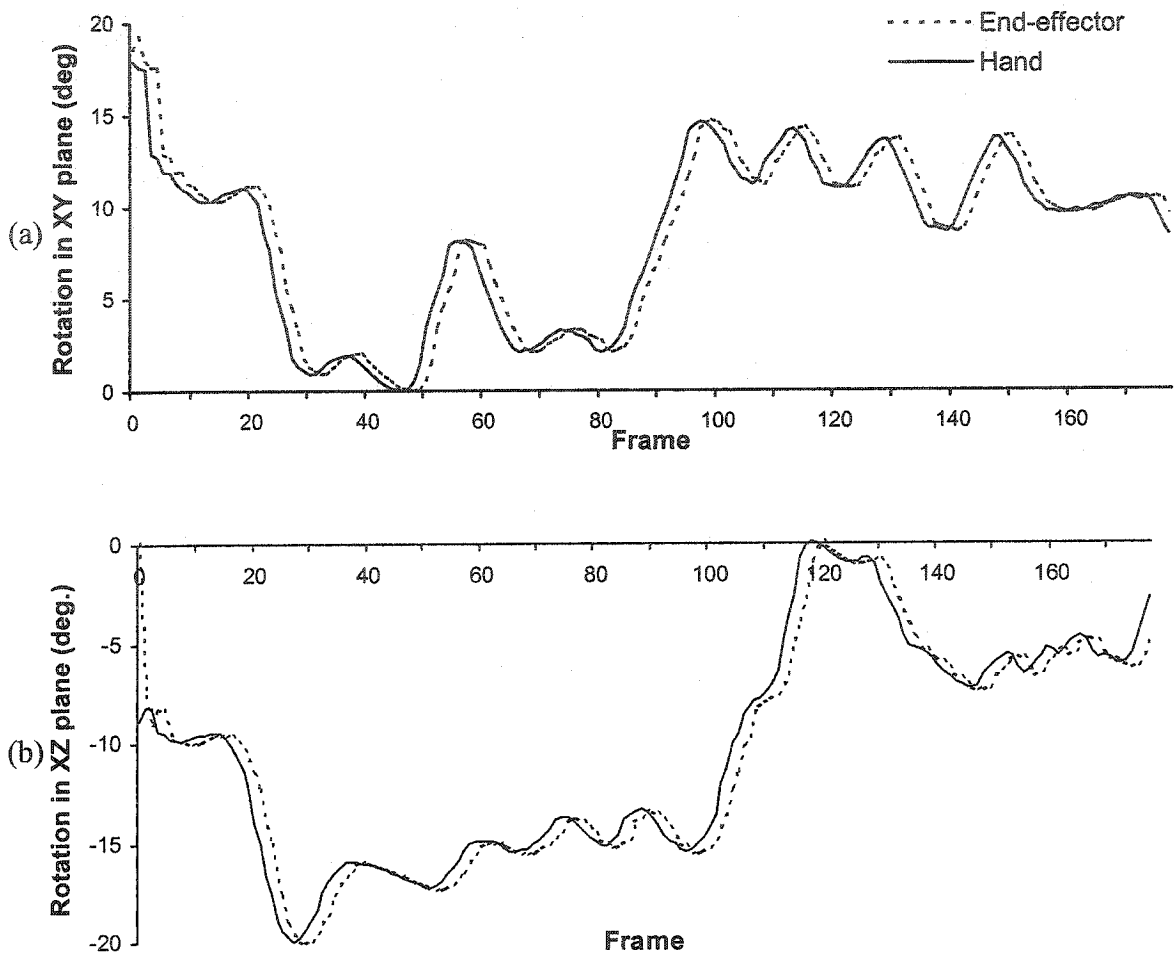


Figure 6.16 Relative orientations of the human-hand and actual robot end-effector in (a) *XY* plane and (b) *XZ* plane.

CHAPTER 7. DISCUSSION

7.1 Contribution of the markerless hand-arm tracking

In the current research, a non-invasive vision-based technique for tracking human hand-arm motion, based on the calculation of the elbow-joint centre, wrist-joint centre and the tips of the thumb and index finger was developed. The technique did not require any sensor on the human body for tracking and therefore avoided problems of hindrance to natural motion that could arise due to the presence of physical sensors and cables. As the tracking was performed using a markerless technique, there were no problems of markers getting occluded during motion. The technique used a simple one-step initialisation, hence requiring no assistance from an extra operator or requiring the operator to perform a series of predefined motion sequences. The technique was able to compute the 3D position and orientation of the hand-arm motion in real-time without the use of any approximate or customised model. The technique was accurate enough for a human operator to coarsely control a robot to perform object manipulation tasks.

7.2 Implications of the test results

The markerless tracking technique for determination of the position and orientation of the hand and arm during motion was evaluated based on the computed elbow and wrist joint-centre positions as well as those of the tips of the thumb and index finger. Three different experiments were performed to evaluate the accuracy of the markerless hand-arm tracking technique. The first experiment was an instantaneous-centre-of-rotation test to evaluate the accuracy of the calculation of elbow and wrist-joint centres. The second experiment was a comparison between markerless tracking and marker-based tracking, to evaluate the accuracy of the computed positions of wrist joint, thumb tip and index-finger tip. The third experiment was a teleoperation test that required full human-operator control of a robot for the complete task of object grasping and placement on a target.

7.2.1 Experiment 1: Instantaneous centre-of-rotation test

The experiment to determine the accuracy of the computation of the elbow and wrist-joint centres was based on the instantaneous-centre-of-rotation as assumed ground truth, as described in Section 5.1. The computed values of elbow (Section 6.1.1) and wrist joint centres (Section 6.1.2) using markerless tracking technique were compared to the ground truth-value of elbow and wrist-joint centre computed by the ICR technique, respectively.

The average absolute errors for the elbow joint were the smallest in *Y* and *Z*, 7.8 mm and 8.5 mm, respectively, compared to a slightly higher value of 10.8 mm in *X*. The average absolute error for the wrist joint was the smallest in *Y*, 7.6 mm, compared to higher values of error, 13.6 mm and 12.0 mm in *X* and *Z*, respectively. Part of the error may be due to experimental error as a result of limb segment movement that occurred when it should have been stationary. The instantaneous-centre-of-rotation test for the elbow, as described in Section 5.1.1, required that only the forearm and hand rotate, and the upper arm and elbow joint remain stationary. As no support or rest for the upper arm was used, the elbow-joint centre moved an average of 0.9 mm, 0.8 mm and 1.1 mm in *X*, *Y* and *Z*, respectively, in each frame for 75 frames. Therefore an average error of approximately 4.5 mm, 4.0 mm and 5.5 mm, respectively, was introduced in each data set that spanned 10 frames. The supports were not used for upper-arm in the experiment, as any support would have interfered with the processing of the arm images in the markerless tracking. For the wrist-centre experiment, explained in Section 5.1.2, it was required that only the hand be permitted to rotate while the upper arm, forearm and the wrist joint remain stationary. Again, an error similar to the error in the elbow-joint centre test was introduced, as there was no support or rest for the forearm. The wrist joint centre moved an average of 1.8 mm, 1.1 mm and 1.3 mm in *X*, *Y* and *Z*, respectively, in each frame, therefore introducing an average error of 9.0 mm, 5.5 mm and 7.5 mm, respectively, over every 10 frames in the experiment. This problem was more evident with the wrist-joint-centre test compared to the elbow-joint-centre test, as it is quite difficult to move only the hand while keeping forearm stationary. This is probably a reason for the value of the error being higher for the wrist-joint-centre test compared to the elbow-joint-centre test. The maximum average absolute errors of 10.8 mm for the

elbow and 13.6 mm for the wrist are somewhat high, yet acceptable for coarse tracking, as discussed later in reference to robot teleoperation.

7.2.2 Experiment 2: Comparison between markerless and marker-based tracking

In the experiment performed to compare markerless tracking to marker-based tracking, described in Section 5.2, the markerless tracking aims to find the tips of the index finger and thumb, while the marker-based tracking computes the centres of markers placed on the top of the index finger and thumb, respectively. Differences of approximately 14.9 mm for the index finger and 15.6 mm for the thumb are expected in the positions computed by the two different techniques, as explained in Section 5.2. The mean of the average total differences for the calculated index-finger position, over all the three tests (22.3 mm, 25.4 mm, 24.2 mm) from Tables 6.4, 6.7 and 6.10 was 24.0 mm, which is 9.1 mm off the expected difference (14.9 mm). The mean of the average total differences for the calculated thumb position, over all the three tests (31.6mm, 38.3 mm, 22.1 mm) from Tables 6.5, 6.8 and 6.11 was 31.3 mm, which is 15.7 mm off the expected difference (15.6 mm). The mean average total difference in positions between the two techniques is considered acceptable for the index finger, and high for the thumb. The high differences for the thumb were partly due to the thumb not being visible in both camera views, and the computation of the thumb position requiring estimation based on the small and index-finger positions as well as information from previous frames. However, the shapes of the thumb curves for the markerless and marker-based tracking do generally match, as shown in Figures 6.5, 6.9 and 6.13.

As the markerless tracking finds the wrist centre, while the marker-based tracking estimates the centroid of the marker placed on the radial side of the wrist, the difference between markerless and marker-based tracking positions is expected to be approximately 33 mm, half the ulnar-radial width of the arm at the wrist (Figure 5.4), as explained in Section 5.2. The average total differences for the calculated wrist joint position, over all three tests (29.3 mm, 29.5 mm, 36.2 mm) from Tables 6.3, 6.6 and 6.9, represent differences of 3.7 mm, 3.5 mm, and 3.2 mm, respectively, with a mean

3.5 mm, from the expected difference of 33.0 mm. These results for the wrist are considered quite good.

The results of the marker-based - markerless comparison experiments show that the markerless tracking technique was able to compute 3D positions with low error for the wrist and index finger. Although position errors for the thumb were greater than expected, the tracking is accurate enough to be acceptable for course control of the robot in the human-robot interface, as discussed further in the next section.

7.2.3 Experiment 3: Teleoperation of the robot

As the human-robot interface is the main application for the markerless hand-arm tracking, teleoperation tests were performed to demonstrate and evaluate the applicability of the markerless tracking as part of the interface, and to ascertain the level of hand-arm position accuracy required for a human operator to control robot movement.

The robot teleoperation test was successfully performed, with full human-operator control of the robot motion in order to perform the task of picking up an object and placing it on a predetermined target, using the vision-based feedback system, as explained in Section 5.3 and Chapter 4. The tests served to demonstrate the ability of the markerless-tracking technique to acceptably determine hand-arm positions, the ability of the robot manipulator to copy human hand-arm motions, and the ability of the operator to use the human-robot-manipulator interface. The operator had little difficulty in the task of grasping the object with the aid of all camera views, considering that the width of the object was only 15 mm less than the maximum-open-gripper width. The inclusion of Camera 4 for a view from the end-effector significantly aided the gripper alignment and grasping of the object. For the task of object placement on the target, the view from the end-effector-mounted camera was not available as the object occluded the target, and as expected, the placement task was more difficult than the object-grasping task. Similar qualitative results were found with the marker-based human teleoperation of the robot manipulator [3].

The average absolute errors for the first series of teleoperation tests, (tool-roll fixed) were 11.6 mm, 11.4 mm, and 11.0 deg, in X and Y translation, and Z rotation, respectively, relative to the target, and for all end-effector orientations permitted, were 33.7 mm, 29.0 mm, and 30.0 deg, in X and Y translation, and Z rotation, respectively. The human control of the robot was much better when one orientation was fixed, compared to when all orientations were permitted. The probable reason for poor results for Teleoperation Test Series 2, were related to the robot control for tool roll. In a similar test using marker-based tracking to control robot motion [3], robot manipulation failed by the robot moving to positions that were not sent to it, when tool-roll was permitted. This problem does not seem to be specifically related to the markerless tracking and is still to be explored. The control over position of the robot, which was governed by the wrist joint position, was quite good for both series of tests, but orientation control of the robot was a little difficult. This was partly due to the error in the 3D position of the wrist being much less than the error for the index finger and thumb tip positions. The results for Teleoperation Test Series 1 were quite good, considering the small size of the target (Figure 5.6) and object in the feedback display (Figure 4.4), the distance of the monitor from the operator (122 cm), as well as the view of the target from the end-effector being unavailable.

The curves in Figures 6.15 and 6.16 indicate that the human operator was able to perform gross object manipulation tasks without much difficulty. However, there was some limitation in the ability to control the fine movement of the end-effector as seen by the overshooting and correction of the gripper position and orientation. It should be noted, however, that similar results were found for object manipulation by marker-based teleoperation [3]. To address the problem of fine adjustment in the marker-based teleoperation, a technique of semi-autonomous control was developed [3], whereby the fine control is performed by a separate vision-based system at the robot site. This method would also be implemented with the markerless tracking to perform fine gripper adjustment.

From the results of the robot teleoperation experiment, it seems that control of the robot motion was possible despite hand-arm tracking errors. This was due to implementation of an averaging filter, which averaged the position and orientation data computed by the markerless tracking over 9 frames, before sending information to the robot. However, this averaging contributes to the lag between the robot motion and hand-arm positions.

7.3 Comparison of results with previous work

The presented markerless motion tracking method is an improvement over other conventional human hand-arm motion tracking techniques. Firstly, the presented method is a markerless technique, not limited to the use of markers or sensors on the human body to track motion [2-3,13]. Furthermore, the technique discussed in this thesis is self-initialising and does not require any extra operator to initialise it or require the subject to perform predetermined a motion sequence [22-24]. The technique described in this thesis uses a model-free method to perform 3D motion tracking of the hand-arm movement in real-time. This was achieved with a low mean-absolute average total difference for the wrist joint, 3.5 mm from the expected value, in the test of marker-based and markerless tracking (Section 5.2). This error is much lower than the error of ± 50 mm along the X and Y direction for a similar test that involved tracing a known path in 2D [18]. None of the other techniques in the scientific literature that performed markerless arm-motion tracking, provided any quantitative experimental data to show the accuracy of their techniques [15-16, 26]. The technique presented in this research calculates the 3D position of the wrist joint, and thumb and index-finger tips to determine the orientation of the hand, but none of the techniques that track arm motion in the literature [15-16, 18, 26] attempted to track the orientation of the hand, along with the position of the arm. The object-positioning errors of Teleoperation Test Series 1, 11.6 mm, 11.4 mm, and 11.0 deg, were slightly high compared to the results of a similar test using marker-based tracking [3], 1.6 mm, 8.5 mm and 3.2 deg, in X and Y translation, and Z rotation, respectively. This was due to the higher errors in markerless tracking; however, different operators with different levels of training were employed for the two types of tracking, and a direct comparison has limited significance. On the other hand, the ability to

perform the task of object manipulation without the use of markers in tracking, is an improvement over the marker-based teleoperation, as the operator can move the hand without the markers getting hidden and can reach orientations and positions that were not attainable with marker-based tracking. Furthermore, the operator can perform motions naturally without having to think about avoiding positions and orientations where the markers could be hidden.

7.4 Limitations

The markerless hand-arm tracking presented in this research has some limitations that are explained in detail below.

The technique for markerless tracking is based on the calculation of the position and orientation of the hand-arm motion in real-time. It is developed to process the images of an unclothed arm against a black background. This requires that the rest of the body does not appear in the image. Also, this technique will only be able to provide any meaningful results when the hand and arm are completely visible in the camera images and the operator is not wearing any clothes on their arm. As the technique is based on thresholding the image against a black background, the method will not be able to segment the arm from the background if the background is cluttered or has high variation in intensity.

The joint-centre positions of the wrist and elbow are calculated from the best-fit medial lines for the forearm and upper-arm. This technique will not be able to compute the joint centre positions if the arm is directly pointing towards any one of the cameras. In such a case, the arm will appear as a circle and medial-line calculations would not be possible. Moreover, as the cameras view the arm from the sides, it is not possible to process orientations of the hand when the hand is hidden by the arm. This problem can be handled by using images from more than two camera views, such that in any hand orientation, the hand, forearm and upper-arm are visible from at least two camera views. One limitation of the study (not the technique) is that not all possible orientations of the hand were tested in tracking.

The elbow and wrist joint centre calculations in the current frame use information of the positions of the joint centres in the previous frame. If the movement between frames is too great, the elbow and wrist separators (Section 3.2.4) will not be positioned properly to divide the arm into three parts. In such a case, the algorithm may not be able to process the image correctly. It is therefore required that the arm motion be slow enough to maintain a small arm movement between two consecutive frames.

The orientation calculation of the hand requires that the thumb and index finger always form a fixed U-shape (Section 3.1.1) to simulate the robot-gripper shape. The tracking technique is based on the assumption that subject will maintain that shape throughout their arm motion, and the technique will therefore not be able to process correct orientation information if the hand deviates from this general shape.

7.5 Sources of error

The computation of the 3D positions of the elbow joint, wrist joint, thumb tip and index-finger tips in the markerless hand-arm tracking has some sources of error which are not part of the tracking algorithms.

One major source of error in the computation of the 3D positions of the arm-joint centres and thumb and index finger tips is the error introduced by the DLT camera calibration. The average absolute error for the DLT camera calibration was 2.6 mm, 4.4 mm, and 3.6 mm in X, Y and Z, respectively, with an average total absolute error of 7.2 mm. The DLT camera calibration can therefore introduce an error of approximately 7.2 mm to each calculated 3D point. This error would appear for both the marker-based and markerless tracking.

The technique is based on the assumption that the images from both camera views are grabbed simultaneously using synchronized cameras. This would ensure that both camera images correspond to the same hand-arm position. As the cameras used for image acquisition were not synchronized due to hardware limitations, an error could therefore

have been introduced due to the motion of the hand and arm between the capture of the two images (from two cameras) for the same frame.

The calculation of wrist-joint centre is based on the position of the elbow joint centre, as the latter is used in wrist-joint centre correction, as explained in Section 3.2.10. The error in the elbow-joint position is therefore accumulated in the computed wrist position. Also, the calculation of the thumb tip is based on the position of the index-finger tip and wrist-joint centre, in the cases where the thumb is not visible in one or both images, as explained in Sections 3.2.11.2 and 3.2.11.3. Therefore, the error in the elbow joint, wrist joint, and index-finger tip positions would be accumulated in the computed thumb position.

The tracking technique is based on the assumption that the calculated 2D image points in an image pair of the same frame (for any given hand-arm point) correspond to the same point in 3D. Any error in determining the positions of the 2D points will cause an error in the 3D calculation. In other words, two image points computed with error do not correspond to the correct 3D point.

CHAPTER 8. CONCLUSION

8.1 Contribution of present work

A non-invasive technique for real-time markerless tracking of human hand-arm motion by calculating the location of the elbow joint, wrist joint, thumb tip and index-finger tip for an unclothed arm has been developed. The technique uses a one step initialisation, followed by model-free tracking of human-arm movement in 3D space. This technique tracks motion of the arm and hand, without any prior knowledge of the subject's arm length, color, and width and distance from the cameras. The markerless tracking technique was successfully used in the human-robot interface to perform robot-teleoperation in real-time.

8.2 Conclusions based on tests and analyses

Based on the instantaneous centre of rotation tests, the maximum average absolute errors of 10.8 mm for the elbow and 13.6 mm for the wrist were acceptable for coarse tracking in robot teleoperation. The results of the marker-based - markerless comparison experiments show that the markerless tracking technique was able to compute 3D positions with low error for the wrist and index finger. Although position errors for the thumb were greater than expected, the tracking is accurate enough to be acceptable for course control of the robot in the human-robot interface. The markerless tracking technique was capable of computing the 3D position and orientation of hand arm-motion in real-time accurately enough for a human operator to control the motion of a robot to perform a task of picking and placing an object on a target.

8.3 Recommendations for further research

The technique for the markerless hand-arm-motion tracking uses two-camera views to track the motion. The technique can be extended and made more robust using more than two camera views for processing, so that the upper-arm, forearm, hand and thumb are clearly visible at all times in at least two views. As the greatest errors occurred when the thumb was occluded, the use of more than two camera views would improve the accuracy

of the technique. Moreover, information from more than two cameras can be used in a least-squares approach to improve the tracking accuracy. The use of higher resolution cameras would improve the accuracy in 2D computations both in the camera calibration and the markerless tracking.

The camera calibration could be performed using alternative techniques, such as those that include more parameters to handle lens distortion, a more accurately constructed calibration device, and a neural-network-based mapping of 2D image coordinates to 3D object space.

The markerless tracking technique could be extended to compute the shoulder-joint centre, such that if the rest of the body appears in the image, it can be disregarded during tracking.

It is recommended to use cameras and image acquisition hardware that allow synchronized image acquisition from multiple cameras. This would eliminate the errors caused by movement of the arm between the capture of images for the same frame. It is also recommended to use a faster processor to reduce computation time for each image. A faster rate of image processing would allow the user to move their hand at a higher speed and therefore in a more natural manner while performing a task.

The vision-based human-robot-manipulator interface permitted fairly accurate real-time object manipulation by robot teleoperation for the camera views available. It is recommended, however, to use higher resolution cameras, with less distortion to give a better view of the object and target in the feedback system. As well, alternative positions of cameras mounted on the robot gripper and robot should be considered to enable close-range visual feedback when placing an object on a target [3].

REFERENCES

1. H. Hügli, G. Maître, F. Tièche and C. Facchinetti, "Vision-based behaviours for robot navigation", *Proceedings of the Fourth Annual SGAICO Meeting, Neuchâtel* (1992).
2. J. Kofman, T. Luu and S. Verma, "A vision-based human-robot-manipulator interface", *Fourth International Conference on Industrial Automation, Montreal, Canada* (2003).
3. J. Kofman, S. Verma, X. Wu and T. Luu, "Teleoperation of a robot manipulator from 3D human hand-arm motion", *SPIE International Symposium on Photonics Technologies for Robotics, Automation, and Manufacturing, Providence, USA*, 5264, pp.257-265 (2003).
4. F. Tièche, N.S. Ghai and H. Hügli, "Self-positioning and localisation of a mobile robot using vision-based behaviours", *Proceedings of mechatronics, Europe-Asia congress on mechatronics, Besançon, France* (1996).
5. C. Peters and O´ Sullivan, "Vision-based reaching for autonomous virtual humans", *Proceedings of AISB´02 symposium: Animating Expressive Characters for Social Interactions* (2002).
6. S. Yeates, E.J. Holden and R. Owens, "Real-time 3D graphics for human modelling and teaching sign language", *International Conference on Computer Vision and Graphics, Poland* (2002).
7. T.E. Boult, R. Micheals, A. Erkan, P. Lewis, C. Powers, C. Qian and W. Yin, "Frame-rate multi-body tracking for surveillance", *In Proceedings of DARPA IUW*, pp. 305-308 (1998).
8. N.A. Borghese and P. Rigiroli, "Tracking densely moving markers", *IEEE First International Symposium on 3D Data Processing and Transmission*, Padova giugno, pp. 682-685 (2002).
9. T. Harada, T. Sato and T. Mori, "Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed", *Workshop on Human Motion*, pp. 99-106 (2000).

10. E.R. Bachmann, I. Duman, U. Usta, R.B. McGhee, X. Yun, and M.J. Zyda, "Orientation tracking for humans and robots using inertial sensors", *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 187-194 (1999).

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

12. E.R. Bachmann, R.B. McGhee, X.Yun and M.J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments", *ACM Symposium on Virtual Reality Software and Technology (VRST), Banff, Canada,* pp. 9-16 (2001).

13. A.M. Chaudhari, R.W. Bragg, E.J. Alexander and T.P. Andriacchi, "A video-based markerless motion tracking system for biomechanical analysis in an arbitrary environment", *BED-50 Bioengineering Conference,* pp. 777-778 (2001).

14. D. Bullock and J. Zelek, "3-D human limb tracking using monocular vision in unconstrained environments", Accepted in *IEEE Computer Graphics and Applications: Interactive Tracking Issue* (2003).

15. D. Bullock and J. Zelek, "Automatic target detection and model acquisition for 3D Visual Tracking of human body parts", Accepted in *Computer Vision and Image Understanding: Event Detection Issue* (2003).

16. I.A. Kakadiaris, D. Metaxas, and R. Bajcsy, "Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach", *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 980-984 (1994).

17. I.A. Kakadiaris, D. Metaxas and R. Bajcsy, "Active motion-based segmentation of human body outlines", *Proceedings of the Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 50-56 (1994).

18. L. Goncalves, E.D. Bernardo, E. Ursella and P. Perona, "Monocular tracking of the human arm in 3D", *Proceedings of IEEE International Conference on Computer Vision, ICCV 95,* pp. 764-770 (1995).

19. E.D. Bernardo, L.Goncalves and P.Perona, "Monocular tracking of the human arm in 3D: Real-time implementation and experiments", *Proceedings of the International Conference on Pattern Recognition, ICPR 96*, pp. 622-626 (1996).

20. J. Triesch and C. Malsburg, "A system for person-independent hand posture recognition against complex Backgrounds", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(12), pp. 1449-1453 (2002).

21. R. Plänkers and P. Fua, "Articulated soft objects for video-based body modelling", *Proceedings of International Conference on Computer Vision, Vancouver, Canada,* pp. 394-401 (2001).

22. R. Rosales and S. Scarloff, "Inferring body pose without tracking body parts", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* **2**, pp. 721-727 (2000).

23. K. Koara, A. Nishikawa and F. Miyazaki, "Hierarchical part decomposition method of articulated body contour, and its application to human body motion measurement", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2000)*, pp. 2055-2060 (2000).

24. K. Koara, A. Nishikawa and F. Miyazaki, "Contour based hierarchical part decomposition method for human body motion analysis from video sequence", *In Human Friendly* Mechatronics (Elsevier *Science),* pp. 235-240 (2001).

25. I.A. Kakadiaris and D. Metaxas, "3D human body model acquisition from multiple views", *Proceedings of the IEEE Fifth International Conference on Computer Vision*, pp. 618-623 (1995).

26. I.A. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(12), pp. 1453-1459 (2000).

27. R. Rosales and S. Scarloff, "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pp. 117-123 (1999).

28. E. Ueda, Y. Matsumoto, M. Imai and T. Ogasawara, "Hand pose estimation for vision based human interface", *10th IEEE International Workshop on Robot and Human Communication (ROMAN 2001)*, pp. 473-478 (2001).

29. I. Cohen, G. Medioni and H. Gu, "Inference of 3D human body posture from multiple cameras for vision-based user interfaces", *5th World Multi-Conference on Systemics, Cybernetics and Informatics* (2001).

30. J. M. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking", *In J. Eklundh, (Ed.), Proceedings of 3rd Europe Conference Computer Vision*, **2**, Springer-Verlag, pp. 35-46 (1994).

31. C. Sminchisescu and B. Triggs, "Covariance scaled sampling for monocular 3D body tracking", *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 447-454 (2001).

32. C.R. Wren, A.J. Azarbayejani, T. Darrell and A.P. Pentland, "Pfinder: Real-time tracking of the human body", *2nd IEEE International Conference on Automatic Face and Gesture Recognition* (1996).

33. C.R. Wren, A.J. Azarbayejani, T. Darrell and A.P. Pentland, "Pfinder: Real-time tracking of the human body", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), pp. 780-785 (1997).

34. Q. Cai and J.K. Aggarwal, "Tracking human motion in structured environments using a distributed-camera System", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(11), pp. 1241-1247 (1999).

35. J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking", *Proceeding of European Conference on Computer Vision*, **2**, pp. 3-19 (2000).

36. A.J. Azarbayejani, C.R. Wren and A.P. Pentland, "Real-time 3-D tracking of the human body", *Proceedings of IMAGE'COM 96, Bordeaux, France* (1996).

37. I.A. Kakadiaris and D. Metaxas, "Vision-based animation of digital humans", *IEEE Computer Animation, CA-98, Philadelphia* (1998).

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

39. Y.I. Abdel-Aziz and H.M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry", *Proceedings Symposium of Close-Range Photogrammetry, Am Soc Photogrammetry,* pp. 1-18 (1971).

40. T. Luu, "Development of a teleoperated robot system based on human-hand tracking", Project report, University of Ottawa (2002).